SeeBeyond ICAN Suite

eVision Studio User's Guide

Release 5.0.2

SEEBEYOND®

SeeBeyond Proprietary and Confidential

Version 20031223133007.

Contents

List of Figures 7

List of Tables 11

Chapter 1

Introduction 12

Document Purpose and Scope 12

Intended Audience 12

Organization of Information 13

Writing Conventions 13

Additional Conventions 13

Online Documentation 14

The SeeBeyond Web Site 14

Chapter 2

About eVision Studio 15

eVision Web Applications 15

ICAN Suite Integration 16

MVC Architecture 17

MVC Application Architecture in eVision 18

eVision Page Flow Engine 18

eVision Web Application Design Components 19

Page Layout Designer 19

Page Flow Designer 19

Chapter 6

Installing eVision Studio 20

System Requirements 20

Database Support 20

Installation Steps 21

Chapter 7

eVision Studio Components 29

Design Tools 29
eVision Studio Toolbar 30
eVision Studio Tools 31
Disabling the Undo Function 32

Graphical User Interface Components 32

GUI Component Selection Palettes 33
HTML Objects 34
About Dynamic Tables 36
Adding Pagination to a Dynamic Table 36
About Image Maps 38
Importing Images 41
Form Objects 43

Defining Component Properties 44
Moving Overlapping Components in the Z-Direction 46
Adding Multiple Options to Form Object Components 47
Creating Multiple Entries in a Drop-Down List 48
Creating a Group of Radio Buttons 49
Creating a Group of Check Boxes 50

Customizing a Style Sheet 51
Creating a new style sheet from eVision.CSS 51

Importing a Style Sheet Into Your Project 53
Applying a Style Sheet to Your Web Application 55
Applying Classes 57
Applying a Class to an Object 58

Linking Pages 58

Chapter 8

Using The Page Layout Designer 62

Laying Out a Web Page 62
Before you Begin 63
Downloading the Sample Project files 63
Starting a New Page Layout 65
Importing Images 68
Placing GUI Components on the Canvas 70
Create the Background Layer Component 71
Create the Page Banner Component 73
Create the Employee Name Component 74
Create the Hours Worked Label GUI Component 74
Create the Rate Label Component 75
Swap the HTML Objects and Form Objects Palettes 75
Create the Employee Name Input Field Component 76
Create the Hours Worked Input Field Component 77

Create the Rate Drop-down List Box Component 78
Create the Submit Button Component 80
Reviewing the Page Layout 80
Aligning Objects Using the Alignment Tools 81
Previewing the Finished Web Page 82
Saving the Project in the Repository 82
Manipulating Objects on the Canvas 83
Moving a Single Object 83
Moving a Temporary Group of Objects 83
Resizing Objects 84
Resizing Horizontal Lines 84
Checking the Page Layout in to Version Control 84
Chapter 9
Page Flow Designer 87
Creating a Page Flow 87
Adding a Page Flow to a Project 87
Adding Web Pages to a Page Flow 88
Creating a Page Flow 88
Using the Page Flow Designer GUI 89
Page Flow Elements 90
Sub-process Elements 90
Branching Elements 91
Adding a Branching element 91
Intermediate Events 92
Adding an Intermediate event 93
While 93
Links 93
Validating a Page Flow 94
Saving a Page Flow 94
Configuring Page Flow Designer Elements 95
About Business Rule Designer 95
Adding an Inline Business Rule 95
Method Palettes 96
Sub-Processes 96
Page Flow Properties 97
General 98
Page Flow Attributes 98
Correlation Keys and Sets 100
WSDL Files 104
Chapter 10
Creating and Deploying a Web Application 106
Overview 106
Creating the Sample Application 106

Downloading the Sample Project 107
Importing the Sample Project into the Repository 108
Version Control: Checking Out the Project Components 112

Assembling the Sample Project Components 112
Step 1: Creating or importing a Project 112
Step 2: Create the Necessary Web Pages 112
Step 3: Create the Page Flow 113
Step 4: Create a New Environment 113
Step 5: Create a Connectivity Map and Deployment Profile 113
Step 6: Run and Test the System 113
Page Flow Designer Menu 113
Creating the Page Flow 113
Adding Page Flow Elements 113
Connecting the Page Flow Components 115
Configuring the Business Rule Elements 116
Configuring the While Loop 117
Defining the Properties for Case 1 121
Mapping the Employee Name Output 123
Mapping the Page Flow Attributes 124
Creating the Connectivity Map 125
Creating the Environment 126
Creating the Deployment Profile 127
Configuring the Deployment Profile 128
Running and Testing the System 130
How to Start the Logical Host 130
Logical Host Startup Parameters 130
Starting the Logical Host 132
Before you begin 132

Accessing the Web Application 132
Importing the Working Sample Project 133

Method Palettes 134

Operators 134

String 137

Number 141

Boolean 143

Nodes 144

Index 147

List of Figures

Figure 1 eVision Studio integration With the ICAN Suite 17
Figure 2 Model/View/Controller (MVC) Architecture 18
Figure 3 Enterprise Manager ADMIN Page 22
Figure 4 Selecting the Products Manifest File 22
Figure 5 ICAN Suite Products List 23
Figure 6 eVision.sar File 23
Figure 7 upload Now Button 24
Figure 8 Product Upload Progress Bar 24
Figure 9 Updating eVision Studio 24
Figure 10 Update Center Wizard: Select Modules to Install 25
Figure 11 Update Center Wizard: Progress Bars 26
Figure 12 Certificates and Installed Modules List 27
Figure 13 Update Center Wizard: Restart the Enterprise Designer 28
Figure 14 The Page Layout Designer Canvas at Startup 30
Figure 15 eVision Studio Toolbar 30
Figure 16 Page Layout Designer GUI Components 33
Figure 17 HTML and Form Objects Palettes 34
Figure 18 Pagination buttons 37
Figure 19 Paginate Menu 37
Figure 20 Pagination Properties 37
Figure 21 Image Map Properties Sheet 38
Figure 22 Selecting an Image for an Image Map 39
Figure 23 Searching the Project for an Image File 40
Figure 24 Resized Bounding Box for Image Map 41
Figure 25 Importing Images into a Project 42
Figure 26 Imported Images in Enterprise Explorer 42
Figure 27 Properties Drop-down Menu 44
Figure 28 New Page Properties 45
Figure 29 Swapping Component Properties Formats 45
Figure 30 Command Button 46
Figure 31 Interface Elements Displayed Via a Command Button 46
Figure 32 Moving Components in the Z-direction 47

Figure 33 Editing Component Options 48

Figure 34 Edit Options Dialog Box 49

Figure 35 Drop-Down List Preview 49

Figure 36 Editing Radio Button Group Options 50

Figure 37 Editing Check Box Group Options 50

Figure 38 Choosing a Style Sheet to Edit 51

Figure 39 eVision Style Editor 52

Figure 40 eVision Style Editor Tools 53

Figure 41 Importing a Style Sheet as a New File 54

Figure 42 CSS Chooser Dialog Box 54

Figure 43 Style Sheet in Enterprise Explorer 55

Figure 44 Changing the Properties Window 55

Figure 45 Exploring for a Style Sheet 56

Figure 46 Moving Up a Level in the CSS Explorer 56

Figure 47 Selecting a Style Sheet to Import Into the Project 57

Figure 48 Style Sheet with Classes 57

Figure 49 Mapping a Class to a Component 58

Figure 50 Creating a Page Link 59

Figure 51 Page 1 of 2 in the Page Link Wizard 60

Figure 52 Page 2 of 2 in the Page Link Wizard 60

Figure 53 Links in Enterprise Explorer 61

Figure 54 The Page Layout Designer Web Page Project 63

Figure 55 Project Context Menu 65

Figure 56 Naming a New Project 65

Figure 57 New Page Layout Context Menu 66

Figure 58 Page Layout Wizard Page 1 66

Figure 59 Page Layout Wizard Page 2 67

Figure 60 Page Layout Icon in Project Explorer 67

Figure 61 eVision GUI with a Blank Canvas 68

Figure 62 Importing an Image into the Repository 69

Figure 63 Image Source Dialog Box 70

Figure 64 Graphic Image in Enterprise Explorer 70

Figure 65 Bringing the HTML Objects Selection Palette Forward 71

Figure 66 Selecting an Image for the Image Object Container 71

Figure 67 Component Properties Sheet for the Image Component 72

Figure 68 HTML and Form Objects Palettes 76

Figure 69 Incomplete Page 78

Figure 70 Drop-Down List Menu 79

Figure 71 Edit Options Dialog Box 79
Figure 72 Finished Input Page 81
Figure 73 eVision Studio Toolbar 82
Figure 74 Web Page Preview 83
Figure 75 Version Control Check-in Procedure 85
Figure 76 Version History 86
Figure 77 Checking Out a Version 86
Figure 78 Page Flow in the Page Flow Designer GUI 88
Figure 79 The Page Flow Designer Toolbar 89
Figure 80 Starting a Link 94
Figure 81 Finished Link 94
Figure 82 Business Rule Designer 95
Figure 83 Business Rule Designer Toolbar 96
Figure 84 Opening the Page Flow Property Sheet 97
Figure 85 Page Flow Properties Dialog Box: General Tab 98
Figure 86 New Page Flow Attribute 99
Figure 87 Page Flow Properties Dialog Box: Page Flow Attributes tab 100
Figure 88 Opening the Page Flow Property Sheet 101
Figure 89 Page Flow Properties: Correlations Tab 102
Figure 90 New Correlation Key Dialog Box 102
Figure 91 New Correlation Set Dialog Box 103
Figure 92 Page Flow Designer Element Properties Window 104
Figure 93 Page Flow Properties: WSDL Tab 105
Figure 94 Load WSDL Dialog Box 105
Figure 95 Sample Vacation Request Application 106
Figure 96 Enterprise Designer Startup Window 109
Figure 97 Enterprise Designer Login Window 109
Figure 98 Import Project Menu 110
Figure 99 Selecting the Sample File to Import 110
Figure 100 Import Progress Bar 110
Figure 101 Import Confirmation Message 111
Figure 102 Repository Refresh Confirmation 111
Figure 103 Sample Project Components in Enterprise Explorer 112
Figure 104 The Page Flow Designer Menu 113
Figure 105 Starting the Page Flow 114
Figure 106 Page Flow Elements in the While Loop 115
Figure 107 Unconnected Page Flow Components 115
Figure 108 Connected Page Flow Components 116

Figure 109 Starting the Business Rule Designer 116
Figure 110 Mapping a String Literal 117
Figure 111 While Loop Icon 118
Figure 112 Mapping Within the While Loop 119
Figure 113 Decision Gate in While Loop 119
Figure 114 Decision Gate Properties Window 120
Figure 115 Launching the Method Palette 120
Figure 116 Method Palette with Operator Tab 121
Figure 117 Defining Decision Gate Properties: Case 1 122
Figure 118 Defining Decision Gate Properties: Case 2 123
Figure 119 Business Rule Element in the While Loop 123
Figure 120 Mapping Employee Name Output to Text Input 124
Figure 121 Adding a Business Rule 124
Figure 122 Output to Input Mapping 125
Figure 123 New Connectivity Map 126
Figure 124 Creating a New Environment 126
Figure 125 Creating a New Logical Host 127
Figure 126 Creating a New Integration Server 127
Figure 127 Creating a new Deployment Profile 128
Figure 128 New Deployment Profile Dialog Box 128
Figure 129 Configuring the Deployment Profile 129
Figure 130 Activating the Deployment Profile 129
Figure 131 Do Not Start the Logical Host 129
Figure 132 eVision Project Startup URL 130
Figure 133 Invoking the Bootstrap Command 132
Figure 134 Sample Web Application Startup Page 133
Figure 135 Method Palette: Operator tab 134
Figure 136 Method Palette: String tab 137
Figure 137 Method Palette: Number tab 141
Figure 138 Method Palette: Boolean tab 143
Figure 139 Method Palette: Nodes tab 144

List of Tables

Table 1 Writing Conventions 13

Table 2 eVision Studio Web page design tools 31

Table 3 HTML Objects 34

Table 4 Form Objects 43

Table 5 eVision Studio Object Alignment tools 82

Table 6 The Page Flow Designer GUI Tools 89

Table 7 Connecting and Sub-process elements 90

Table 8 Branching Elements 92

Table 9 Intermediate Events 92

Table 10 While loop 93

Table 11 Command Arguments for bootstrap 130

Table 12 Operator Methods 134

Table 13 String Methods 138

Table 14 Number Methods 141

Table 15 Boolean Methods 143

Table 16 Nodes Methods 145

Chapter 1

Introduction

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

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

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

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

1.3 Writing Conventions

The following writing conventions are observed throughout this document.

Table 1 Writing Conventions

Text	Convention	Example
Button, file, icon, parameter, variable, method, menu, and object names.	**Bold** text	▪ Click **OK** to save and close. ▪ From the **File** menu, select **Exit**. ▪ Select the **logicalhost.exe** file. ▪ Enter the **timeout** value. ▪ Use the **getClassName()** method. ▪ Configure the **Inbound** File eWay.
Command line arguments and code samples	`Fixed` font. Variables are shown in ***bold italic***.	`bootstrap -p` ***password***
Hypertext links	**Blue** text	For example: See **"Online Documentation" on page 14**.

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

Chapter 2

About eVision Studio

This chapter provides an overview of eVision Studio.

This Chapter Includes:

- **"eVision Web Applications"**
- **"MVC Architecture" on page 17**
- **"eVision Page Flow Engine" on page 18**
- **"eVision Web Application Design Components" on page 19**

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 Suite Integration

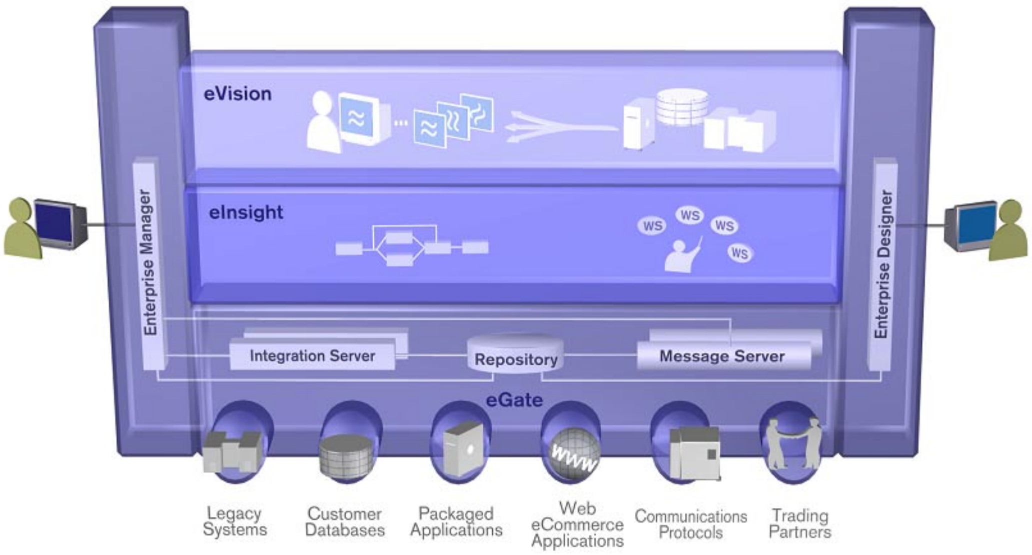
eVision is fully integrated with the ICAN Suite and runs as a component within the ICAN Suite run-time environment. The Page Layout and Page Flow Designers run as components within the Enterprise Designer GUI. eVision Web applications are stored in the SeeBeyond Repository.

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

- Resource Management - The ICAN Suite uses a distributed and open architecture that enables components to access system resources (memory and processing power) as needed and in conjunction with other components.
- Security - The ICAN Suite provides a security module, to fulfill security needs such as authentication and authorization access to eVision functions.
- Repository storage and access - The setup, component, and configuration information for the elements of a Project, including related eVision components, are stored in the Repository.
- Deployment abilities - The ICAN Suite provides deployment profiles that contain all 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 17**.

Figure 1 eVision Studio integration With the ICAN Suite



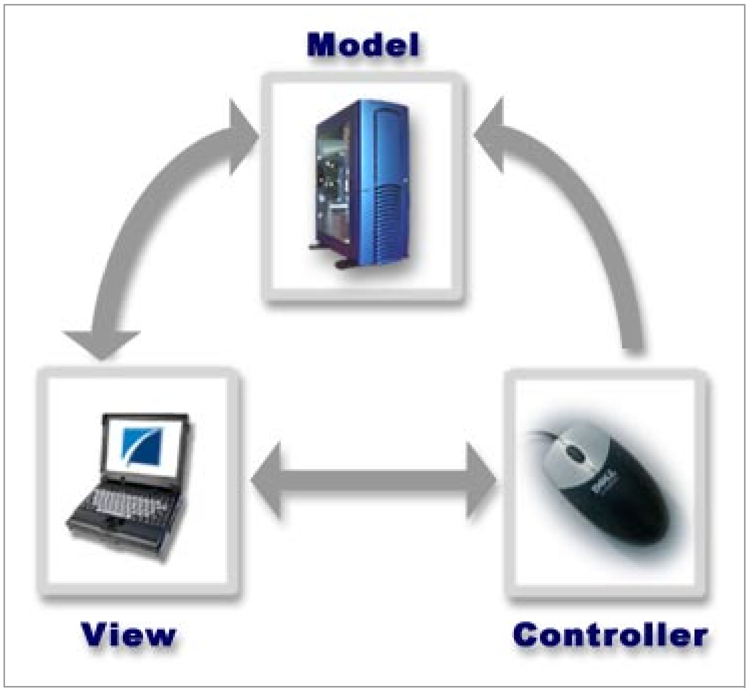
2.2 MVC Architecture

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 18**.

- 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 the **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

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.

2.4 eVision Web Application Design Components

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

2.4.1 Page Layout Designer

The Page Layout Designer allows the Web designer to specify the details of individual pages in the application using familiar drag-and-drop techniques to place **Graphical User Interface Components** (Web page design objects) onto a blank page design canvas. The Page Layout Designer provides the user with a comprehensive collection of these prebuilt GUI components, and a friendly, WYSIWYG Web page editor. The Page Layout Designer components are described in detail in **"Design Tools" on page 29**

2.4.2 Page Flow Designer

The Page Flow Designer is the component within the SeeBeyond ICAN Suite that facilitates the flow of Web-based business activities. The Page Flow Designer function includes 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 the Page Flow Designer's set of intuitive, graphical modeling tools. At run-time, the page flow drives the display of eVision Web pages and orchestrates the interactions with backend systems.

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

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
- 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

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

6.1.1. Database Support

Note: *eVision does not require a database.*

eVision supports the following databases:

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

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

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 ***SeeBeyond ICAN Suite Installation Guide***, which is available via the Enterprise Manager's Documentation Web page.

Before You Begin

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

To install eVision Studio, follow these steps:

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

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

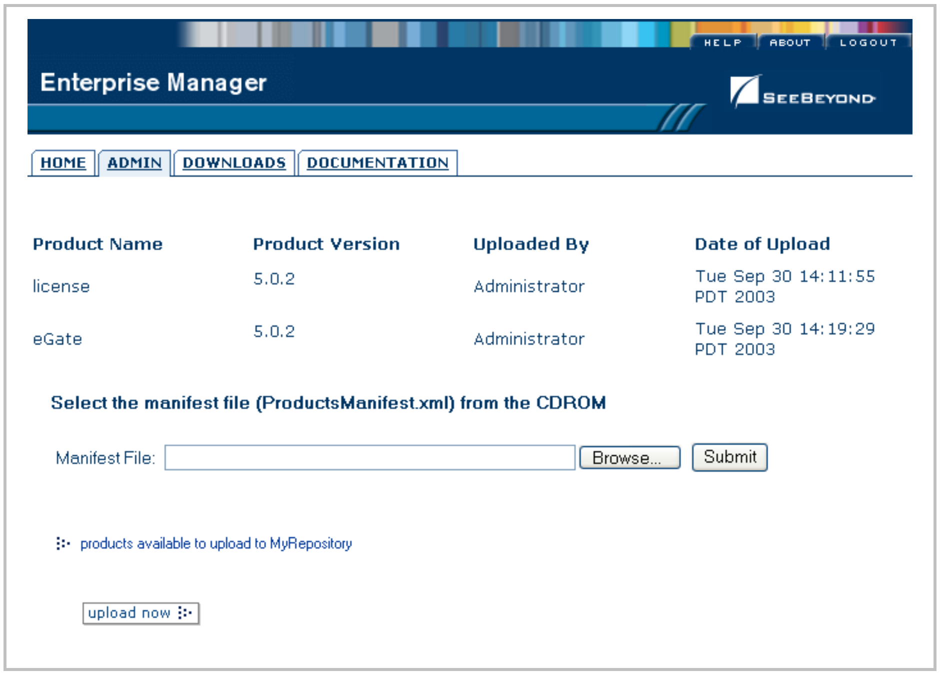
   http://<***localhost***>:<***portnumber***>

   where:

   localhost *is* the name of the machine running the Repository server and,

   portnumber is the access port you assigned to the Repository at installation time, for example: 12000.
3. On the Enterprise Manager **Login** page, type your 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	5.0.2	Administrator	Tue Sep 30 14:11:55 PDT 2003
eGate	5.0.2	Administrator	Tue Sep 30 14:19:29 PDT 2003

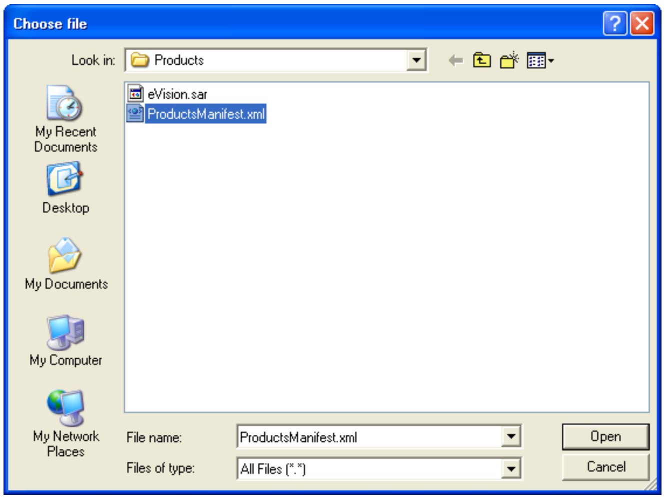
Select the manifest file (ProductsManifest.xml) from the CDROM

Manifest File:

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

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

Figure 4 Selecting the Products Manifest File



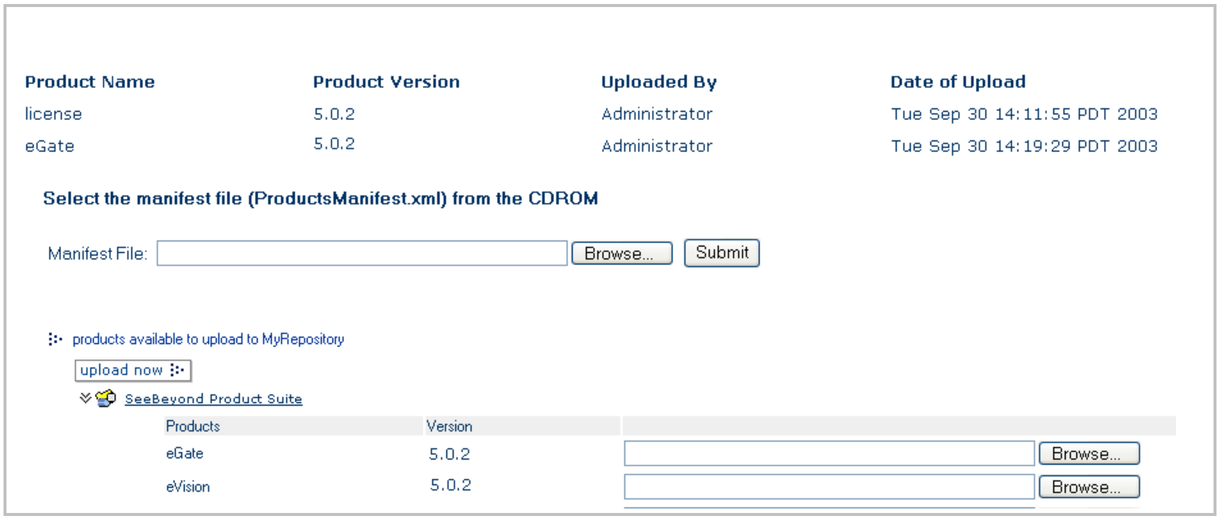
File name: ProductsManifest.xml

Files of type: All Files (*.*)

7 To see the list of products available for uploading, on the **ADMIN** page, click **Submit**.

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

Figure 5 ICAN Suite Products List



Product Name	Product Version	Uploaded By	Date of Upload
license	5.0.2	Administrator	Tue Sep 30 14:11:55 PDT 2003
eGate	5.0.2	Administrator	Tue Sep 30 14:19:29 PDT 2003

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

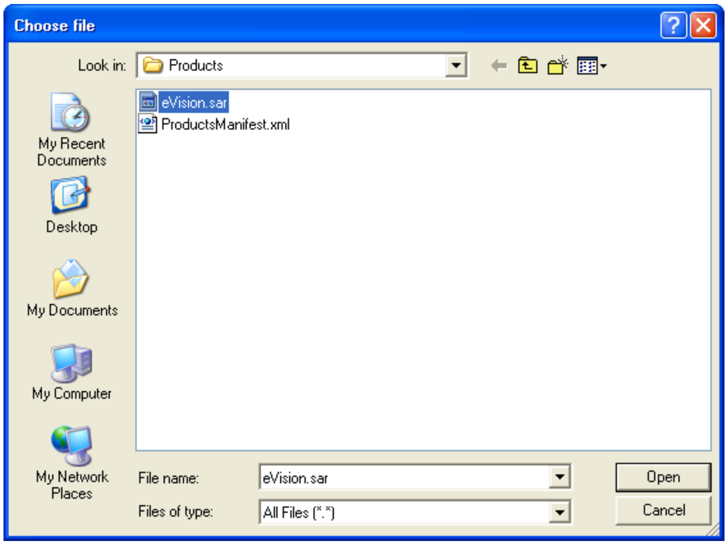
Manifest File:

Products	Version
eGate	5.0.2
eVision	5.0.2

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

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

Figure 6 eVision.sar File



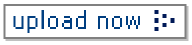
File name: eVision.sar

Files of type: All Files (*.*)

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

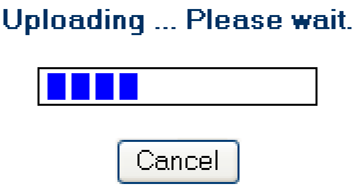
10 On the Enterprise Manager **ADMIN** page, click **Upload Now**. See **Figure 7 on page 24**.

Figure 7 upload Now Button



11 Watch the progress bar while the eVision archive file is uploaded. See Figure 8.

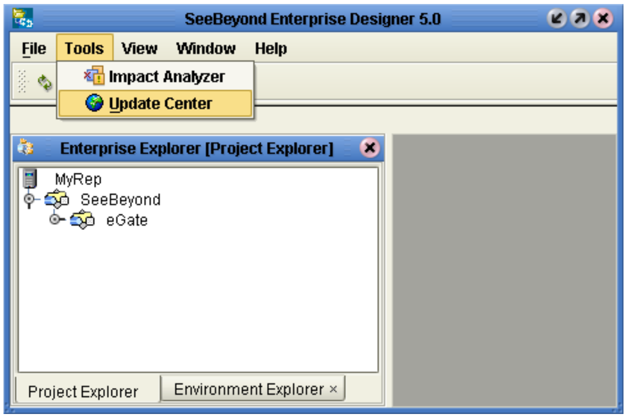
Figure 8 Product Upload Progress Bar



12 Once the upload is finished, start the Enterprise Designer.

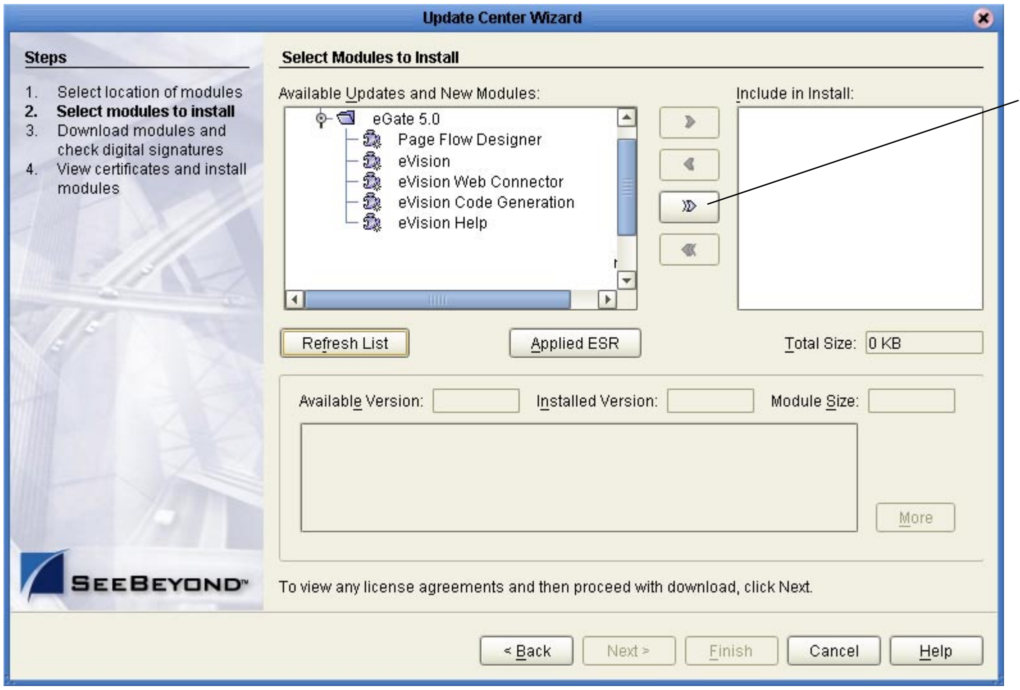
13 On the Enterprise Designer menu bar, click **Tools**, and then click **Update Center** to update the eVision Studio components. See Figure 9.

Figure 9 Updating eVision Studio



The **Update Center Wizard** appears with a list of components that are ready to be installed. See **Figure 10 on page 25**.

Figure 10 Update Center Wizard: Select Modules to Install



Add all

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

Select Modules to Install

Available Updates and New Modules:

- eGate 5.0
  - Page Flow Designer
  - eVision
  - eVision Web Connector
  - eVision Code Generation
  - eVision Help

Include in Install:

Total Size: 0 KB

Available Version: Installed Version: Module Size:

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

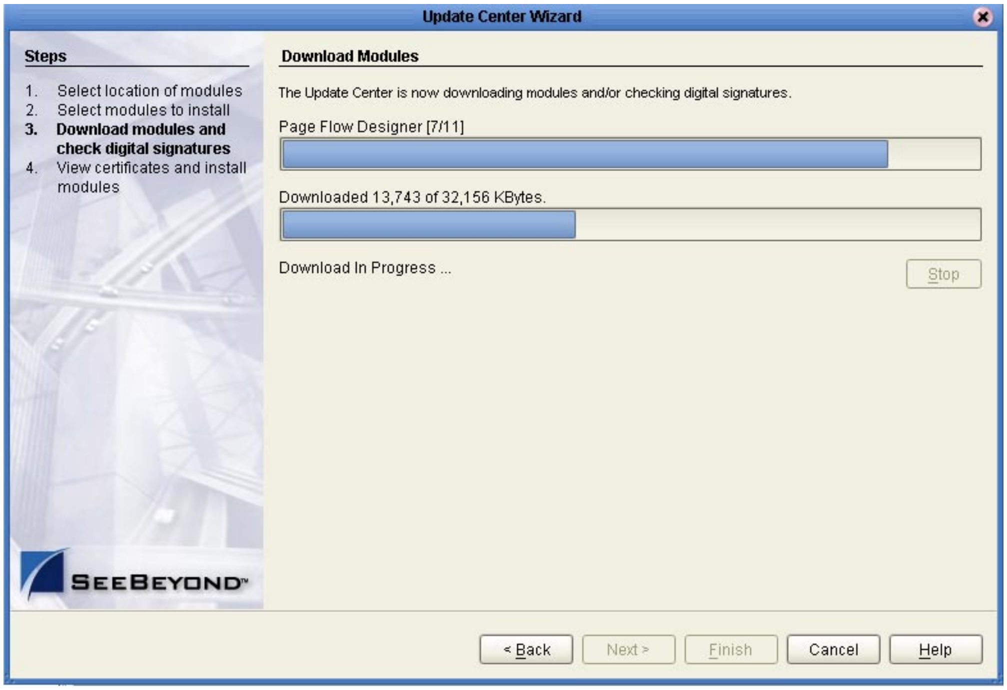
14 Click the **Add all** button to move the eVision components to the **Include in Install** window.

15 Click **Next.**

16 To accept the license agreement, on the **License Agreement** window, click **Accept**.

Watch the progress bars as the files are downloaded. See **Figure 11 on page 26**.

Figure 11 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.

Page Flow Designer [7/11]

Downloaded 13,743 of 32,156 KBytes.

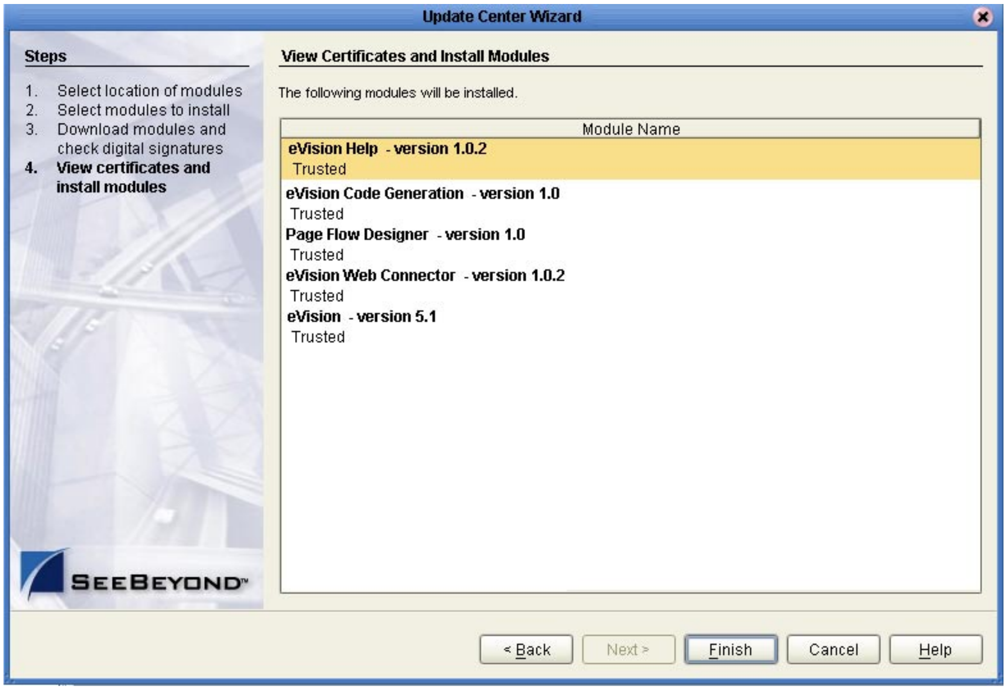
Download In Progress ...

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

18 Click **Next**.

19 The **Update Center Wizard** displays the list of certificates and installed modules. See **Figure 12 on page 27**.

Figure 12 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
SEEBEYOND
< Back
Next >
Finish
Cancel
Help

View Certificates and Install Modules

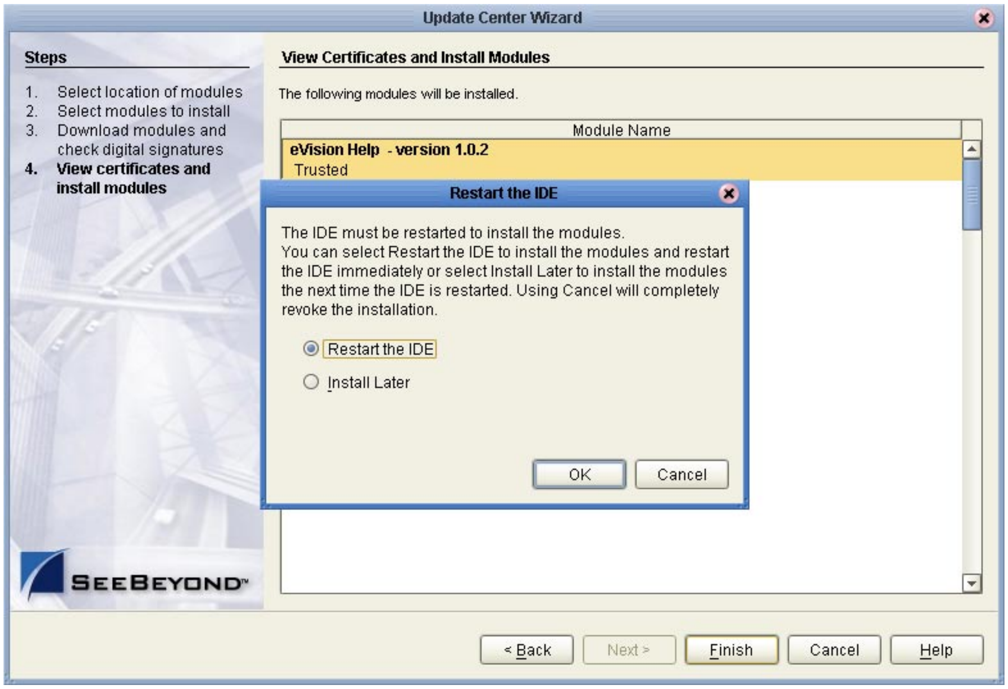
The following modules will be installed.

Module Name
eVision Help - version 1.0.2 Trusted
eVision Code Generation - version 1.0 Trusted
Page Flow Designer - version 1.0 Trusted
eVision Web Connector - version 1.0.2 Trusted
eVision - version 5.1 Trusted

20 Before continuing, briefly review what you have just installed, and then click **Finish**.

21 When the Update Center Wizard prompts you to do so, restart the Enterprise Designer. See **Figure 13 on page 28**.

Figure 13 Update Center Wizard: Restart the 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
eVision Help - version 1.0.2 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 the 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:

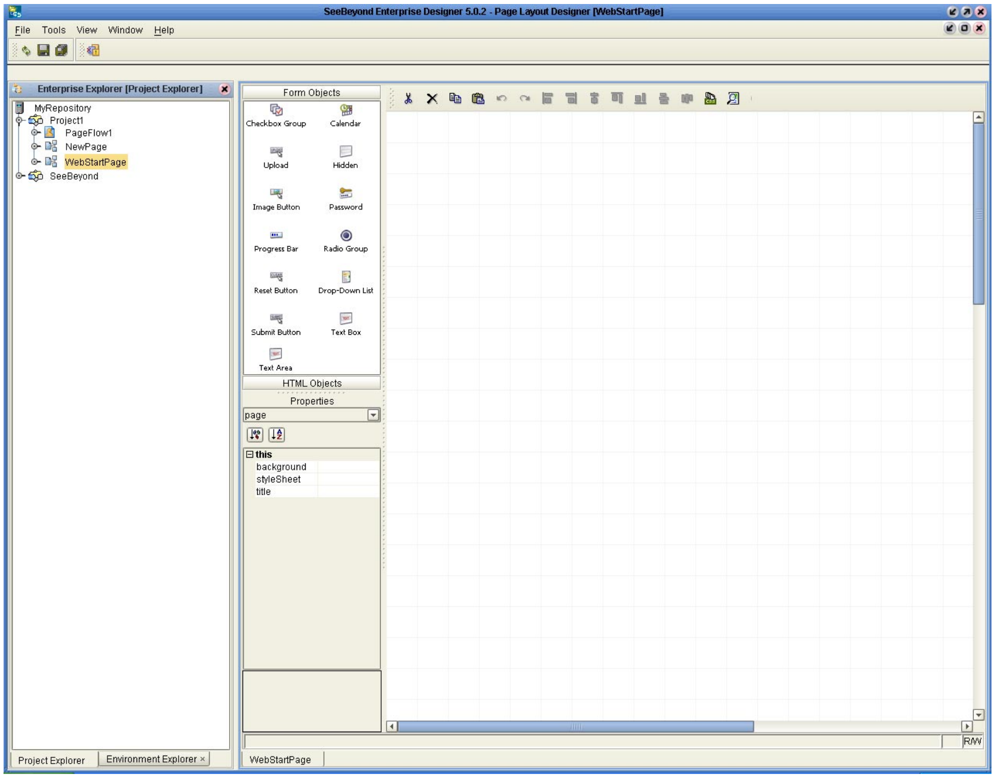
- **"Design Tools"**
- **"Graphical User Interface Components" on page 32**
- **"GUI Component Selection Palettes" on page 33**
- **"Defining Component Properties" on page 44**
- **"Customizing a Style Sheet" on page 51**
- **"Importing a Style Sheet Into Your Project" on page 53**
- **"Linking Pages" on page 58**

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 a Web page. 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, the **Page Layout Designer** with the **Form Objects** palette active is shown in **Figure 14 on page 30**.

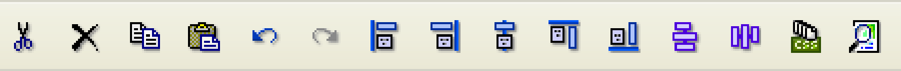
Figure 14 The Page Layout Designer Canvas at Startup



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 15. For a description of these tools, see **“eVision Studio Tools” on page 31**.

Figure 15 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 then **continues backward up to the last time the layout was saved**. To enhance performance, you may wish to disable the Undo function. See **"Disabling the Undo Function" on page 32**.
↷	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 three or more, based on an averaging algorithm.

Tool	Name	Purpose
	Vertical Spacing	Creates equal vertical spacing between objects in a group of three or more, based on an averaging algorithm.
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.

7.1.3 Disabling the Undo Function

To enhance eVision design-time performance, you may wish to disable the **Undo** function.

To Disable the Undo Function

1. On your computer, explore to:

   <*drive*>:\<*directory*>\edesigner\bin,

   where:

   drive is the drive letter and,

   directory is the directory where the Enterprise Designer is installed.

2. Edit the **runed.bat** file.

3. Locate the following line:

   ```
   runidew -J-DNETBEANS_HOME=%NETBEANS_HOME%
   ```

4. Insert the following argument after `%NETBEANS_HOME%`:

   ```
   -J-DeVisionUndo=false
   ```

   For example:

   ```
   runidew -J-DNETBEANS_HOME=%NETBEANS_HOME%-J-DeVisionUndo=false
   ```

5. Save **runed.bat**

If you want to re-enable the **Undo** function, edit **runed.bat** and delete the `-J-DeVisionUndo=false` argument.

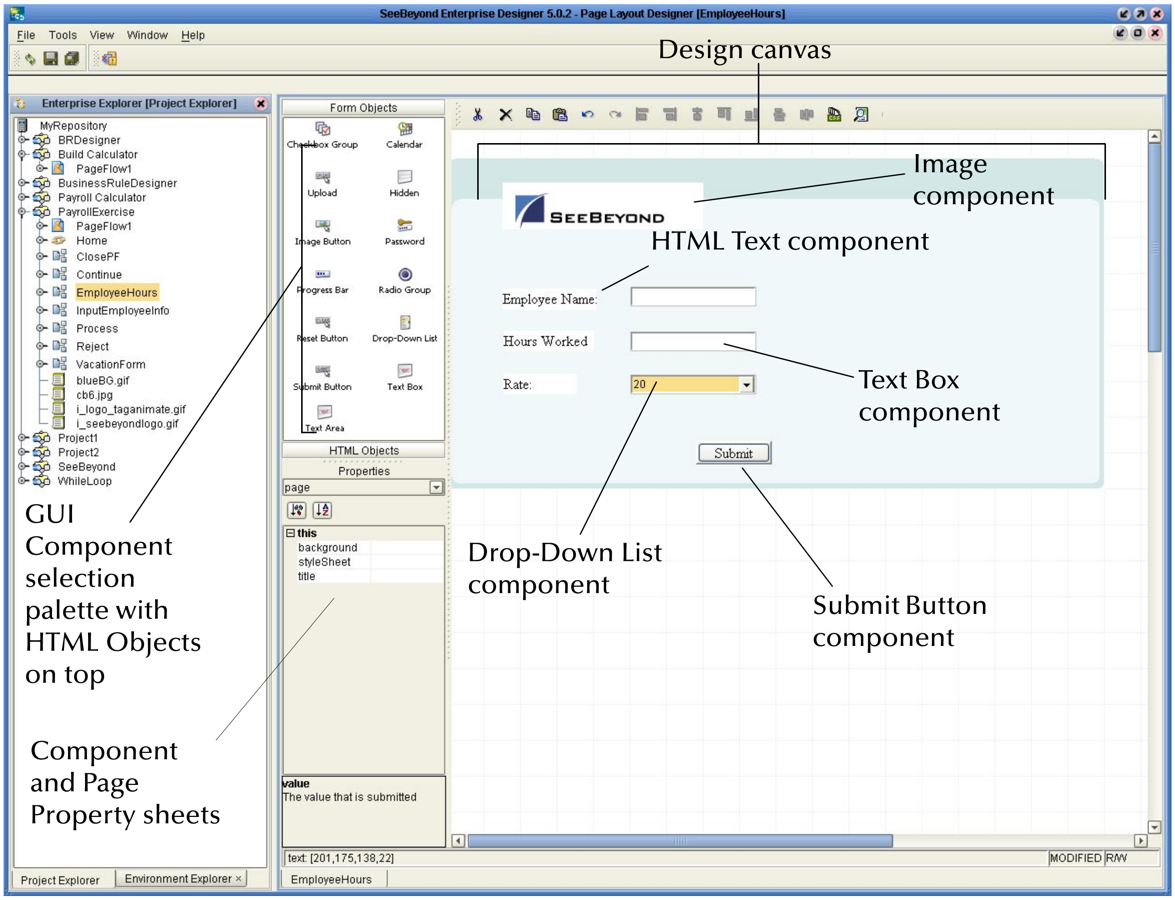
7.2 Graphical User Interface Components

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, submit buttons, etc.

GUI components are pre-built combinations of Java classes and JSP code that represent Web interface elements. When you place a GUI component on the canvas, 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 the user in a browser.

In the Page Layout Designer, the upper left pane contains the Graphical User Interface (GUI) Component libraries. See Figure 16.

Figure 16 Page Layout Designer GUI Components



value
The value that is submitted

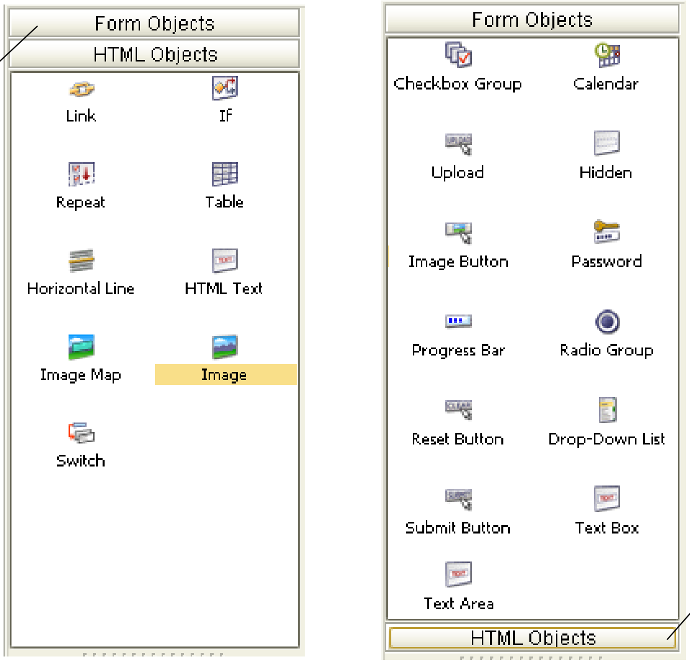
text: [201,175,138,22]

7.3 GUI Component Selection Palettes

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

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 17.

Figure 17 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 58**.
	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.

Component	Name	Purpose
	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, up or down). 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. 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.
	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 38**.
	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 on dynamic tables, see **"About Dynamic Tables" on page 36**.
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.

Component	Name	Purpose
🖼	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 41**

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, select **Set as Dynamic Row**.

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

Adding Pagination to a Dynamic Table

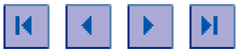
You can add a “page-forward, page-back”, “page-first, page-last” function to a row or rows in a dynamic table. The Pagination feature allows table cell data to be incrementally displayed; for example: If the table is set to read the results of a database query, the table will display the first 10 items returned by query, and allow the user to page forward to the second 10 items, and so on.

Before you can use this feature, you must import the pagination images:

- paginatefirst.gif
- paginatelast.gif
- paginatenext.gif
- paginateprevious.gif

See **Figure 18 on page 37**.

Figure 18 Pagination buttons

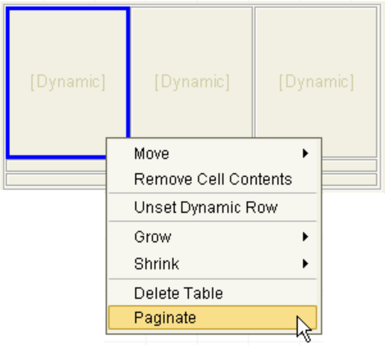


For detailed instructions on importing images, see **"Importing Images" on page 41**.

To Add Pagination to a Row in a Dynamic Table

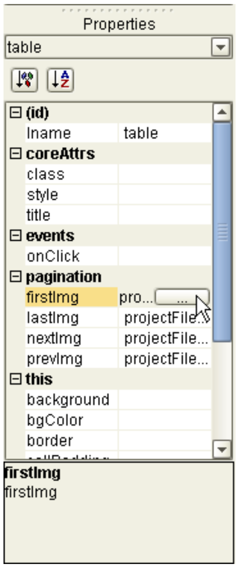
1. Click and drag a bounding box around the table.
2. Right-click the mouse, and on the context menu, click **Paginate**. See Figure 19.

Figure 19 Paginate Menu



3. In the table Properties sheet, select a pagination button (image) for each pagination property. See Figure 20.

Figure 20 Pagination Properties



Properties

table

⊟ **(id)**	
lname	table
⊟ **coreAttrs**	
class	
style	
title	
⊟ **events**	
onClick	
⊟ **pagination**	
firstImg	pro... ...
lastImg	projectFile...
nextImg	projectFile...
prevImg	projectFile...
⊟ **this**	
background	
bgColor	
border	

firstImg
firstImg

4. Save the Page.

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 41**.

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.

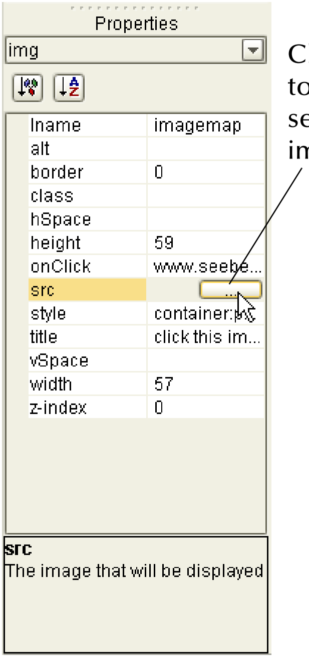
   When the icon transfers onto the canvas it changes into a dashed-line rectangle.

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 21.

Figure 21 Image Map Properties Sheet



Properties

img

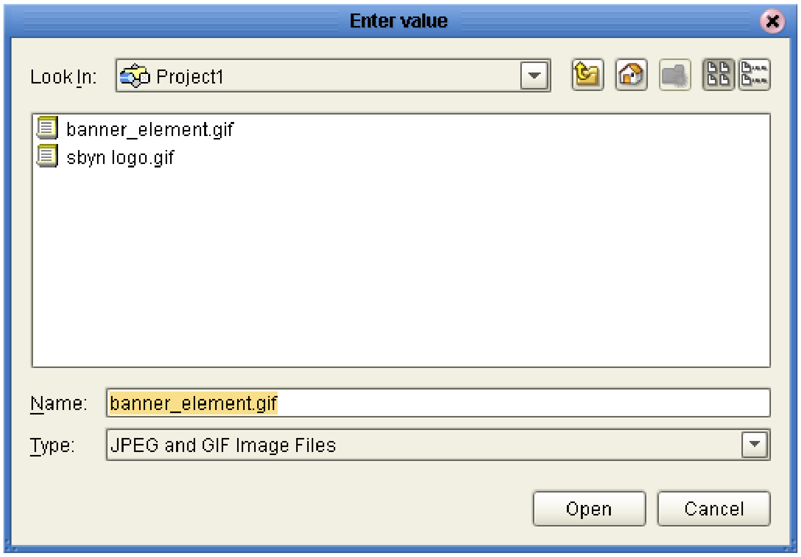
Iname	imagemap
alt	
border	0
class	
hSpace	
height	59
onClick	www.seebe...
src	...
style	container:p...
title	click this im...
vSpace	
width	57
z-index	0

src
The image that will be displayed

Click to explore to the image to select for the image map

The **Enter value** dialog box appears. See Figure 22.

Figure 22 Selecting an Image for an Image Map



Name: banner_element.gif

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 41**.

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**).

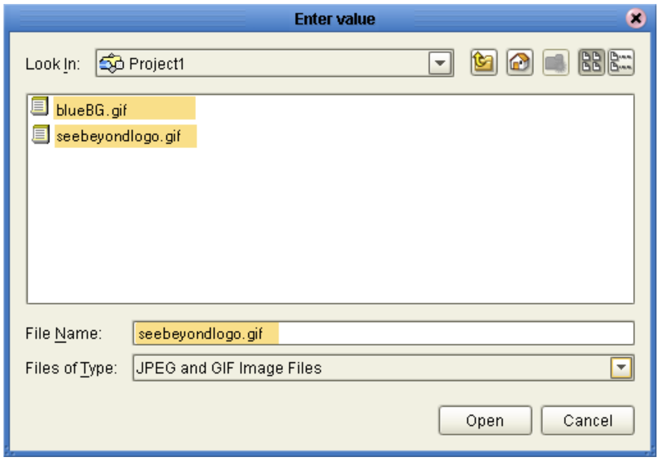
lname is the internal, logical name for the component, which identifies the component in the Page Flow.

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 23.

Figure 23 Searching the Project for an Image File



File Name: seebeyondlogo.gif

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 41** *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.

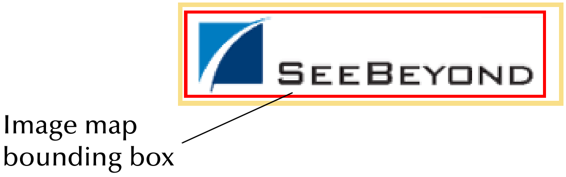
Note: *An image map holds a link area. The link area enables you to link-edit and attach a target destination to the image map.*

9 Modify the image map area, if necessary. See **Figure 24 on page 41**.

- 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.

Note: *On small images, it is easier to manually enter the coordinates in the Property Sheet.*

Figure 24 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 on disk.

To Import a Graphic Image

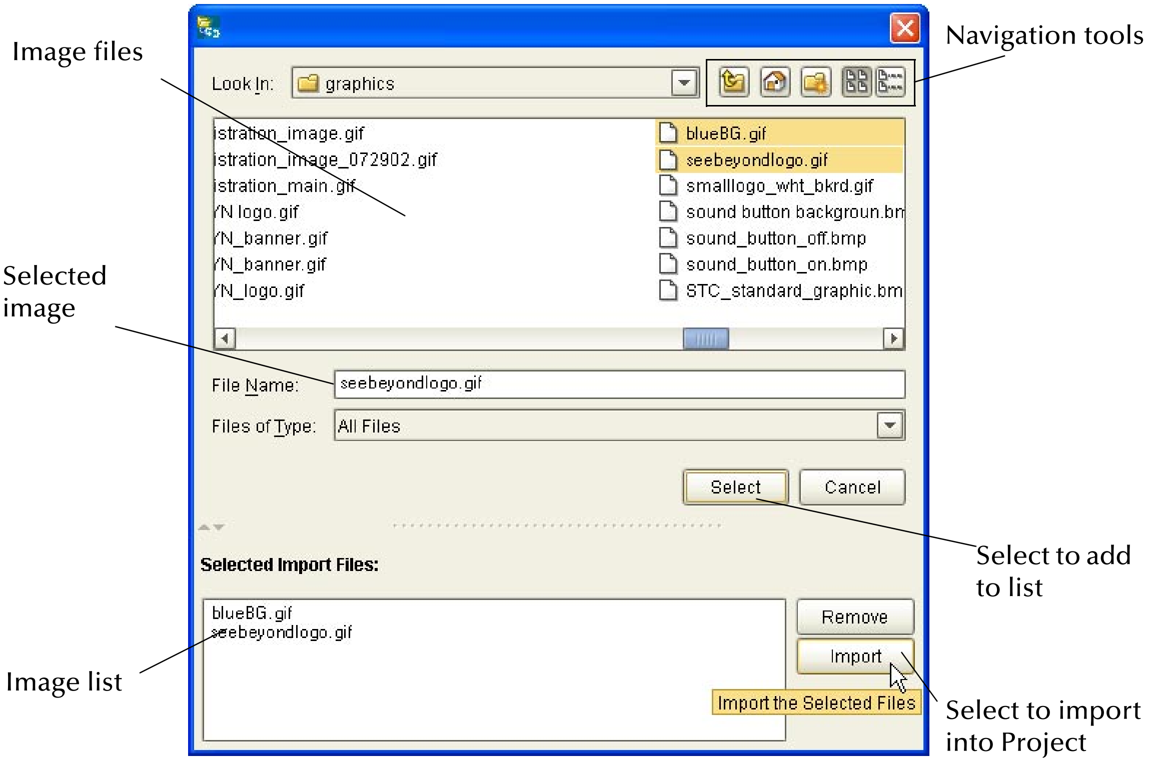
1 In the **Enterprise Explorer**, right-click the **Project** icon.

2 On the context menu, select **New**, and then select **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 25 on page 42**.

Figure 25 Importing Images into a Project



Look In: graphics

istration_image.gif
istration_image_072902.gif
istration_main.gif
'N logo.gif
'N_banner.gif
'N_banner.gif
'N_logo.gif
blueBG.gif
seebeyondlogo.gif
smalllogo_wht_bkrd.gif
sound button backgroun.bm
sound_button_off.bmp
sound_button_on.bmp
STC_standard_graphic.bm

File Name: seebeyondlogo.gif

Files of Type: All Files

Selected Import Files:

blueBG.gif
seebeyondlogo.gif

Image files

Navigation tools

Selected image

Select to add to list

Image list

Select to import into Project

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

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

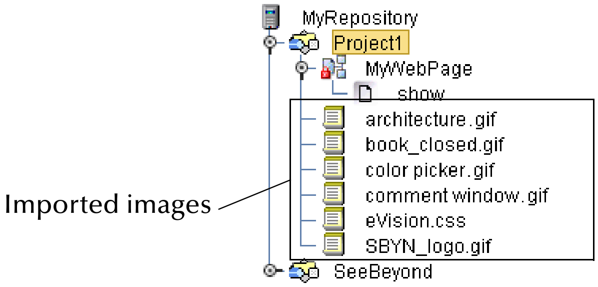
6 Select additional images, if desired.

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 26 Imported Images in Enterprise Explorer



MyRepository
Project1
MyWebPage
show
architecture.gif
book_closed.gif
color picker.gif
comment window.gif
eVision.css
SBYN_logo.gif
SeeBeyond

Imported images

The images will be displayed under your Project in the **Enterprise Explorer** tree. See Figure 26.

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. You can create multiple check boxes within a group.
	Calendar	Adds a calendar under a drop-down arrow. On a Web page, selecting a date on the calendar object populates an attached text box with the selected date.
	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.
■■■	Progress Bar	Creates a progress bar that allows the user to monitor the progress of a particular operation on a Web page.
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. You can create multiple buttons in a group. ***Note:*** *The Value property cannot be empty. This object requires that a value be assigned in order to pass the true/false condition.*

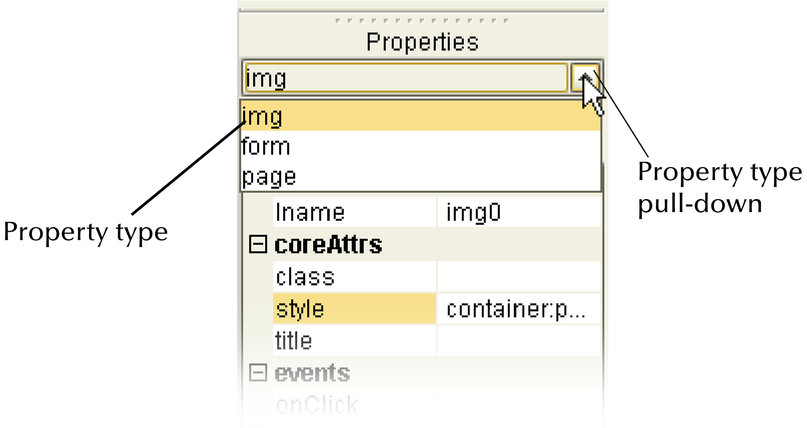
Component	Name	Purpose
	Drop-Down List	Allows the user to select an entry from a drop-down list. You can create multiple entries on a list.
SUBMIT	Submit Button	Allows the user to launch an operation or submit text to the application. Passes accumulated values to the ICAN system.
TEXT	Text Box	Allows the user to type text that will be displayed on the Web page surrounded by a bounding box.

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 27.

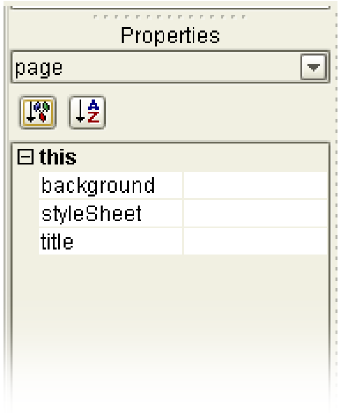
Figure 27 Properties Drop-down Menu



lname	img0
class	
style	container:p...
title	

For example, when you click the Page entry from the Properties drop-down, the Page Properties sheet pops to the front. See Figure 28 on page 45.

Figure 28 New Page Properties



Properties

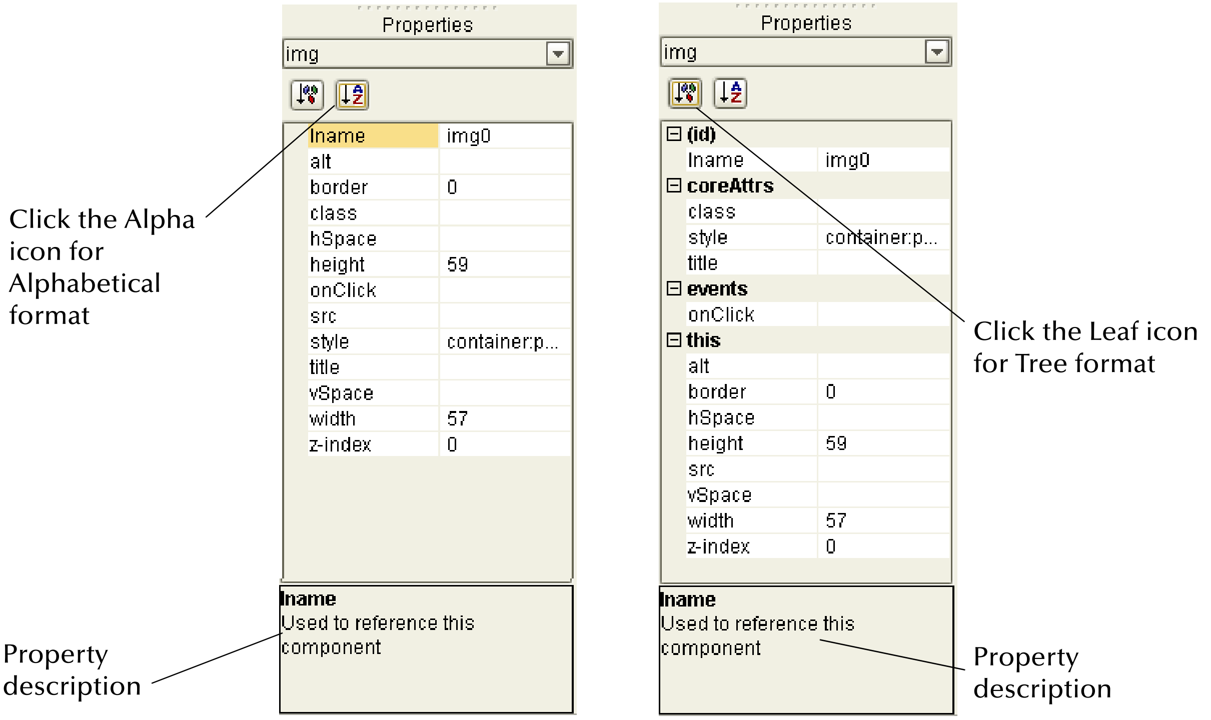
page

⊟ **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 29.

Figure 29 Swapping Component Properties Formats



Click the Alpha icon for Alphabetical format

Property description

Click the Leaf icon for Tree format

Property description

Properties

img

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

Properties

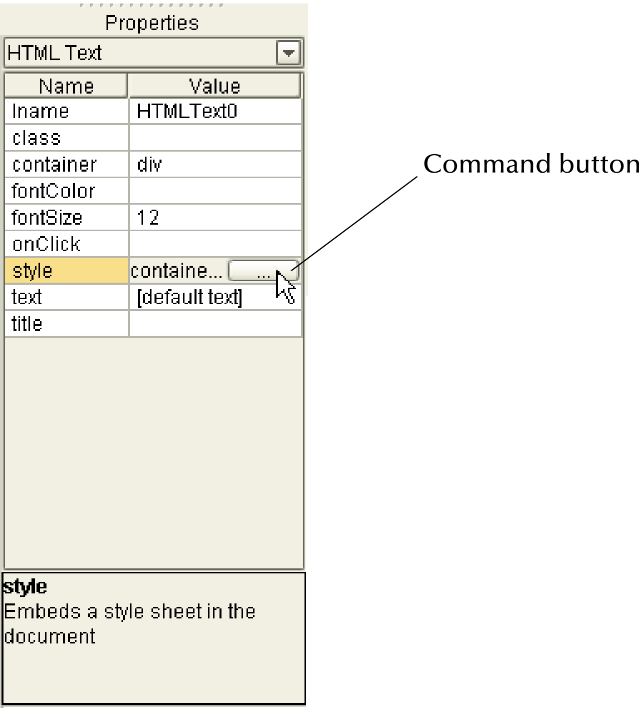
img

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

lname
Used to reference this component

To define a property, select any field 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.

Figure 30 Command Button

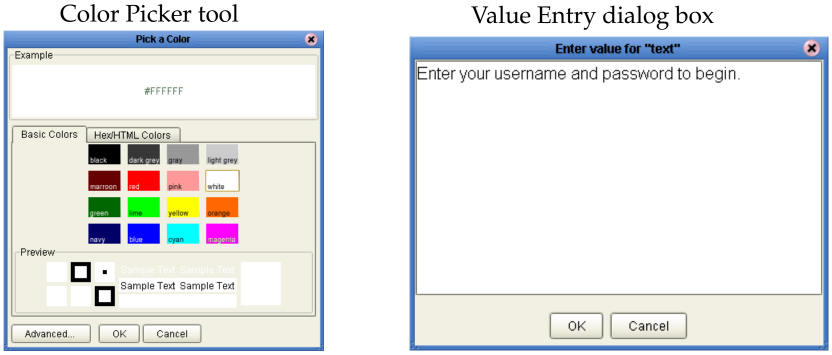


Name	Value
Iname	HTMLText0
class	
container	div
fontColor	
fontSize	12
onClick	
style	containe... ...
text	[default text]
title	

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 31.

Figure 31 Interface Elements Displayed Via a **Command** Button



Color Picker tool

Value Entry dialog box

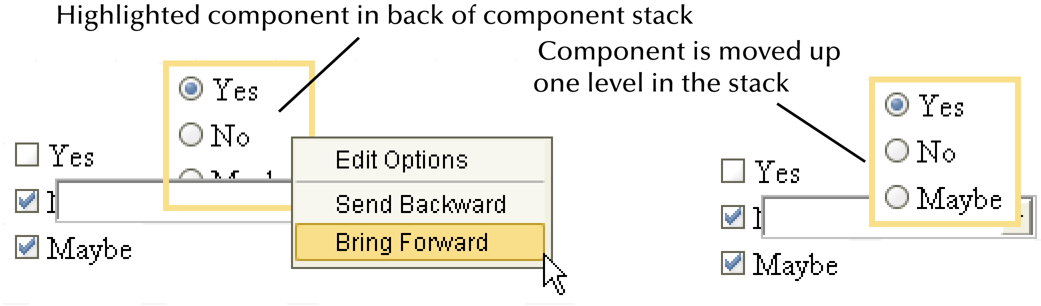
Enter your username and password to begin.

7.4.1 Moving Overlapping Components in the Z-Direction

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

1. Highlight the component you want to move.
2. Place the cursor inside the component's highlight boundary and right-click the mouse.
3. On the context menu, select **Bring Forward**, or **Send Backward** as appropriate. See Figure 32.

Figure 32 Moving Components in the Z-direction



Highlighted component in back of component stack

Component is moved up one level in the stack

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

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

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

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

- To move an image down (back) one or more levels, click the **z-index** field on the property sheet, and type a negative integer; for example: **-9**.
- To move an image up (forward) one or more levels, click the **z-index** field on the property sheet and type a positive integer; for example **+7**.

All GUI components can be moved up or down in the stack order.

*Note: The **Checkbox Group**, **Radio Group**, and **Drop-Down List** components will also have **Edit Options** on the context menu.*

7.4.2 Adding Multiple Options to Form Object Components

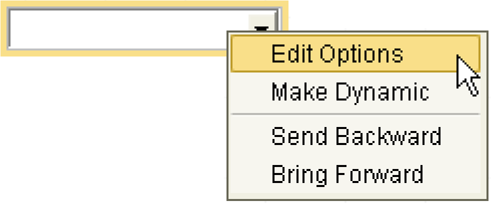
The **Checkbox Group**, **Radio Group**, and **Drop-Down List** components allow you to create multiple user-selectable options in a group, or list.

These components can also be made static or dynamic. Dynamic components are extended or replicated by external sources.

Creating Multiple Entries in a Drop-Down List

1. Drag the **Drop-Down List** component onto the canvas.
2. Right-click the component.
3. On the context menu, select **Edit Options**. See Figure 33.

Figure 33 Editing Component Options



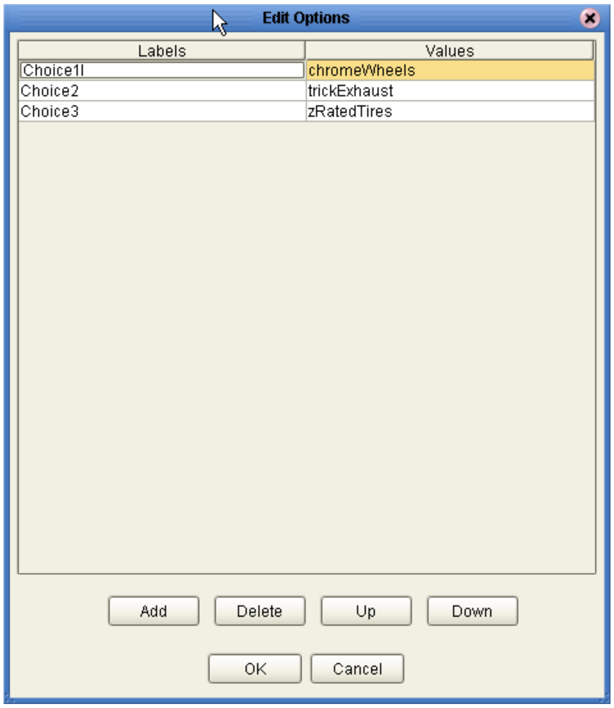
The **Edit Options** dialog box appears.

4. In the **Edit Options** dialog box, click **Add**.
5. Select a **Label** field and type a label.

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

   The value will be submitted when the user clicks the label at run-time.
7. Repeat the previous steps to create additional **Drop-Down List** entries. See **Figure 34 on page 49**.

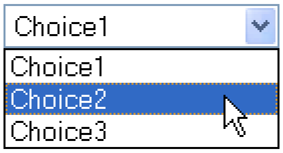
Figure 34 Edit Options Dialog Box



Labels	Values
Choice1	chromeWheels
Choice2	trickExhaust
Choice3	zRatedTires

8 To see how the list will be presented to users at run-time, click the **Preview** icon. See Figure 35.

Figure 35 Drop-Down List Preview



9 To make the **Drop-Down List** component dynamic, right-click the component.

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

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

Creating a Group of Radio Buttons

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

2 Right-click the component.

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

4 In the **Edit Options** dialog box, click **Add**.

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

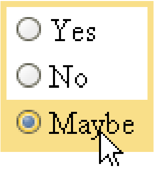
The label will be displayed next to the button at run-time.

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

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

7 Repeat the previous step to create additional buttons in the group. See Figure 36.

Figure 36 Editing Radio Button Group Options



8 To make the **Radio Button** component dynamic, drag it into a table cell and make the host cell dynamic.

At run-time, a dynamic table cell can be modified by external sources to generate additional cells containing radio buttons.

Creating a Group of Check Boxes

1 Drag the **Check Box Group** component onto the canvas.

2 Right-click the component.

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

The **Edit Options** dialog box appears.

4 In the **Edit Options** dialog box, click **Add**.

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

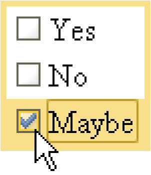
The label will be displayed next to the check box at run-time.

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

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

7 Repeat the previous steps to add additional check boxes to the group. See Figure 37.

Figure 37 Editing Check Box Group Options



8 To make the **Check Box Group** component dynamic, drag it into a table cell and make the host cell dynamic.

At run-time, a dynamic table cell can be modified by external sources to generate additional cells containing check boxes.

7.5 Customizing a Style Sheet

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

eVision Studio provides two style sheets for immediate use: *eVision-default.css* and *eVision.css*. To apply a style sheet, see **"Applying a Style Sheet to Your Web Application" on page 55**.

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

If you do not have a preferred style sheet that you want 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.

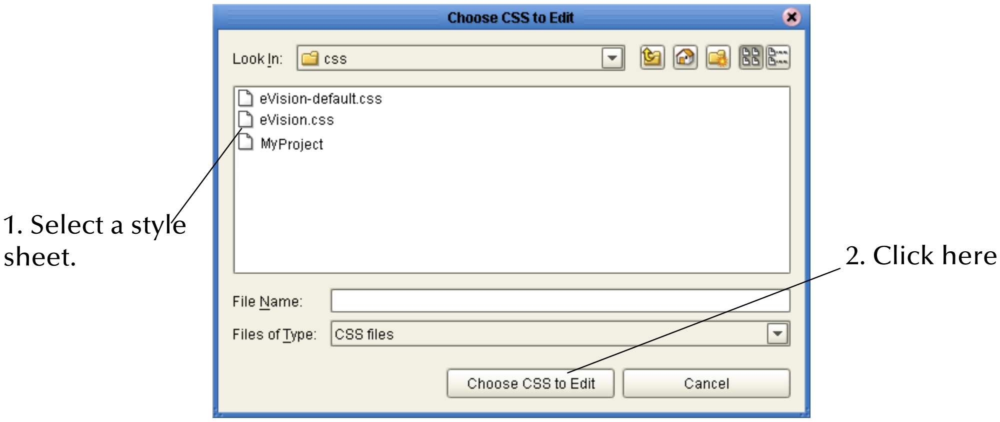
Creating a new style sheet from eVision.CSS

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 (recommended):

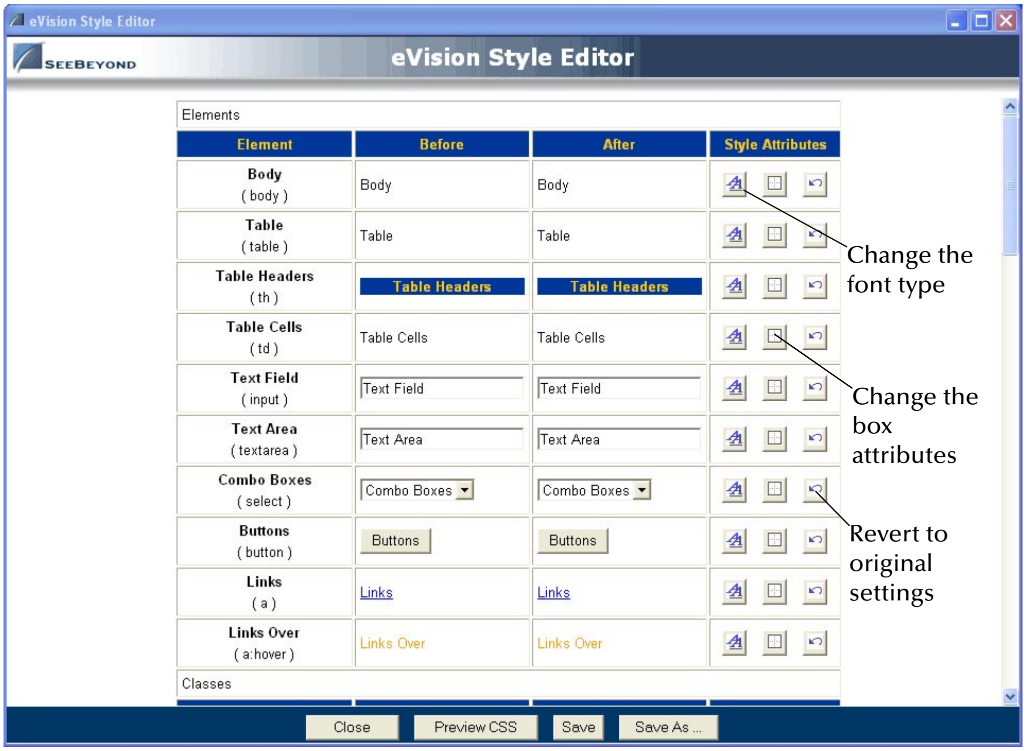
1. On the eVision toolbar, click the **CSS** icon.
2. On the **Choose CSS to Edit** dialog box, click **eVision.css**
3. Click **Choose CSS to Edit**. See Figure 38.

Figure 38 Choosing a Style Sheet to Edit



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 39.

Figure 39 eVision Style Editor



Elements

Element	Before	After	Style Attributes
Body (body)	Body	Body	A □ ↶
Table (table)	Table	Table	A □ ↶
Table Headers (th)	Table Headers	Table Headers	A □ ↶
Table Cells (td)	Table Cells	Table Cells	A □ ↶
Text Field (input)	Text Field	Text Field	A □ ↶
Text Area (textarea)	Text Area	Text Area	A □ ↶
Combo Boxes (select)	Combo Boxes ▾	Combo Boxes ▾	A □ ↶
Buttons (button)	Buttons	Buttons	A □ ↶
Links (a)	Links	Links	A □ ↶
Links Over (a:hover)	Links Over	Links Over	A □ ↶

Classes

Change the font type

Change the box attributes

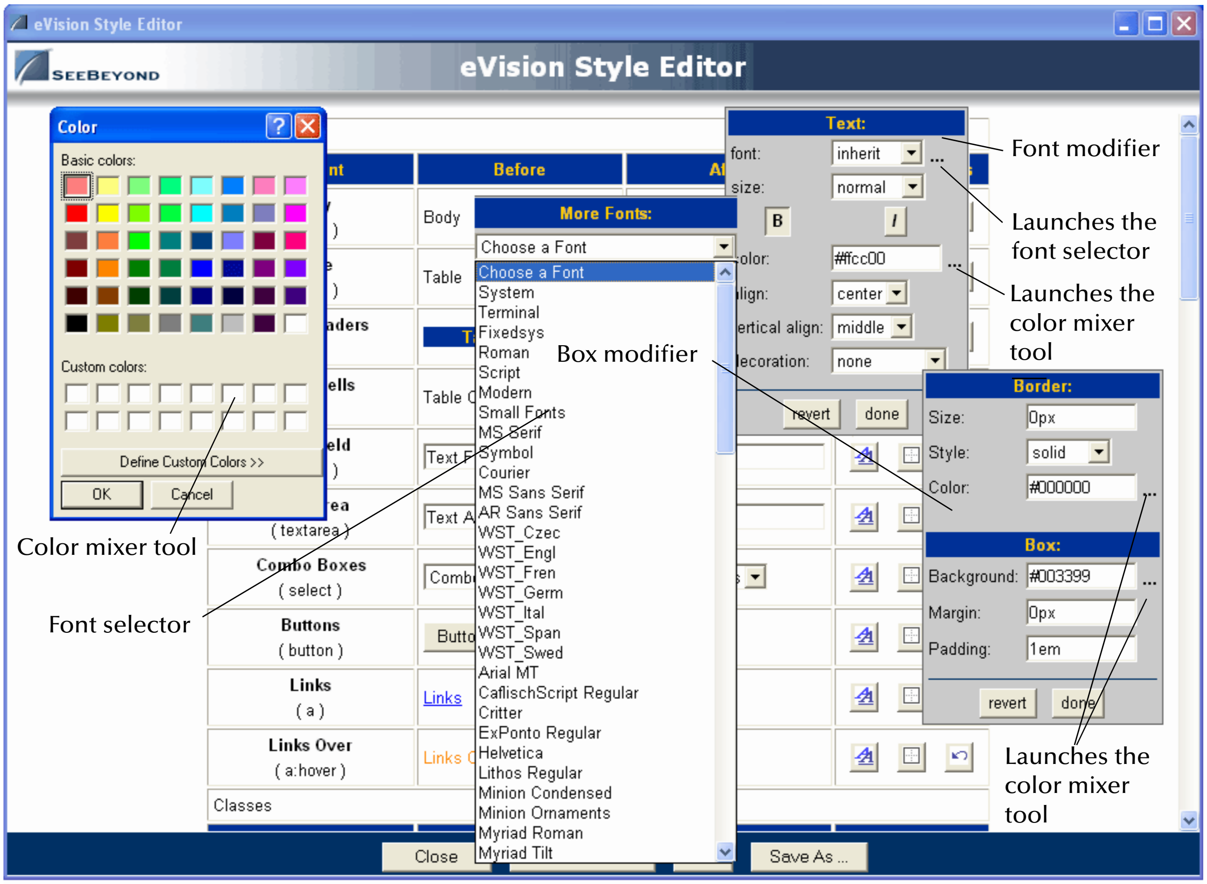
Revert to original settings

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 40 on page 53**.

Figure 40 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

More Fonts:
Choose a Font

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

7.6 Importing a Style Sheet Into Your Project

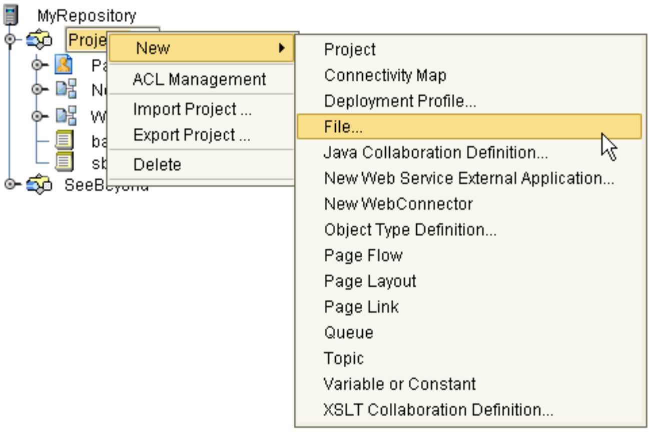
When you are satisfied with the element configuration in your style sheet, you must import it into your Project before it can be applied.

To Import a Custom Style Sheet

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

2 On the context menu, click **New**, and then select **New File**. See Figure 41.

Figure 41 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:\ICAN50***, 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:

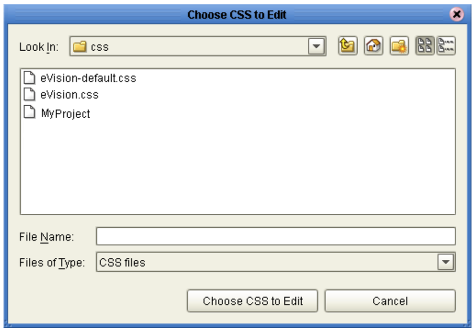
usrdir > modules > ext > stc > evision_core > tigris > css

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

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

8 On the **Choose CSS to Edit** dialog box, click **Choose CSS to Edit**, and then click **Import**. See Figure 42.

Figure 42 CSS Chooser Dialog Box



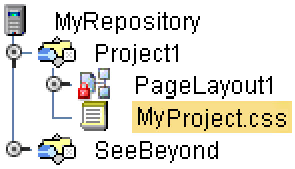
Look In: css

eVision-default.css
eVision.css
MyProject

Files of Type: CSS files

The style sheet(s) will appear in your Project in **Enterprise Explorer**. See **Figure 43 on page 55**.

Figure 43 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.

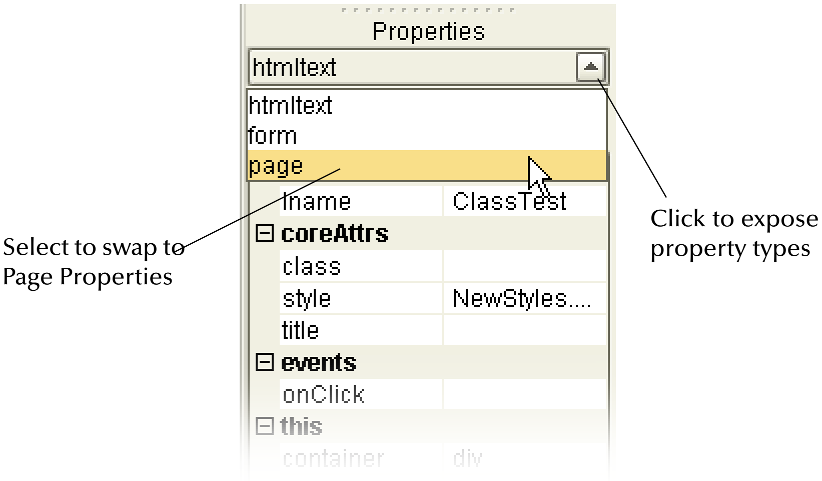
7.6.1 Applying a Style Sheet to Your Web Application

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**.

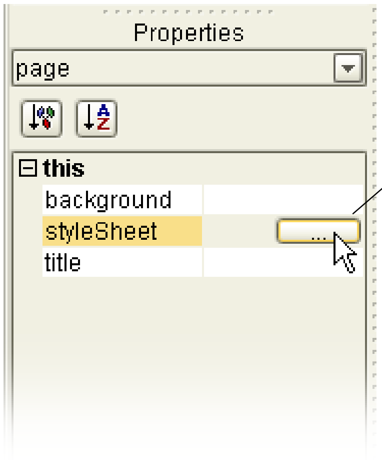
   The **Page Properties** window will come to the front. See Figure 44.

Figure 44 Changing the Properties Window



3. To launch the **Style Sheet Explorer** dialog box, click the **Command** button (**...**) in the **Page Properties** window. See **Figure 45 on page 56**.

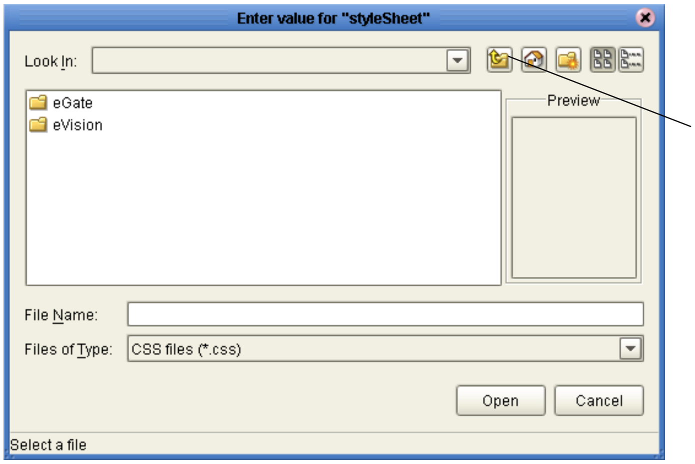
Figure 45 Exploring for a Style Sheet



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 directory. See Figure 46.

Figure 46 Moving Up a Level in the CSS Explorer

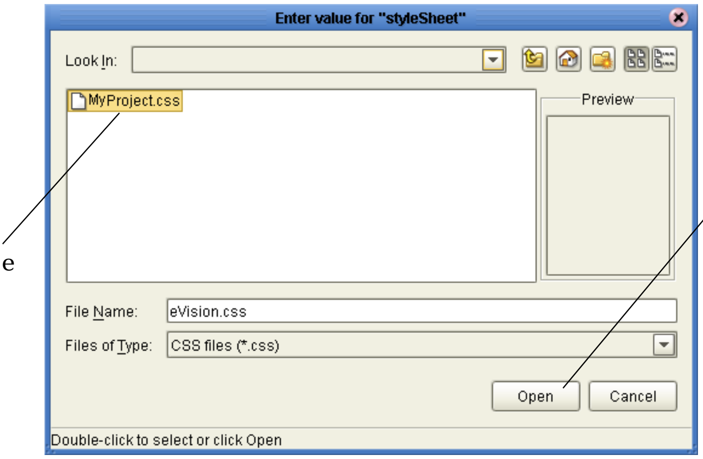


Click to move up one level in the directory

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 47 on page 57.

Figure 47 Selecting a Style Sheet to Import Into the Project



Look In:

MyProject.css

File Name: eVision.css

Files of Type: CSS files (*.css)

Double-click to select or click Open

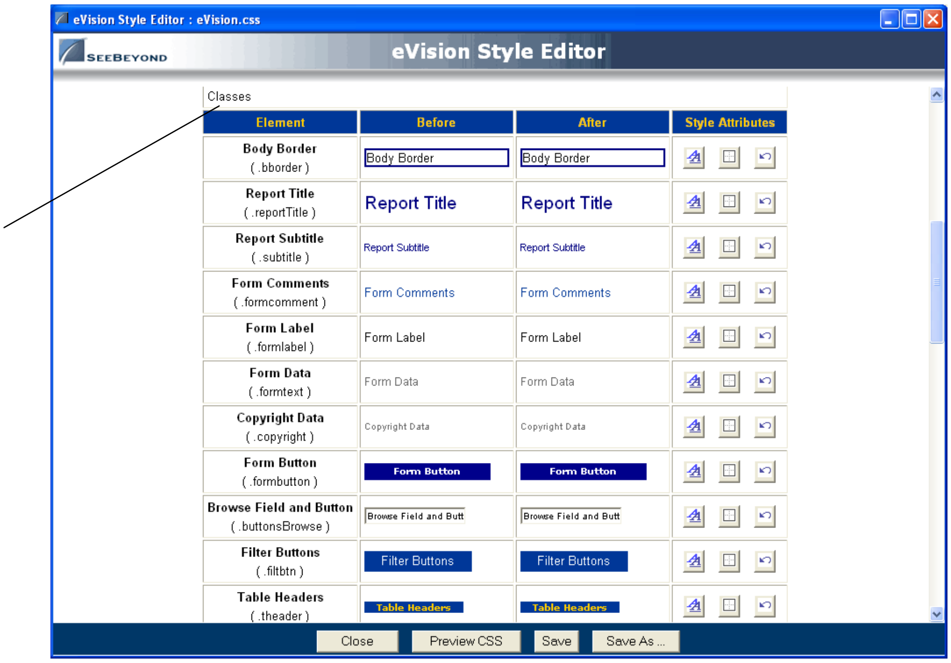
Select imported style sheet file

Click Open

7.6.2 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 48.

Figure 48 Style Sheet with Classes



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	

Classes heading

You can modify classes when you create or modify a style sheet. Classes cannot be applied until the style sheet is imported into the Project. (See **“Importing a Style Sheet**

Into Your Project" on page 53.) After importing, follow the instructions in **"Applying a Class to an Object"**.

Applying a Class to an Object

To apply a class to 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.

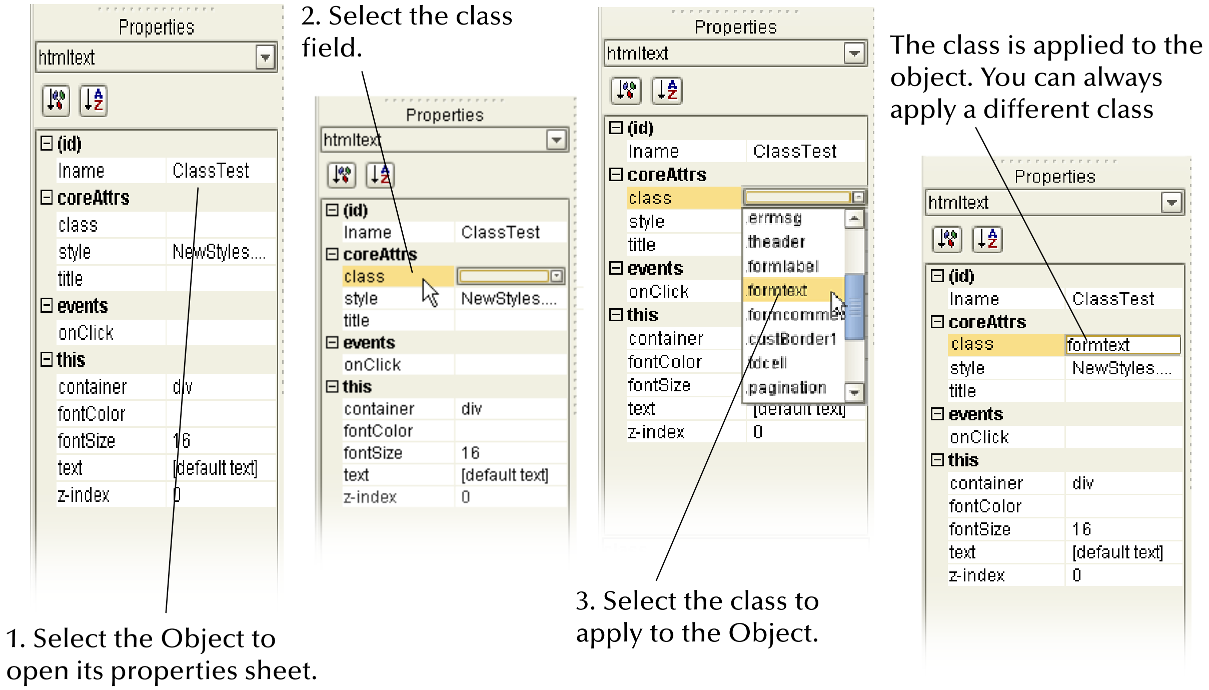
   The **Component Properties** sheet will open showing the object's properties.

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 49.

Figure 49 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

7.7 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

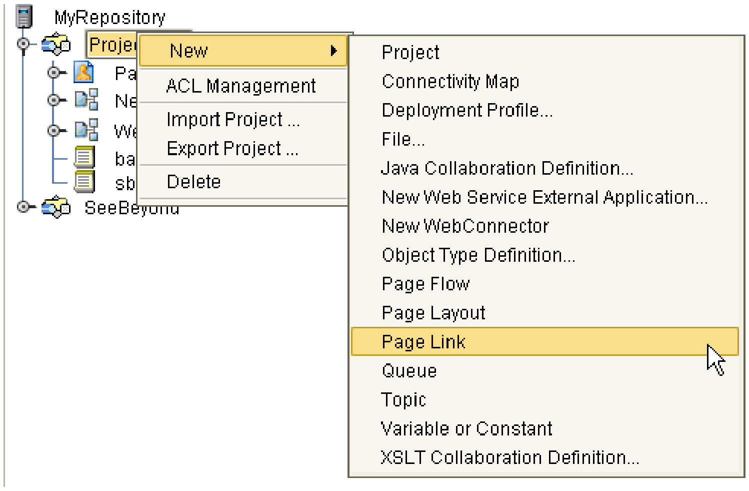
- Create text links that connect Web pages and other resources to the Page Flow
- Specify parameters to pass to other objects on destination pages

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.

To Create an Application Page Link

1. In **Enterprise Explorer**, right-click your **Project** icon.
2. On the context menu, click **New**, and then select **Page Link**. See Figure 50.

Figure 50 Creating a Page Link



The **Page Link Wizard** appears. See **Figure 51 on page 60**.

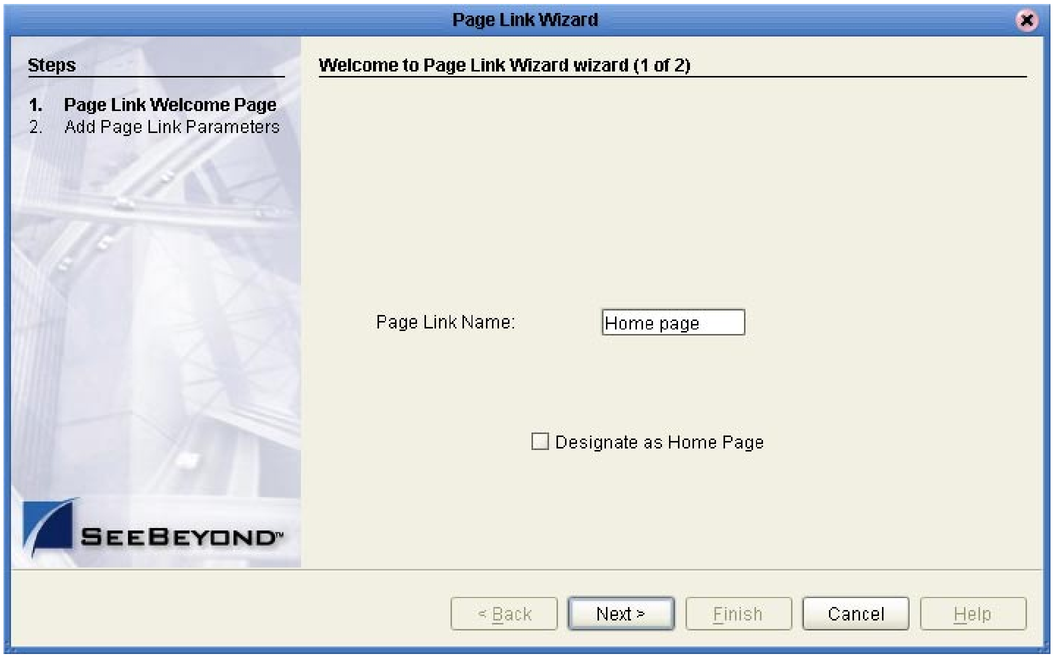
3. On the **Page Link Wizard**, in the **Page Link Name** field, type a descriptive name for the link.

   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 51 Page 1 of 2 in the Page Link Wizard



Page Link Wizard

Steps

1. Page Link Welcome Page
2. Add Page Link Parameters

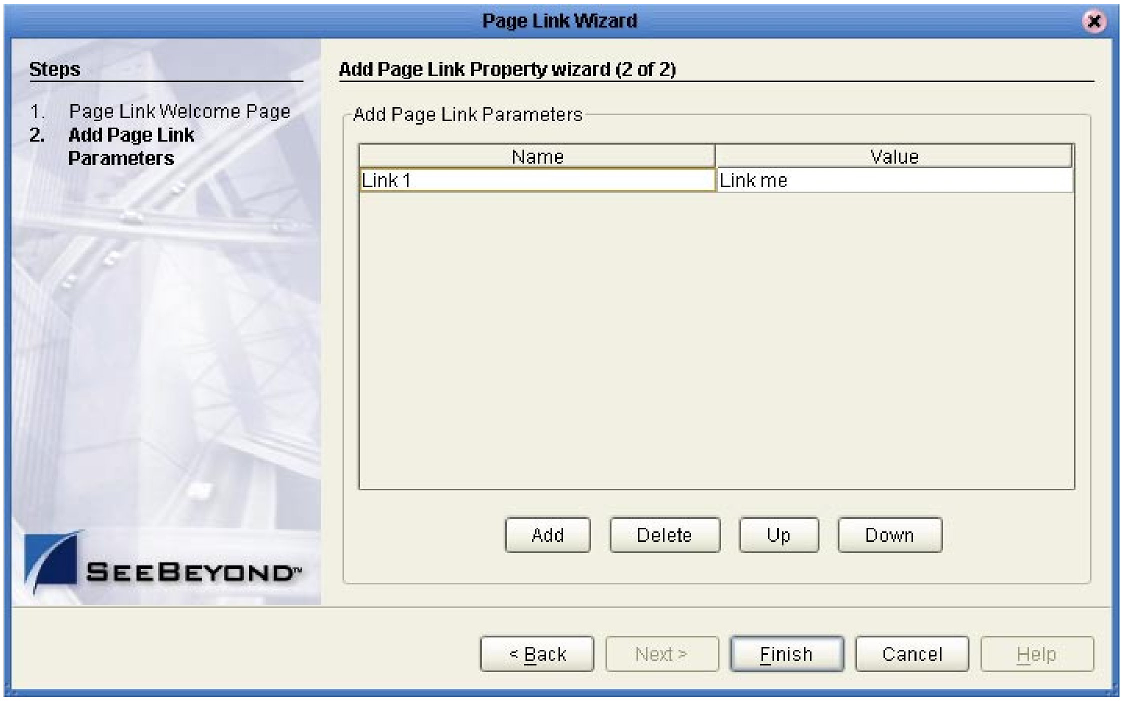
Welcome to Page Link Wizard wizard (1 of 2)

Page Link Name: Home page

☐ Designate as Home Page

5 Click **Next**. See Figure 52.

Figure 52 Page 2 of 2 in the Page Link Wizard



Page Link Wizard

Steps

1. Page Link Welcome Page
2. Add Page Link Parameters

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.

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

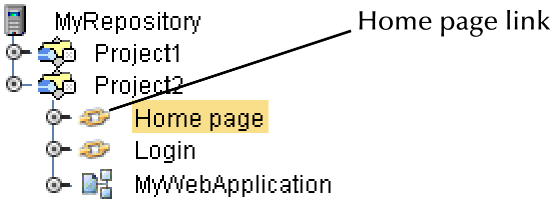
7 Modify the Page Link parameter fields, if necessary:

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

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

The Link is added to your Project in the **Enterprise Explorer** tree. See Figure 53.

Figure 53 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 87**.

Chapter 8

Using The 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:

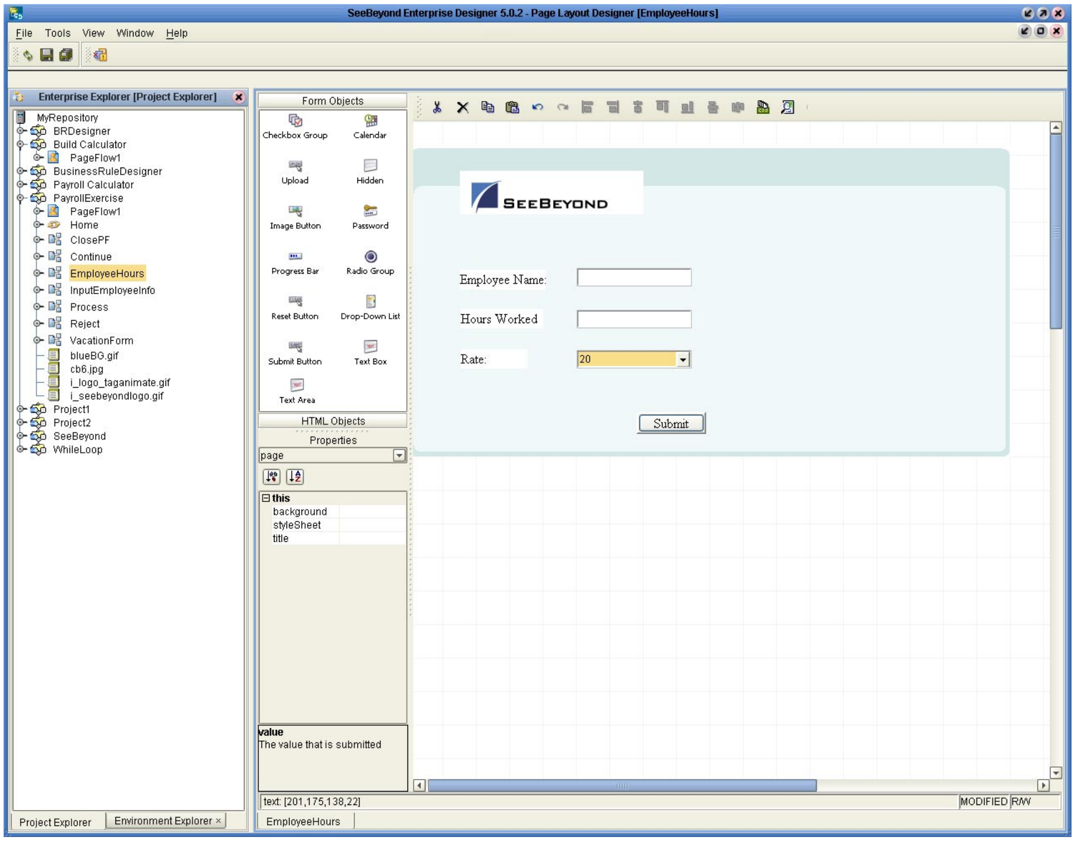
- **"Laying Out a Web Page"**
- **"Manipulating Objects on the Canvas" on page 83**

8.1 Laying Out a Web Page

This section describes how to build an input page for a small payroll calculation system using eVision's suite of 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. For an example of the finished page, see **Figure 54 on page 63**.

The page you will be building is also provided in finished form in the sample project that is packaged with eVision. You will see how the page is used in a Page Flow when you build the sample application described in **"Creating and Deploying a Web Application" on page 106**.

Figure 54 The Page Layout Designer Web Page Project



Rate: 20

value
The value that is submitted

text: [201,175,138,22]

8.1.1 Before you Begin

Before starting this exercise, extract the graphics files from the sample project into a folder on your desktop. Remember where you placed the files; you will be using them later in the exercise.

Downloading the Sample Project files

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

The Repository Server must be running before you can use either the Enterprise Manager or the Enterprise Designer. The first step of any procedure is to make sure the server is running, and if not, to start it.

To Start the Repository Server

1 On your computer, explore to:

```
<drive>\<path>\server\runServer.bat
```

Where:

drive is the drive letter where you installed eGate Integrator, and

path is the directory path to the **\server** folder.

2 Double-click **runServer.ba**t.

The command console will start to scroll the server startup messages. Wait until you see a message that confirms that the server has started.

To Download the Project Archive Files

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

1 Start your browser.

2 On the **Address** line, type the following:

```
http://<localhost>:<portnumber>
```

where:

localhost is the name of the machine that is hosting the Repository server, and

portnumber is the port number that was assigned to the server. For example:

http://localhost:12000

3 When the Enterprise Manager **HOME** page appears, click the **DOCUMENTATION** tab.

The Documentation Web page provides links to the documentation (PDFs), Javadoc(s), and code samples for the products you have installed.

Note: *Documentation is uploaded in a process that is separate from the product uploads. If you do not see a link to the eVision index page in the left frame, in Enterprise Designer, go to the* ***ADMIN*** *page and upload the eVision Documentation .sar file.*

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

5 In the right frame of the Documentation page, click the **Sample** link to launch the archive utility and prepare to download the sample archive files.

6 Using an archive utility (such as WinZip) save the archive file in a folder on your machine. You can use the archive program to create the folder. Remember where you saved the file.

7 Open the folder where you saved the archive file.

The extraction process will place two .zip files and two image (.gif) files into the designated folder. The contents of the folder will be:

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

Remember where you stored these files.

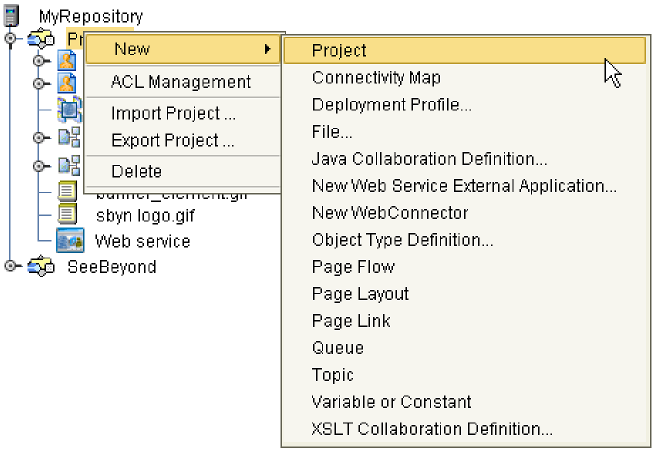
8.1.2 Starting a New Page Layout

1 Start the 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, select **Project**. See Figure 55.

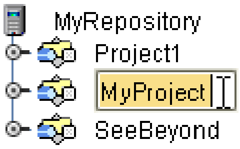
Figure 55 Project Context Menu



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 56.

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

Figure 56 Naming a New Project

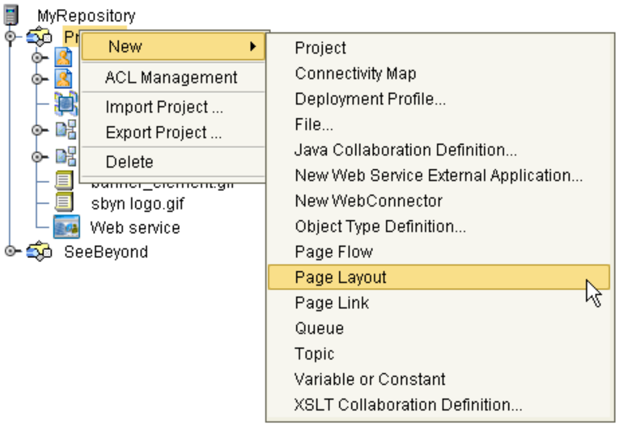


You do not need to press **Enter**; the next mouse click outside the name box will set the Project name.

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.

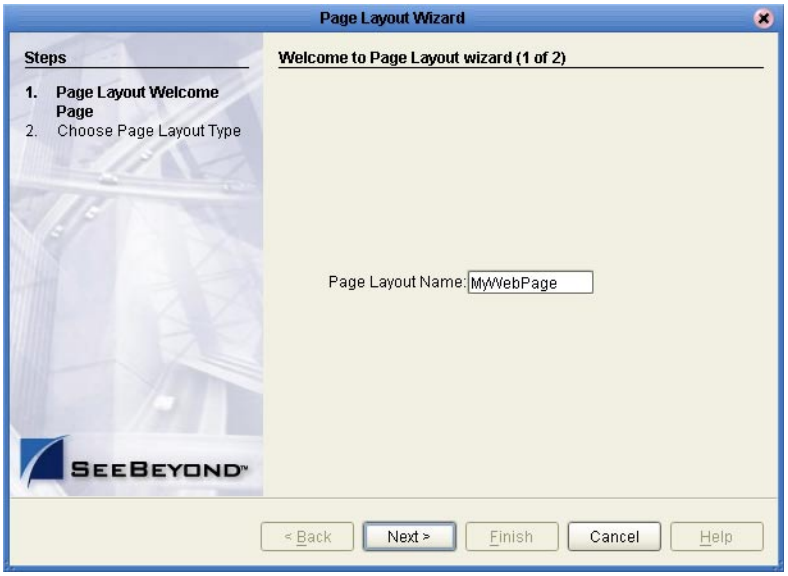
6 On the context menu, click **New**, and then select **New Page Layout**. See **Figure 57 on page 66**.

Figure 57 New Page Layout Context Menu



The Page Layout Wizard appears. See Figure 58.

Figure 58 Page Layout Wizard Page 1



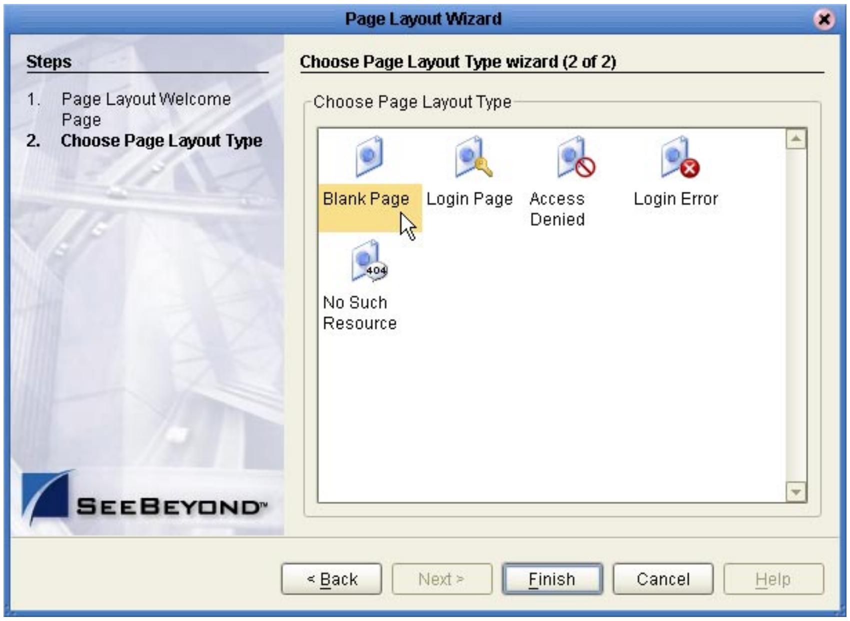
Welcome to Page Layout wizard (1 of 2)

Page Layout Name: MyWebPage

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

8 Click **Next**.

Figure 59 Page Layout Wizard Page 2



Page Layout Wizard

Steps

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

Choose Page Layout Type wizard (2 of 2)

Choose Page Layout Type

On the Page Layout Wizard, you can select any of the preconfigured Web pages based on a particular project's requirements. For example, to build this project, you start with a blank page.

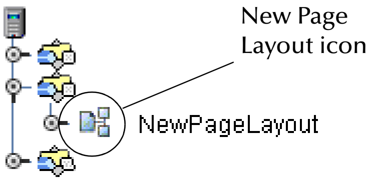
Note: *To create a Login Page for future projects, you must use the* ***Login Page template*** *provided on the Page Layout Wizard.*

9 In the **Choose Page Layout Type** window, click the **Blank Page** icon.

10 Click **Finish**.

A new **Page Layout** icon is added to the **Project Explorer** tree. See Figure 60.

Figure 60 Page Layout Icon in Project Explorer

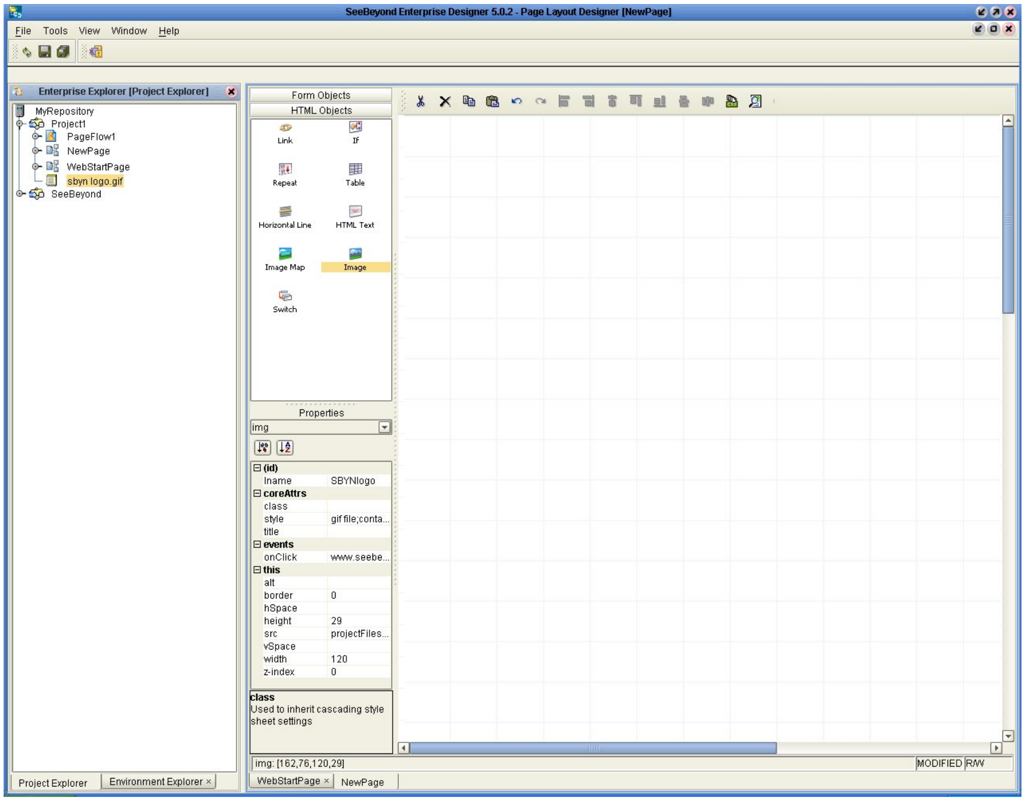


The name you provided in the Page Layout Wizard is appended to the eVision application icon name. The **eVision Studio** GUI appears in the work area of the Enterprise Designer with a blank canvas. See **Figure 61 on page 68**.

You have just created a blank Web page and started the Page Layout Designer. The following section describes how user the Page Layout Designer tools and pre-built GUI components to create a Web page that receives and passes user input.

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

Figure 61 eVision GUI with a Blank Canvas



⊟ **(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]

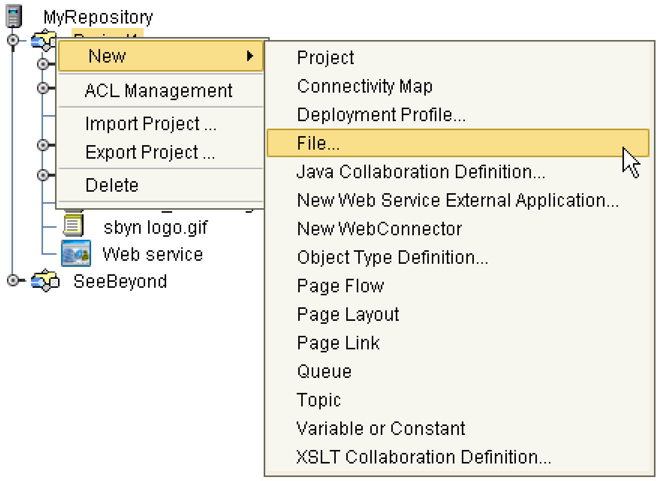
8.1.3 Importing Images

The sample Web page described in this section requires that you import two graphic images (a SeeBeyond logo and a background image) for use in the page layout. Before graphic images can be used in an eVision Web page, they must first be imported into the Project. In this section, the page layout procedure starts with importing the image files.

To Import a Graphic Image

1. In the **Enterprise Explorer** tree, right-click the **MyProject** icon.
2. On the context menu, click **New**, and then select **File**. See **Figure 62 on page 69**.

Figure 62 Importing an Image into the Repository



3 In the **Selected Import Files** dialog box, explore to the folder containing the image files that you extracted from the sample project archive file. See **Figure 63 on page 70**.

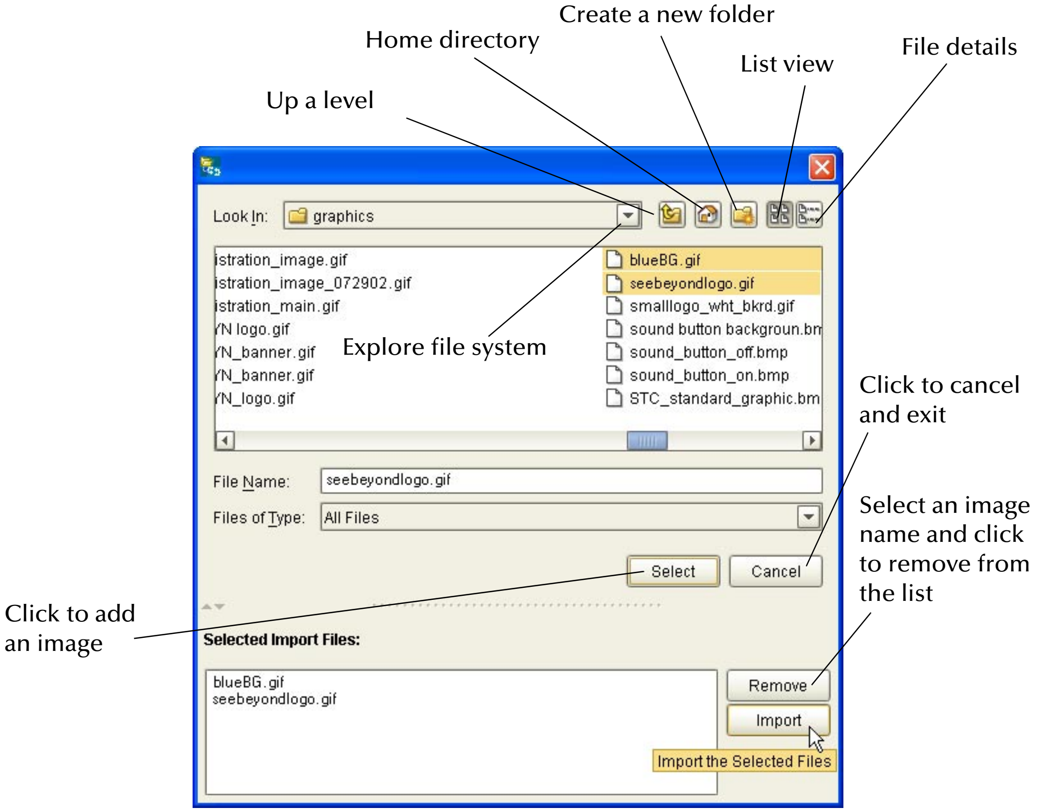
To explore to the location, double-click the folders to traverse the folder hierarchy. Graphic images can reside in any directory that is accessible to your computer, locally or on a network.

4 Select **blueBG.gif**, and click **Select**.

5 Click **seebeyondlogo.gif** and click **Select**.

6 Click **Import**.

Figure 63 Image Source Dialog Box



Look In: graphics

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

File Name: seebeyondlogo.gif

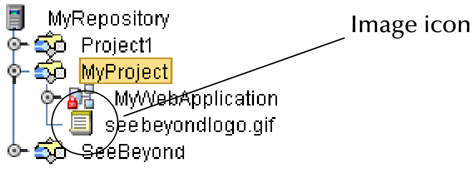
Files of Type: All Files

Selected Import Files:

blueBG.gif
seebeyondlogo.gif

An named icon for each image will be nested under the Project in the **Enterprise Explorer** tree. See Figure 64.

Figure 64 Graphic Image in Enterprise Explorer

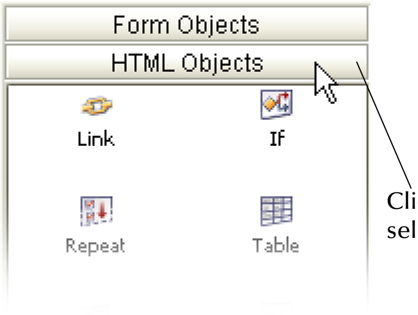


The image is now a part of your Project and can be imported into a image container on the Page Layout Designer canvas.

8.1.4 Placing GUI Components on the Canvas

This section describe how to place GUI components on the canvas using the Component selection palettes. Before you begin, make sure that the **HTML Objects** selection palette is in front. See **Figure 65 on page 71**.

Figure 65 Bringing the HTML Objects Selection Palette Forward



Click to swap selection palettes

Create the Background Layer Component

To create the **Background Layer** GUI component, on the **HTML Objects** selection palette, drag the **Image** icon onto the canvas. When the icon transfers onto the canvas it changes to a dashed-line rectangle.

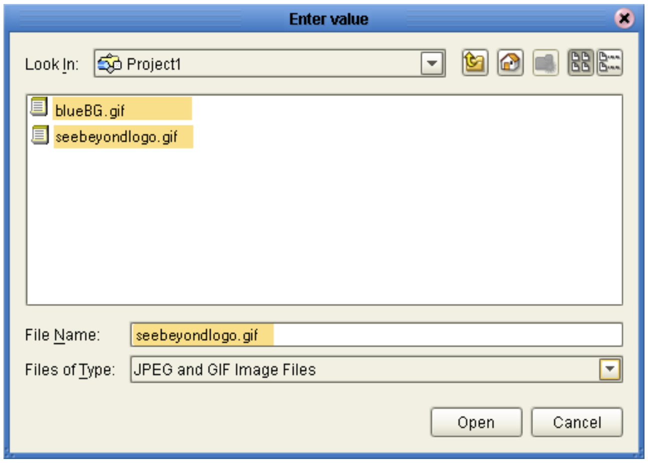
Do the following:

7 Drop the rectangle on the approximate center 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 66.

Figure 66 Selecting an Image for the Image Object Container



Look In: Project1

blueBG.gif
seebeyondlogo.gif

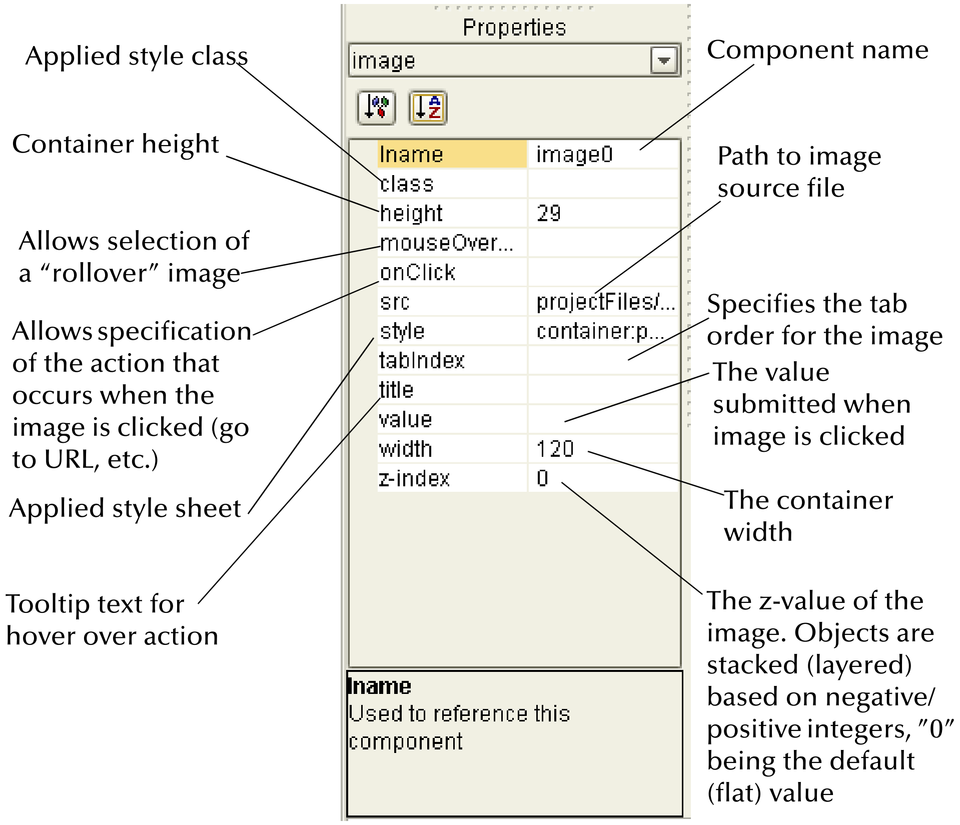
File Name: seebeyondlogo.gif

Files of Type: JPEG and GIF Image Files

8 In the dialog box, select the **blueBG.gif** file, and then click **Open**.

The image is linked and displayed in the container. The container is automatically resized to accommodate the image. The **Component Properties** sheet for the **Image** component is automatically displayed. See **Figure 67 on page 72**.

Figure 67 Component Properties Sheet for the Image Component



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 (layered) based on negative/ positive integers, ”0” being the default (flat) value

9 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 **Background**).

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.

About the Z-value Index: Adjusting the Stack Order of GUI Components

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

- To modify the z-value of the image, right click the image container, and on the context menu, do the following:

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

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

- To move an image down (back) one or more levels, click the **z-index** field on the property sheet, and type a negative integer; for example: **-9**.
- To move an image up (forward) one or more levels, click the **z-index** field on the property sheet and type a positive integer; for example **+7**.

Create the Page Banner Component

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 to a dashed-line rectangle.

Do the following:

1. Drop the rectangle inside the boundaries of the background image on 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 again.

2. In the **Enter value** dialog box, select the **seebeyondlogo.gif** file and click **Open**.

   The image is linked and displayed in the container. The **Component Properties** sheet for the **Image** component is automatically displayed.

3. 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 **logo**).

   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.

4. To mover the logo image to the top of the stack, click the **z-index** field in the property sheet and type **+5** (this number is arbitrary and can be changed at any time).

Create the Employee Name Component

To create the **Employee Name** GUI component, on **HTML Objects** selection palette, drag the **HTML Text** icon onto the canvas.

1 Place the component left-aligned with the **logo image** component.

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

2 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: **EmployeeName**.

3 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: **Employee Name**.

D Click **OK**.

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

Initially, on the canvas, the text string may 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 can remain approximately left-aligned with the other components on the page. The components will be aligned in another step.

Create the Hours Worked Label GUI Component

To create the **Hours Worked** GUI component, from **HTML Objects** selection palette, drag the **HTML Text** icon onto the canvas.

1 Place the icon under, and left-aligned with the **EmployeeName** component.

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

2 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: **HoursWorked**.

3 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: **Hours Worked**.

4 Click **OK**.

Create the Rate Label Component

To create the **Rate Label** GUI component, on **HTML Objects** selection palette, drag the **HTML Text** icon onto the canvas.

1 Place the component under, and left-aligned with the **HoursWorked** component.

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

2 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: **HourlyRate**.

3 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: **Rate**.

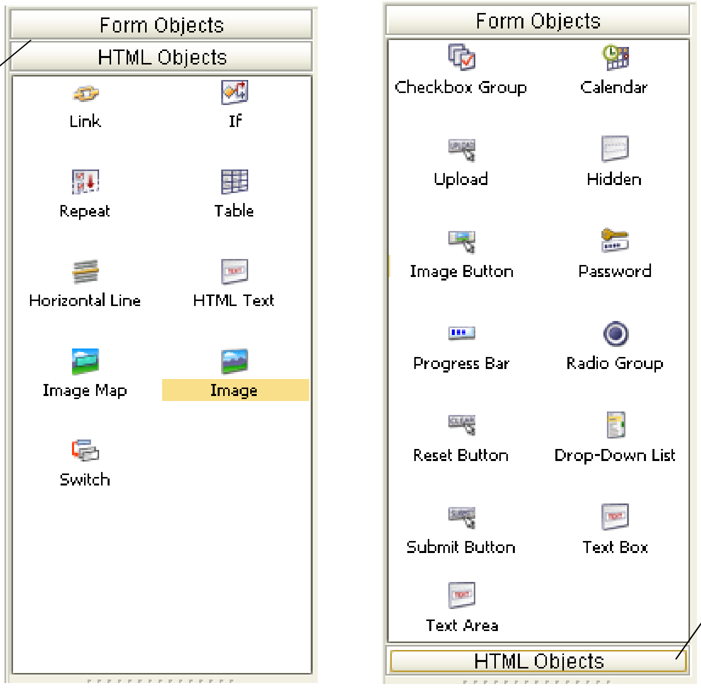
4 Click **OK**.

The text will be displayed to users on the Web page as a field label, which prompts them to enter their hourly pay rate in the associated 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 68.

Figure 68 HTML and Form Objects Palettes



Click to bring Form Objects to the front

Click to bring HTML Objects to the front

Create the Employee Name Input Field Component

To create the **Employee Name** input GUI component, on the **Form Objects** selection palette, drag the **Text Box** icon onto the canvas., and then do the following:

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

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

2. 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: **EmpNameInput**.

3. 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 this 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: **Employee Name**

Create the Hours Worked Input Field Component

To create the **Hours Worked input** GUI component, from the **Form Objects** selection palette, drag the **Text Box** icon onto the canvas.

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

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

2. 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: **HoursWorked**.

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

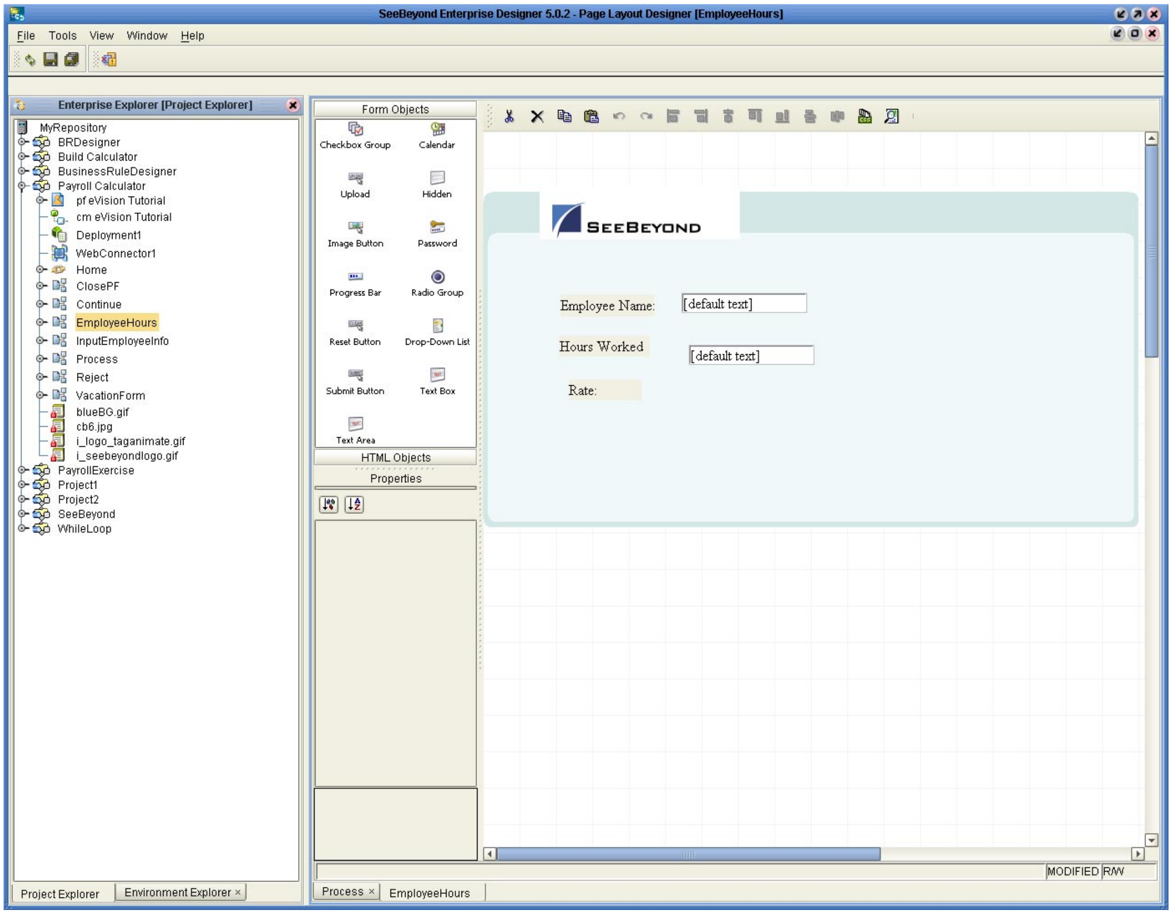
   A Under the **Value** heading, click the **[default text]** field.

   B Delete the **[default text]** string. (You want this 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: **Hours Worked**.

At this point, the unfinished canvas should look like the example shown in **Figure 69 on page 78**.

Figure 69 Incomplete Page



Create the Rate Drop-down List Box Component

To create the **Rate Drop-down List** GUI component, on the **Form Objects** selection palette, drag the **Drop-Down List** icon onto the canvas.

1. Align the component horizontally with, and to the right of the **Hours Worked Input** component. Make sure the components' vertical edges do not overlap.

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

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

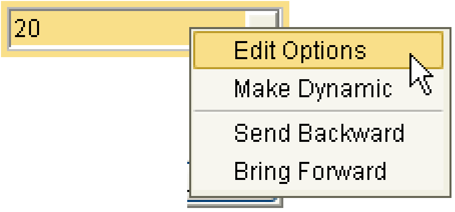
   A. Under the **Value** heading, click the **lname** field. (The default name will be **password(*n*)**).

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

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

4. On the context menu, select **Edit Options**. See **Figure 70 on page 79**.

Figure 70 Drop-Down List Menu



5 In **Edit Options** dialog box, under the **Labels** heading, click the empty input field, and type: **20**. This is the default pay rate the will be displayed to users in the drop-down list on the Web page. See Figure 71.

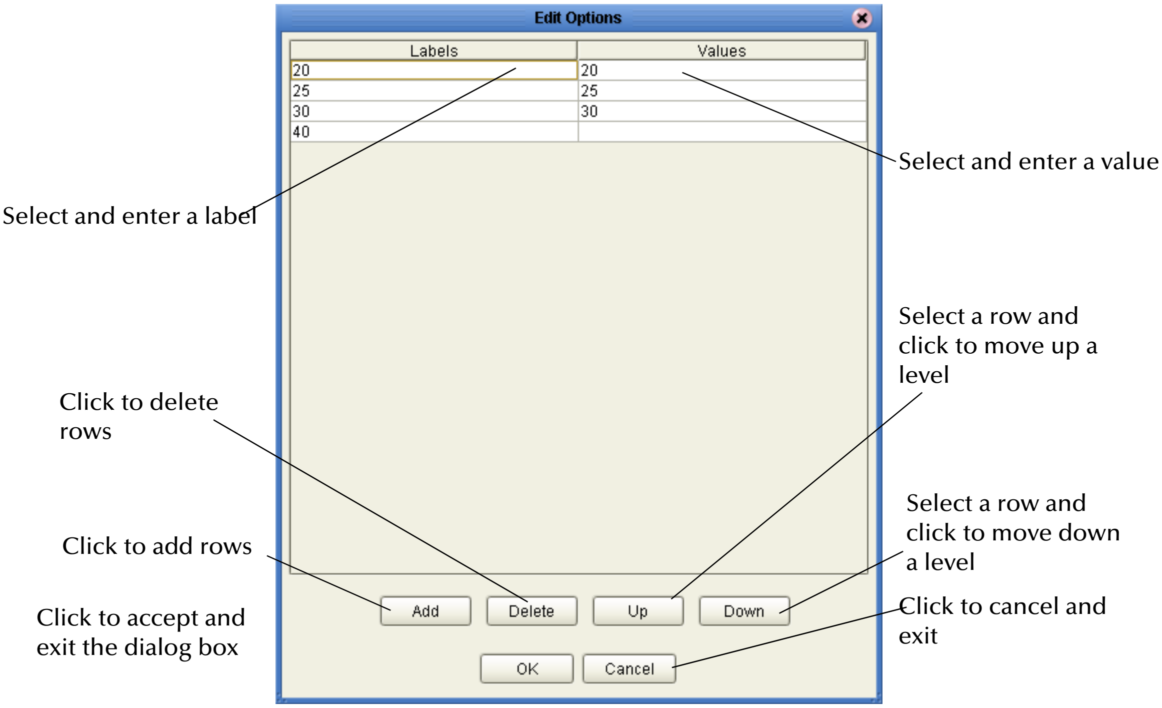
6 In **Edit Options** dialog box, under the **Values** heading, click the empty input field, and again type: **20.** This is the label for the drop-down list entry.

7 To add three additional input row, select the top row, click **Add** three times in succession, and then do the following:

- In row 2, select the **Labels** field and type **25** and repeat in the **Values** field.
- In row 3, select the **Labels** field and type **30** and repeat in the **Values** field.
- In row 4, select the **Labels** field and type **40** and repeat in the **Values** field.

The **Edit Options** dialog box should now look like Figure 71.

Figure 71 Edit Options Dialog Box



Labels	Values
20	20
25	25
30	30
40	

Select and enter a value

Select and enter a label

Select a row and click to move up a level

Click to delete rows

Select a row and click to move down a level

Click to add rows

Click to cancel and exit

Click to accept and exit the dialog box

Create the Submit Button Component

To create the **Submit Button** GUI component, on the **Form Objects** selection palette, drag the **Submit** icon onto the canvas, and do the following:

1. Drop the component below, and right-aligned with the **Rate Drop-down List** component.

   The **Component Properties** sheet for the **Submit Button** 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 **submit(*n*)**).

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

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

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

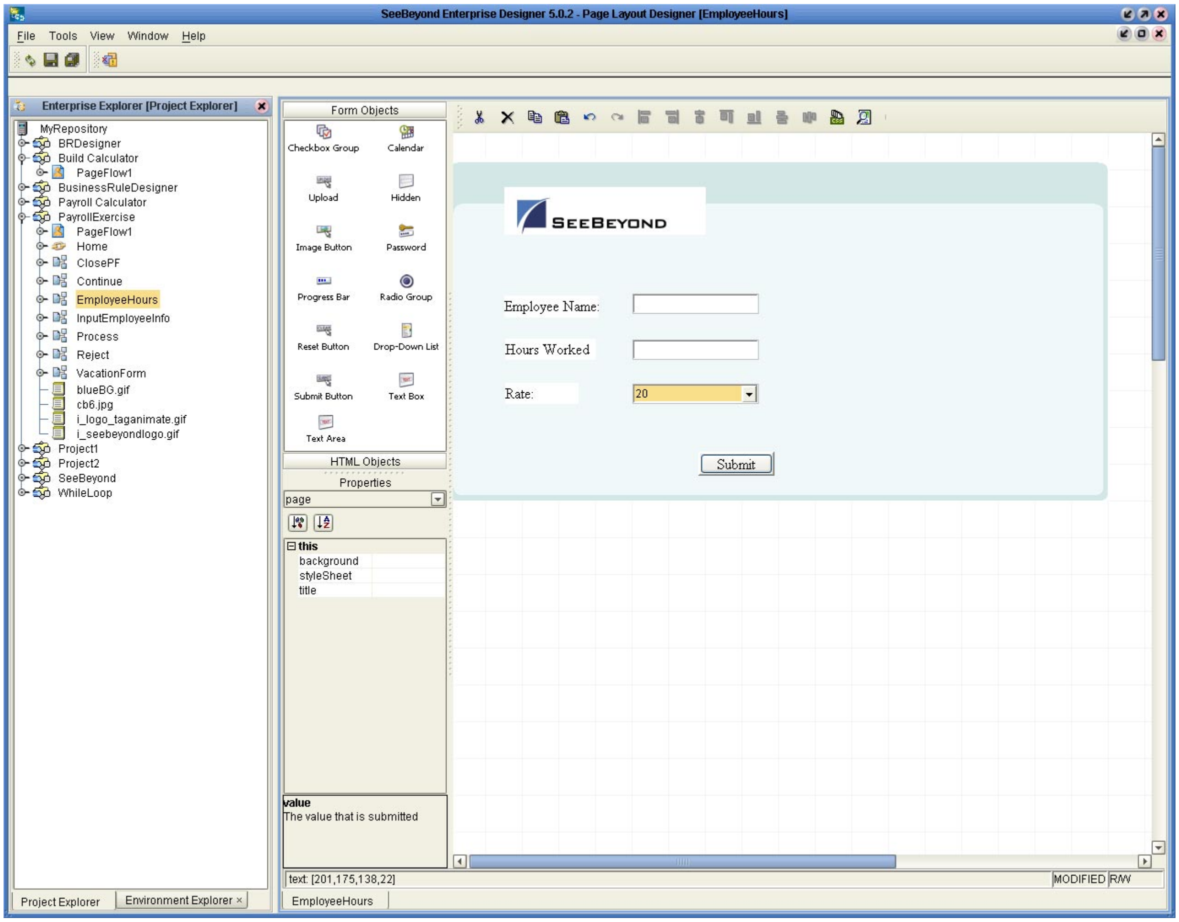
   B. In the **value** field, type: **Submit**.

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

8.1.5 Reviewing the Page Layout

On the design canvas, the finished **Login** page should look like the example shown in **Figure 72 on page 81**.

Figure 72 Finished Input Page



Rate: 20

value
The value that is submitted

text: [201,175,138,22]

8.1.6 Aligning Objects Using the Alignment Tools

If the GUI components are not precisely aligned, or the space between the objects is inconsistent, you can adjust the alignment of the components in several different ways 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 desired alignment tool. Alignment will occur as described in Table 5.

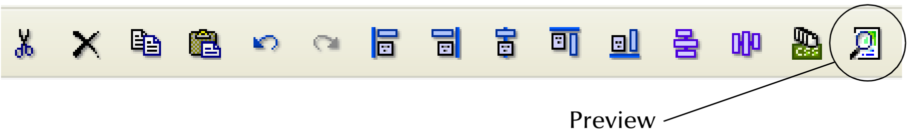
Table 5 eVision Studio Object Alignment tools

Tool	Name	Purpose
	Left Align	Aligns two or more selected components with the left-most component in the selected group.
	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.

8.1.7 Previewing the Finished Web Page

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

Figure 73 eVision Studio Toolbar



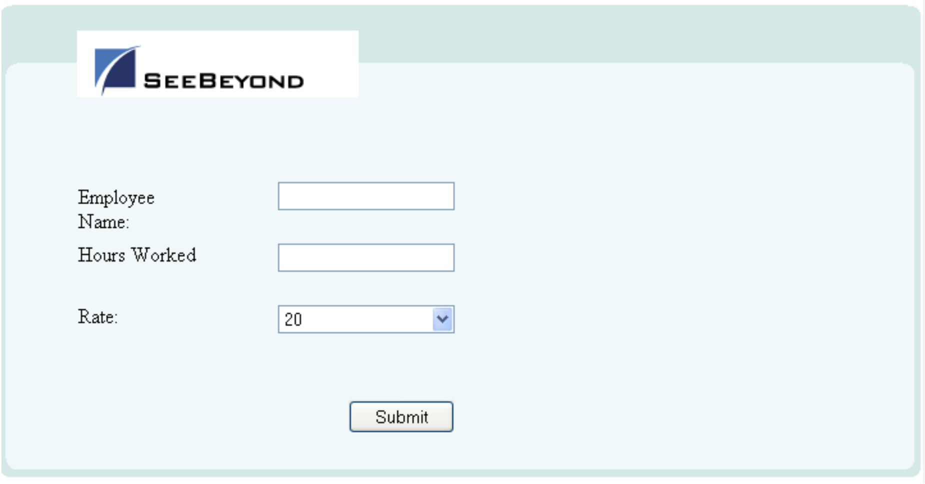
The page will display in a new browser window. It should look like the example shown in **Figure 74 on page 83**.

8.1.8 Saving the Project in the Repository

To save the page layout to the Repository, on the Enterprise Designer menu, select **File,** and then 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 84**.

Figure 74 Web Page Preview



8.2 Manipulating Objects on the Canvas

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

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 sequentially select the objects you want to move.

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

8.2.3 Resizing Objects

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

Select the object or objects as described in **"Moving a Single Object" on page 83**, and **"Moving a Temporary Group of Objects" on page 83**. 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.

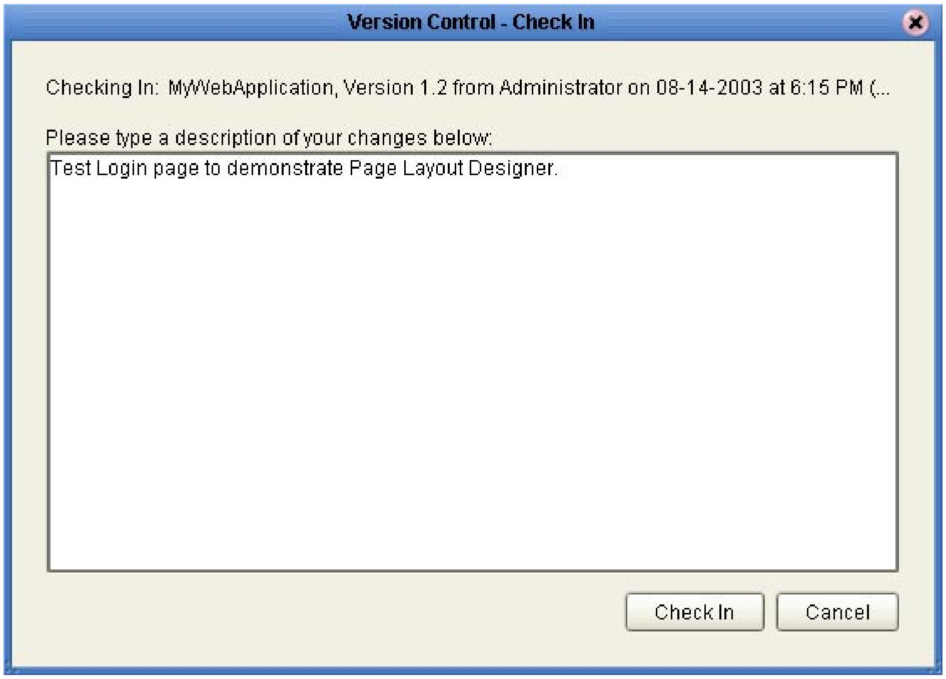
8.2.4 Checking the Page Layout in to Version Control

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

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, select **Check In**.
2. In the **Version Contro**l dialog box, type a comment to accompany the Page Layout. See Figure 75.

Figure 75 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.

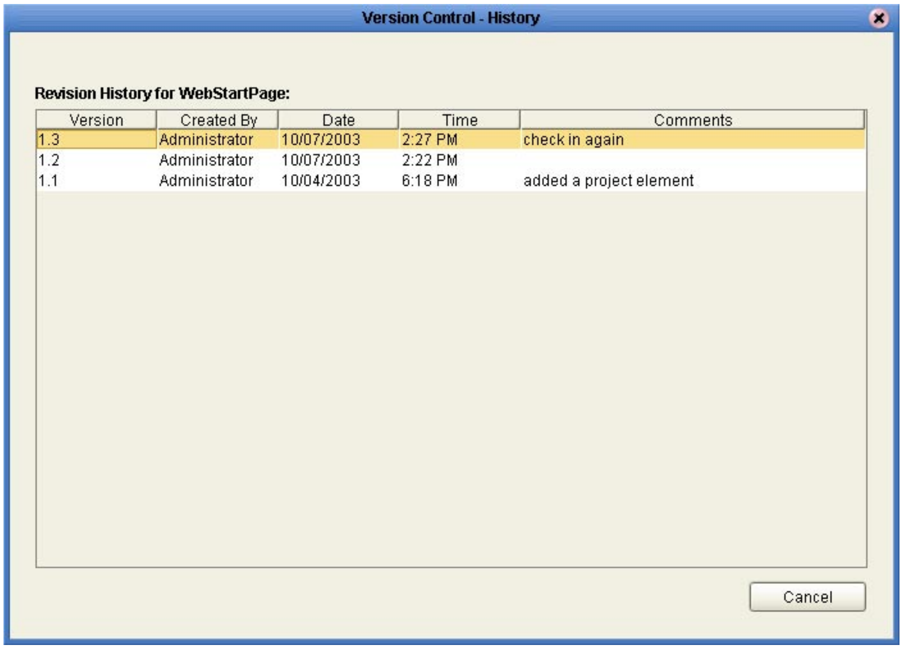
3 Click **Check In**.

To Check Out a Page Layout

1 In the **Enterprise Explorer**, right-click the name of the Page you want to check out.

2 To review the version history of the Page, on the context menu, select **Version History**. See **Figure 76 on page 86**.

Figure 76 Version History



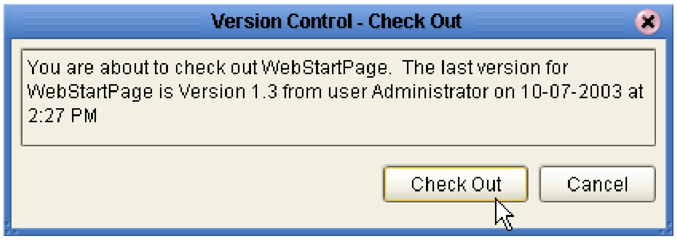
Revision History for WebStartPage:

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, select **Check Out**. See Figure 77.

Figure 77 Checking 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**.

For more information on the Version Control system, see the ***SeeBeyond ICAN Suite User's Guide***. This guide is available via the Documentation Web page.

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, eVision's Page Flow Designer allows you to link the Web pages together and create a logical chain called a Page Flow.

This Chapter Includes:

- **"Creating a Page Flow"**
- **"Using the Page Flow Designer GUI" on page 89**
- **"Page Flow Elements" on page 90**
- **"Configuring Page Flow Designer Elements" on page 95**
- **"Page Flow Properties" on page 97**

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.

Adding a Page Flow to a Project

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

To add a Page Flow to a Project

1. From the **Enterprise Explorer** pane, right-click **Project**.
2. From the context menu, click **New**, and then select **Page Flow**.
3. In the Explorer window, 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 Page Flow elements on the canvas to visually create a flowing design for the flow of the Web application. Then, from the Enterprise Explorer, you drag and drop Web pages onto Page Flow elements, and then link the collection of elements together to create a logical page flow for the application. The Page Flow Designer provides the tools to lay out and connect Page Flow elements and other specialized sub-process elements, including editing capabilities that allow you to adjust and align Web page and sub-process components. An example of the Page Flow Designer GUI is shown in Figure 78.

Creating a Page Flow

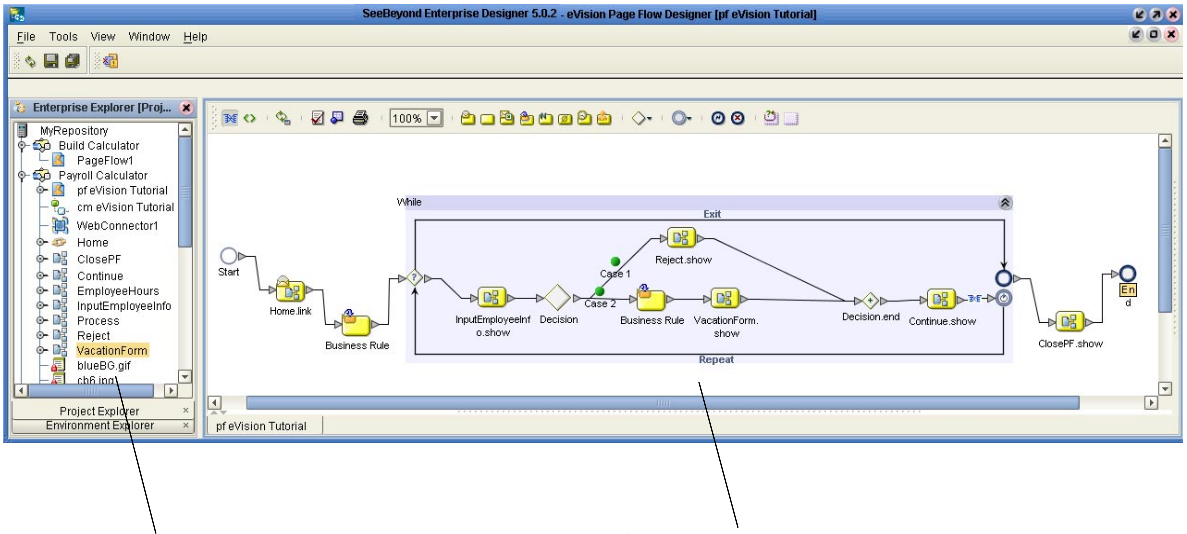
When you start a new Web application project, the **Start** and **End** Page Flow icons automatically appear on the blank Page Flow Designer canvas. (There can be only one starting point for a Page Flow; however, there can be multiple end points.)

To create an initial Page Flow

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

Figure 78 Page Flow in the Page Flow Designer GUI



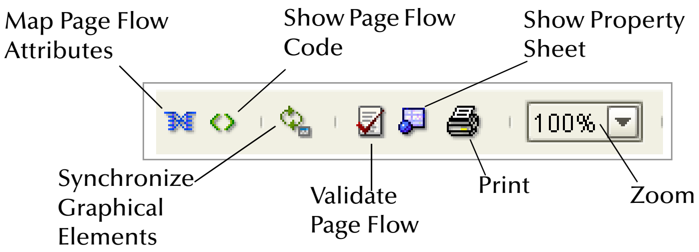
Project Explorer

Page Flow Designer canvas

9.2 Using the Page Flow Designer GUI

The Page Flow Designer toolbar is shown in Figure 79.

Figure 79 The Page Flow Designer Toolbar



The Page Flow Designer tools are described in Table 6.

Table 6 The 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 code generated by the Page Flow Designer.
	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

The 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 process elements, branching elements, and intermediate events.

Sub-process Elements

There are several different types of elements that can be added to a Page Flow. See Table 7.

Table 7 Connecting and Sub-process elements

Button	Command	Function
○	**Start Node**	The **Start Node** indicates the start of the process. The Start node is automatically added to the canvas when new Page Flow is initialized. A Start Node can connect to a **Page Link**, or an **Event-based Decision** element.
▷─▷	**Link**	**Links** define the connectivity of the Page Flow by connecting page and sub-process elements together. When you select a link, a context menu allows you to configure how data is going to be passed to and from the underlying component or Web Service using Attributes. The Page Flow Designer ensures the Page Flow is correctly linked by rejecting invalid links. Links can also accept mapped values. A link with mapped values will displayed with the “map” icon.
○	**End Node**	The **End Node** indicates the completed state of a Page Flow. The End node is automatically added to the canvas when new Page Flow is initialized.
✉	**Receive**	The **Receive** element can connect to a **Page Link** or can connect to a **Start** node via a **Page link**. It is used to indicate the invocation of the Page Flow. The Receive element represents the actual method by which a Page Flow is initiated, for example: A user types a URL into 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.

Table 7 Connecting and Sub-process elements

Button	Command	Function
	Reply	The **Reply** element allows a Page Flow to respond to the external system or user that originally invoked the Page Flow. The original receive at the beginning of the business process is paired with the Reply at the end of the process. In cases where a message must be sent back to the caller of the process. the Reply uses information that correlates the message in the calling system. A Reply acts as the last step in a Page Flow, in which the process is acting as a Web Service or Sub-Process. A Reply correlates the outbound message back to the calling process, for example, it can reply to an external system as a Web Service.
	Assign	The **Assign** element sets data values, including task assignments. It is used when pages have multiple data mappings between the invocation of human tasks or automated systems.
	Compensate	The **Compensate** element is used to invoke compensation on an inner scope that has already completed normally. This construct can be invoked only from within a fault handler or another compensation handler.
	Empty	This **Empty** element allows data to pass through without changes.
	Wait	The **Wait** element acts as a timer. If the user builds a Page Flow in which there are two simultaneous paths within a set framework (one for the page flow, one for the timer, if the timer condition takes place first, an exception will be thrown, handled, and the Page Flow will then be abandoned.

Branching Elements

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

Adding a Branching element

To add a Branching element to the Page Flow Designer canvas

1. On the Page Flow Designer toolbar, click the expansion arrow on the **Branching Activities** icon, and the select the type of branching you would like to add.
2. Click the type you want to use and drag it to the Page Flow Designer canvas.

   The selected **Branching element** appears on the Page Flow Designer canvas.

Branching elements are described inTable 8.

Table 8 Branching Elements

◇	**Decision**	The **Decision** element allows one of several possible paths to execute, based on expression logic. This element is used to create complex expressions that determine the path of the Page Flow. It also contains the expression and connection names. The Decision element allows you to define expressions that are evaluated to determine the routing of the Page Flow. Expressions are built using the mapping interface and Page Flow attributes. ***Note:*** *The Decision element is structured to automatically raise a run-time Exception to alert you to Page Flow errors during construction. You can set this value to Return True, or Return False; however, it is not recommended during the construction phase; errors can be hidden during development and only surface in the post-deployment, run-time environment.*
◇	**Event Based Decision**	The **Event-Based Decision** element allows multiple messages to be juxtaposed against a time-out condition, enabling the message type to specify the appropriate Page Flow path.
◇	**Flow**	The **Flow** element specifies that one or more pages and/or processes 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 the **Intermediate event** toolbar icon and select the type of **Intermediate event** you would like to add.
2. Click 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	The **Timer Event** imposes a time-out condition on Pages, groups of Pages, or a Page Flow as a whole to ensure that processes complete within a specified time-frame. Conditions also allow the creation of the process that takes place after a time-out condition takes place.
◎	Message Event	The **Message Event** is similar to a Receive Activity, but it occurs only in the middle of a process. Each of these elements can be a different message.

Table 9 Intermediate Events

	Catch Named Exception	Each automated system (backend system) or Web service can publish their possible error codes (for instance, fault 15 is "bad data"). Those codes can be mapped to exception handlers. Each exception handler is connected to the scope that surrounds one or more steps in a Page Flow. The components within that scope will throw the exceptions when errors occur and the exception handler will automatically initiate the appropriate process to handle the problem.
	Catch All Exceptions	The **Catch All Exceptions** handler is configured to handle all exceptions.
	Compensation Handler	The **Compensation** handler is used when something in a Page Flow fails and requires a rollback based on upstream activities. On an automatic basis in the Page Flow, upstream steps in the Page Flow are notified that the failure has occurred and certain transactions need to be reversed, sometimes in a sequential order. The Compensation handler allows you to design the process and circumstances, in which the compensation takes place.
	Throw	The **Throw** handler throws exceptions.
	Terminate	The **Terminate** handle 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 expansion arrow on the **Intermediate event** icon, and then select the type of **Intermediate event** you would like to add.
2. Click the type you want to use and drag it to the Page Flow Designer canvas.

   The selected **Intermediate event** appears on the Page Flow Designer canvas.

9.3.1. While

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

Table 10 While loop

	While	The While loop creates and maintains a looping process within a Page Flow. A loop continues a process until an event takes place that signals that the Page Flow is to continue.

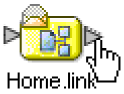
9.3.2. Links

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

To Link Web Pages and Other Page Flow Elements

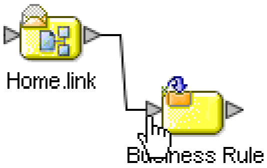
1. Move your cursor over the connector portion of the Page Flow element.
2. Hold the cursor over the outside edge of the element until it changes from the arrow pointer to the hand pointer. See Figure 80.

Figure 80 Starting a Link



3. Drag a line from the forward connector on the first Web page to the connector on the second Web page or design element, then release the mouse. See Figure 81.

Figure 81 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.*

9.3.3. 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**.**

9.3.4. 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

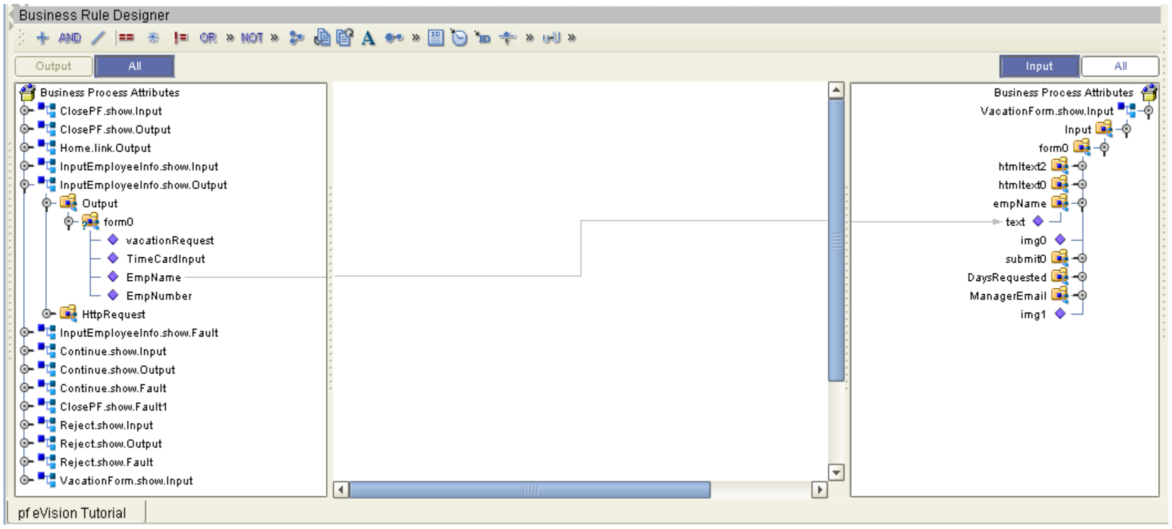
Some elements in Page Flow Designer have configurable options. This section describes the elements and explains how to configure the options.

9.4.1. About Business Rule Designer

The **Business Rule Designer** allows you to configure relationships between Input and Output Attributes. Some attributes are automatically configured for each sub-process when you drag and drop a component on the Page Flow Designer. The area where you map attributes in the **Business Rule Designer** is called the *Mappe*r. See Figure 82.

The **Business Rule Designer** appears whenever you click the **Map Page Flow Attributes** icon. Also, it is automatically activated when you add or select an inline Business Rule or Assign element.

Figure 82 Business Rule Designer



Adding an Inline Business Rule

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

To add an inline Business Rule

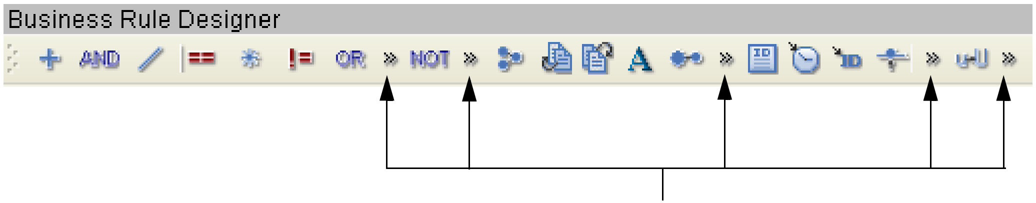
1. In the **Page Flow**, select a link that connects two Page Flow elements.
2. Right-click the mouse.
3. On the context menu, select **Add Business Rule**.

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

Method Palettes

Use the **Method Palettes** in the Business Rule Designer to configure data that is passed between input and output Pages. You can drag and drop a method from a method palette to the Business Rule Designer and then configure the method. To open a Method Palette, click the horizontal chevrons on the Business Rule Designer toolbar. See **Figure 83 on page 96**.

Figure 83 Business Rule Designer Toolbar



Click to open a Method Palette

The methods in the Method Palettes are detailed **“Method Palettes” on page 134**.

Sub-Processes

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

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

To Use a Sub-process

1. Create the parent process – add an empty Activity as a placeholder for the sub-process.
2. Create the child process – add an empty Receive at the beginning and a Reply at the end.
3. To create a WSDL file that will represent the inputs and outputs of the process, within the child sub-process, do the following:

   A In the Environment Explorer, right click the Page Flow to which you want to add the sub-process.

   B On the context menu, select **Open Property Sheet**.

   C On the **Page Flow Properties** dialog box, select the **WSDL** tab.

   D On the **WSDL** tab, click **Create**.

   E Specify values for **portType**, **Operation**, **Input** and **Output**.

   F Click **OK**.

   G Open the properties of the initial Receive and select the appropriate partner, portType, and operation.

   H Do the same for the Reply.

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

4. Save the child process.
5. Return to the parent process – from the Project Explorer, drag the 1st operation under the child process to the sub-process placeholder in the parent process.
6. Perform any appropriate mappings using **Assign Activities** in the parent Page Flow.
7. Deploy both processes.

   For more information on the **Page Flow Properties** dialog box, and how to add WSDL to a Page Flow, see **"Page Flow Properties: WSDL Tab" on page 105**.

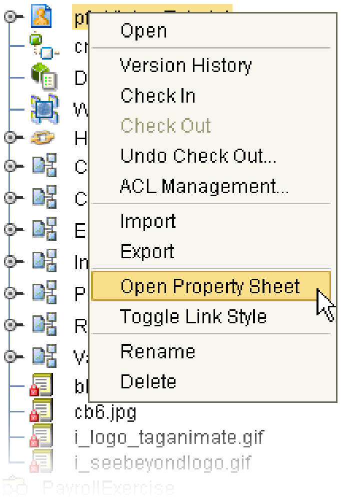
9.5 Page Flow Properties

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 Rule Designer.

To Edit Page Flow Properties

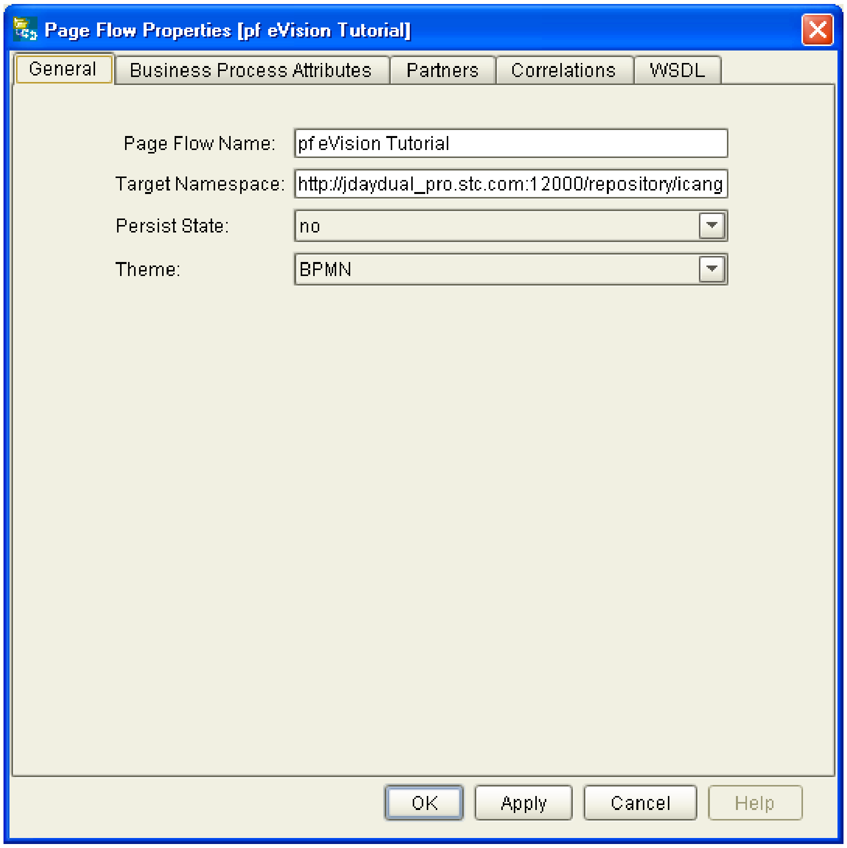
1. Right-click the **Page Flow** you want to edit.
2. On the context menu, select **Open Page Flow Properties**. See Figure 84.

Figure 84 Opening the Page Flow Property Sheet



The **Page Flow Properties** dialog appears. See **Figure 85 on page 98**.

Figure 85 Page Flow Properties Dialog Box: General Tab


Page Flow Properties [pf eVision Tutorial]
General
Business Process Attributes
Partners
Correlations
WSDL
OK
Apply
Cancel
Help

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

To edit General Page Flow Properties

1. From the **General** tab, you can edit:
   - Page Flow Name
   - Target Namespace
   - Persist State
2. To Save your changes and exit the **Page Flow Properties** dialog box, click **OK**.

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 a Page Flow.

Some examples of Page Flow Attributes are:

- customer names
- addresses
- order quantities
- item descriptions

Page Flow Attributes are used to pass values between the Page Flow and external sources. Page Flow Attributes can also be assigned to specific activities. For example, the customer name is passed to an order process from the originating source. The customer name may be used by several of the activities in the Page Flow and is included in the Page Flow output.

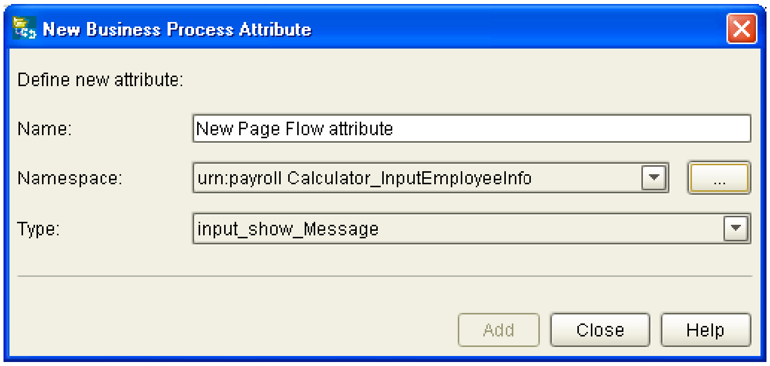
eVision can pass all or part of a complex structure or it can even assemble a composite input to a component or Web Service from multiple Page Flow attributes.

To Create a New Page Flow Attribute

1. Select the **Page Flow Attributes** tab.
2. In the Enterprise Explorer, right-click the **Page Flow** icon.
3. On the context menu, select **Open Property Sheet**.
4. Select the **Page Flow Attributes** tab.
5. On the **Page Flow Properties** dialog box, click **Create**.

   The **New Page Flow Attribute** dialog box appears. See Figure 86.

Figure 86 New Page Flow Attribute



New Business Process Attribute

Define new attribute:

Name: New Page Flow attribute

Namespace: urn:payroll Calculator_InputEmployeeInfo

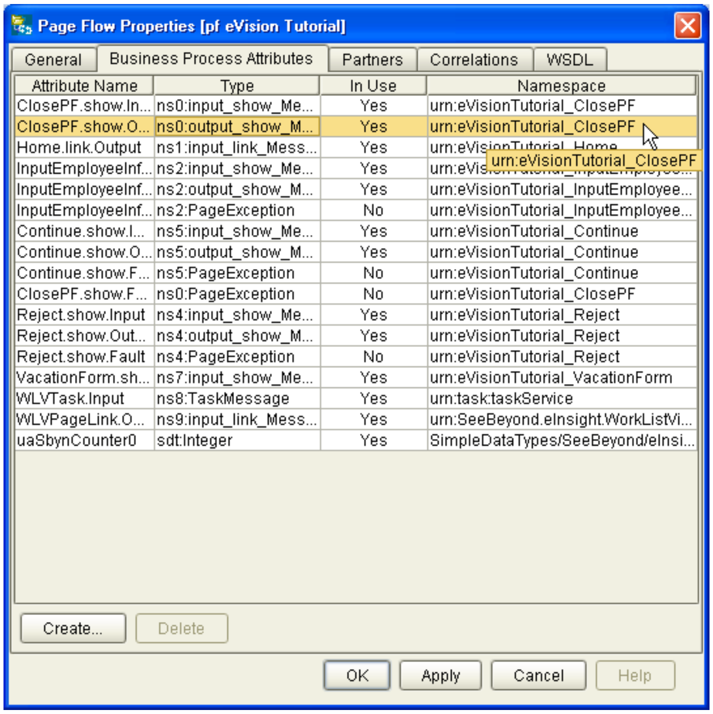
Type: input_show_Message

6. In the **New Page Flow Attribute** 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.
7. To save the attribute, click **Add**.
8. To return to the **Page Flow Properties** dialog box, click **Close**.

To edit a Page Flow Attribute

1 On the **Page Flow Attributes** dialog box, select the **Page Flow Attributes** tab. See Figure 87.

Figure 87 Page Flow Properties Dialog Box: Page Flow Attributes tab


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

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

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

- To rename an attribute, double click the attribute text and type a new name. Some attributes cannot be renamed.
- To remove an attribute, select the attribute text and click **Delete**.

3 To save your changes and exit the **Page Flow Properties** dialog box, click **OK**.

Correlation Keys and Sets

Configuring Correlation comprises the following steps:

- **"Creating a Correlation Key" on page 101**

  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.

- **"To Add Correlation Sets" on page 103**

  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.

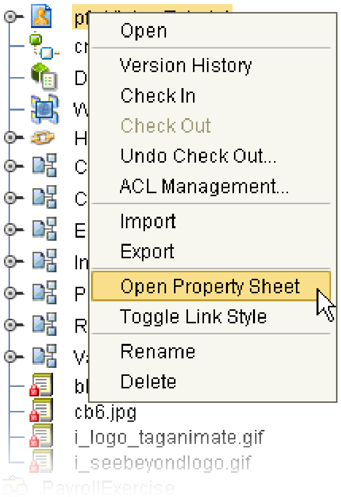
- **"To bind Correlation Sets to Page Flow elements" on page 103**

  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).

Creating a Correlation Key

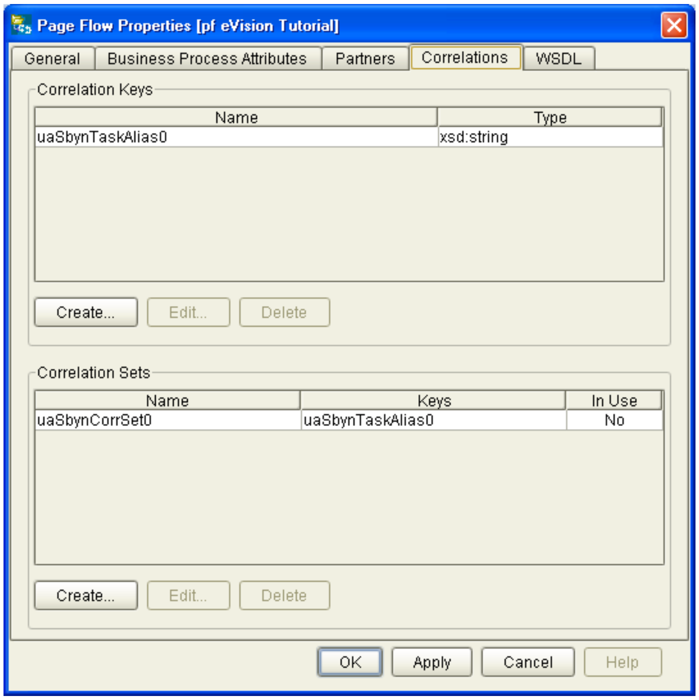
1. To invoke the **Page Flow Properties** dialog box, in the Enterprise Explorer, right-click the Page Flow icon.
2. On the context menu, select **Open Property Sheet**. See Figure 88.

Figure 88 Opening the Page Flow Property Sheet



3. On the **Page Flow Properties** dialog box, select the **Correlations** tab. See **Figure 89 on page 102**.

Figure 89 Page Flow Properties: Correlations Tab

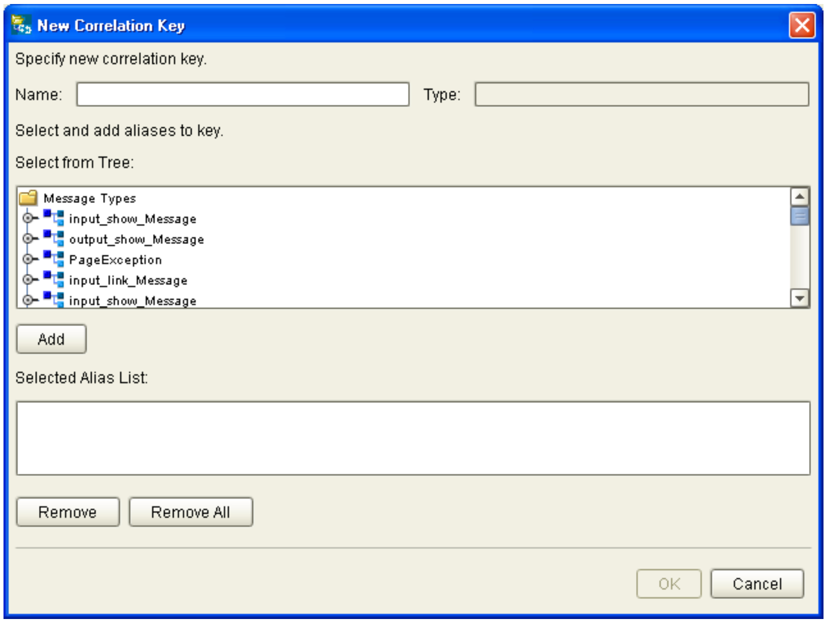


Name	Type
uaSbynTaskAlias0	xsd:string

Name	Keys	In Use
uaSbynCorrSet0	uaSbynTaskAlias0	No

4 On the **Correlation Keys** section of the Page Flow Properties dialog box, click **New**. The **New Correlation Key** dialog appears. See Figure 90.

Figure 90 New Correlation Key Dialog Box



5 In the **New Correlation Key** dialog box:

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

J 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.

6 To save the new alias to the **Selected Alias List**, click **Add**

7 To save your changes and exit the **New Correlation Key** dialog box, click **OK**.

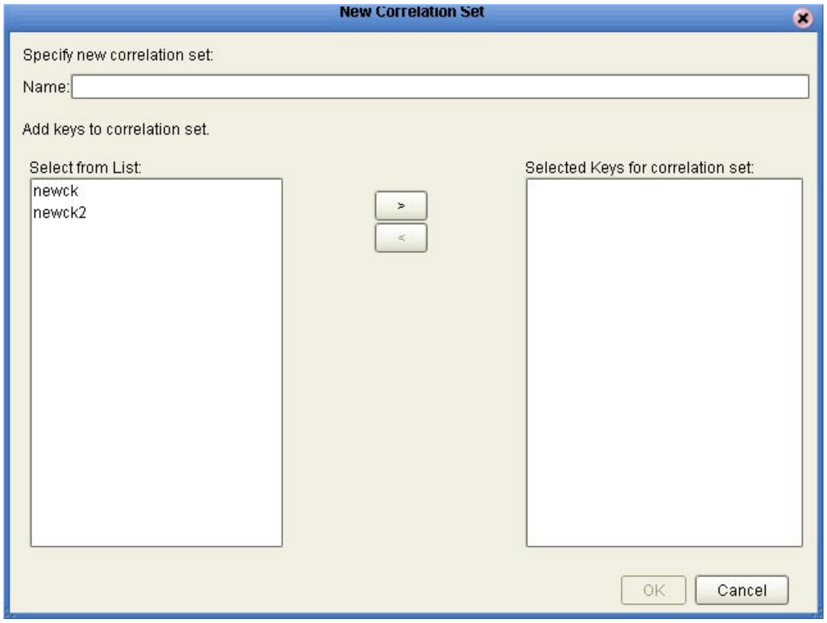
To Add Correlation Sets

1 Select the **Correlations** tab. See Figure 89.

2 In the **Correlation Set** section of the dialog box, select **New**.

The **New Correlation Set** dialog appears. See Figure 91.

Figure 91 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 In the **New Correlation Set** dialog box:

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

B To add to the Correlation Set, select **Correlation Keys** from the list.

C To move your selections to the Correlation Set, click the arrow button.

4 To save your changes and exit the New Correlation Set dialog box, click **OK**.

To bind Correlation Sets to Page Flow elements

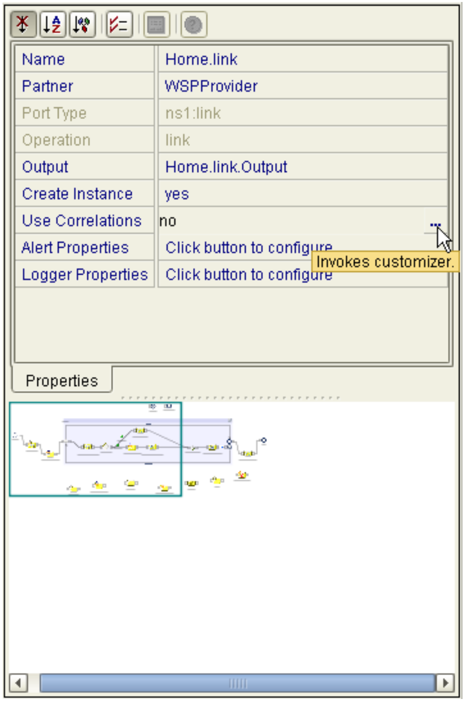
1 Select a Page Flow element.

2 On the Page Flow Designer toolbar, click the **Show Property** icon to open the **Element Properties** window.

3 Locate **Use Correlations**, and then click the **no** field.

4 In the **no** field, click the **Command (...)** button. See **Figure 92 on page 104**.

Figure 92 Page Flow Designer Element Properties Window



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

5 On the **Use Correlations** dialog box, click **Add**.

6 On the **Assign Correlation Sets** dialog box, in the left pane, select the Correlation Set that you want to add to the Page Flow element.

7 Click the arrow button to move it to **Selected Correlation Set(s)** window.

8 Click OK.

9 On the **Use Correlations** dialog box, click **OK**.

9.5.1. WSDL Files

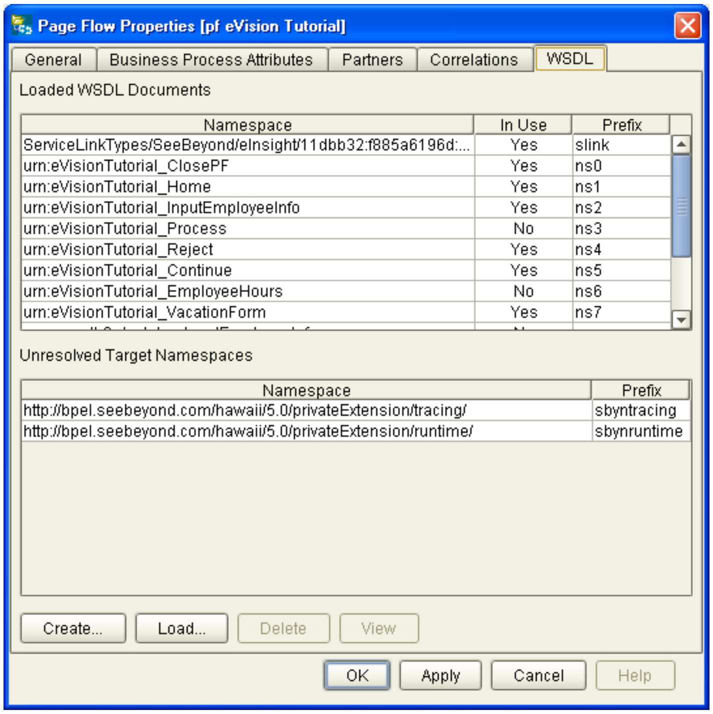
The Web Services Description Language (WSDL) is an XML -based language, which is used to describe business services. WSDL provides a way for individuals and other businesses to electronically access those services. In the Page Flow Designer, WSDL files are used to invoke and operate Web services on the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

The WSDL tab is available from the Page Flow Properties dialog. 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

1 On the **Page Flow Properties** dialog box, select the **WSDL** tab. See **Figure 93 on page 105**.

Figure 93 Page Flow Properties: WSDL Tab



Loaded WSDL Documents

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

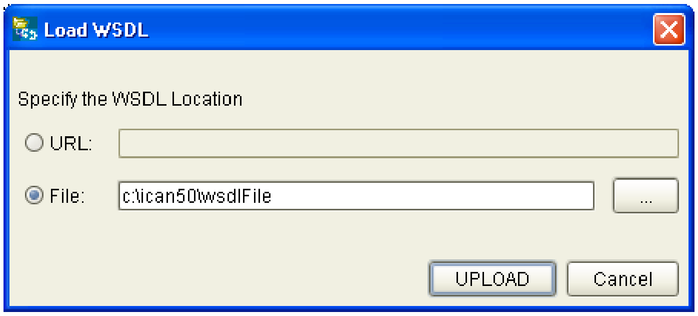
Unresolved Target Namespaces

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

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

The **Load WSDL** dialog box appears. See Figure 94.

Figure 94 Load WSDL Dialog Box



Specify the WSDL Location

○ URL:

● File: c:\ican50\wsdlFile

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

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

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

4 Click **UPLOAD**.

Chapter 10

Creating and Deploying a Web Application

This chapter describes how use the Page Flow Designer to create a Web application.

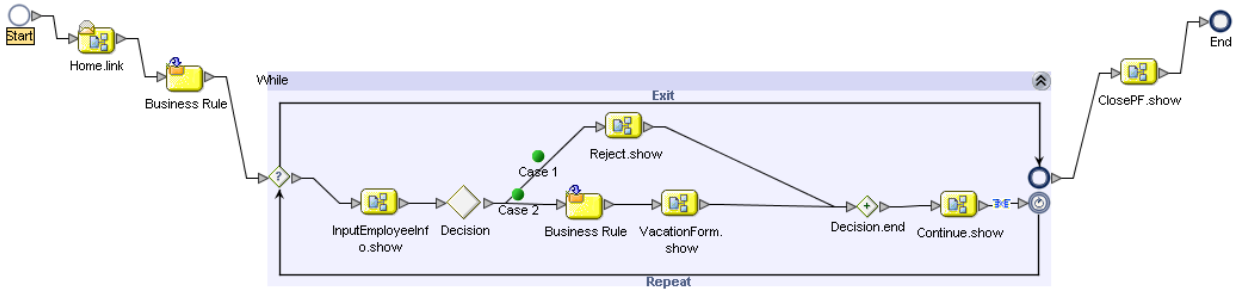
This Chapter Includes:

- **"Overview"**
- **"Assembling the Sample Project Components" on page 112**

10.1 Overview

This section provides an overview of importing, assembling, and running the sample Web application that is packaged with the eVision Studio product. Going through the Page Flow creation process allows you to work with the several of the most-used Page Flow Designer tools while assembling the necessary components to complete a Web application that you can deploy, then run on the Logical Host, and access with your browser. An example of the sample application is shown in Figure 95.

Figure 95 Sample Vacation Request Application



10.1.1 Creating the Sample Application

Creating the sample application consists of the following procedures:

- Downloading the sample project .zip files
- Extracting the contents of the archive files onto your computer
- Importing the sample project components into the Enterprise Designer
- Starting a new Page Flow Project

- Adding Page Flow Components to the Page Flow Designer canvas
- Adding pre-configured Web page components to the Page Flow Designer canvas
- Connecting the components
- Configuring the application logic
- Creating a Connectivity Map
- Creating an Environment
- Creating a Deployment Profile
- Activating, then running application
- Accessing the application Web pages with your browser

10.1.2 Downloading the Sample Project

Download the sample project only if you have not already done so. If you created the sample Web page as described in **"Laying Out a Web Page" on page 62**, you have already downloaded the files. Skip to **"Importing the Sample Project into the Repository" on page 108**.

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

The Repository Server must be running before you can use either the Enterprise Manager or the Enterprise Designer. The first step of any procedure is to make sure the server is running, and if not, to start it.

To Start the Repository Server

1. On your computer, explore to:

   ```
   <drive>\<path>\server\runServer.bat
   ```

   Where:

   drive is the drive letter where you installed eGate Integrator, and

   path is the directory path to the **\server** folder.

2. Double-click **runServer.ba**t.

   The command console will start to scroll the server startup messages. Wait until you see a message that confirms that the server is started.

To Download the Project Archive Files

Before you can import the sample project, you need to access and download the combined project archive files onto your computer.

1. Start your browser.
2. On the **Address** line, type the following:

   ```
   http://<localhost>:<portnumber>
   ```

   where:

localhost is the name of the machine that is hosting the Repository server, and

portnumber is the port number that was assigned to the server. For example:

http://localhost:12000

3 When the Enterprise Manager **HOME** page appears, click the **DOCUMENTATION** tab.

The Documentation Web page provides links to the documentation (PDFs), Javadoc(s), and code samples for the products you have installed.

Note: *Documentation is uploaded in a process separate from the product uploads. If you do not see a link to the eVision index page in the left frame, go to the ADMIN page and download the eVision documentation archive file.*

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

5 In the right frame of the Documentation page, click the **Sample** link to launch the archive utility and prepare to extract the sample archive files.

6 Using the archive utility (WinZip) dialog box, save the **eVision_Sample.zip** file in a folder on your machine. Remember where you saved the file.

7 Open the folder where you saved the archive file.

8 Double-click the file and extract the contents into any folder. You can use the archive program to create the folder.

The extraction process will place two .zip files in the designated folder.The extracted contents of the master archive file will appear as follows:

- **eVisionTutorial_sample.zip**

  This file is for reference and analysis after you finish successfully building the sample application from the pre-built components.

- **eVisionSampleComponents.zip**

  This file contains the pre-built Web pages that you will use to create the sample application.

Note: *Remember where you placed the extracted files. You will be importing the eVisionSampleComponents.zip file into the Enterprise Designer.*

10.1.3 Importing the Sample Project into the Repository

The **eVisionSampleComponents.zip** Project is imported using the Enterprise Designer. Make sure the Repository Server is running before you start the Enterprise Designer.

To Import the Sample Project Components

1 To start the Enterprise Designer, on your computer, explore to:

```
<drive>\<path>\edesigner\bin\runed.bat
```

Where:

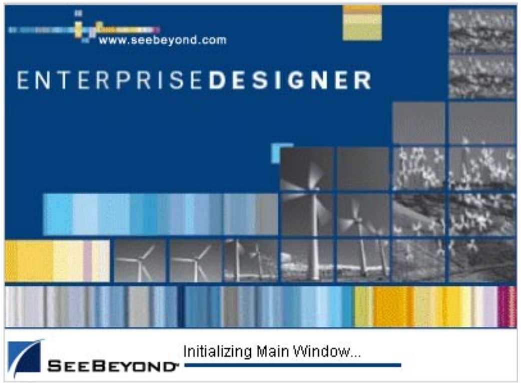
drive is the disk drive where you installed the Enterprise Designer, and

path is the directory path to the **\edesigner\bin** folder.

2 Double-click **runed.ba**t.

The Enterprise Designer startup window opens and scrolls the status of the startup process. See Figure 96.

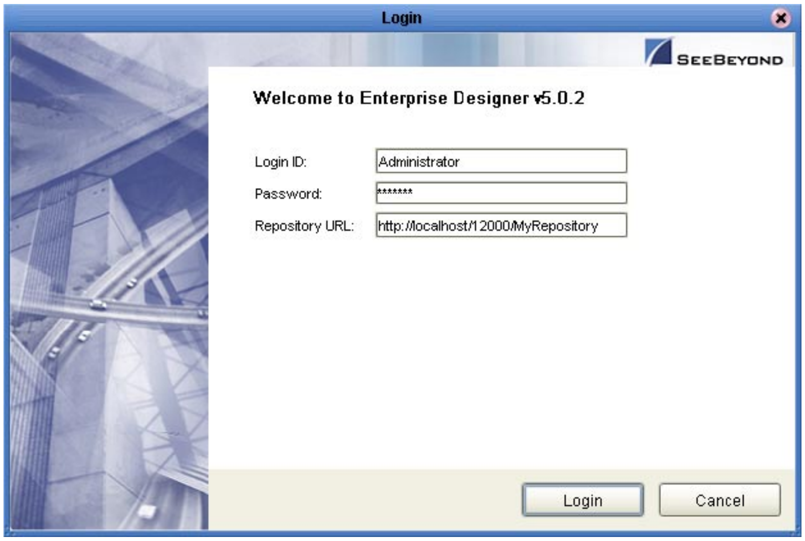
Figure 96 Enterprise Designer Startup Window



Initializing Main Window...

3 When the scrolling stops, the **Startup** window will be replaced by the **Login** window. See Figure 97.

Figure 97 Enterprise Designer Login Window



Welcome to Enterprise Designer v5.0.2

Login ID: Administrator

Password: *******

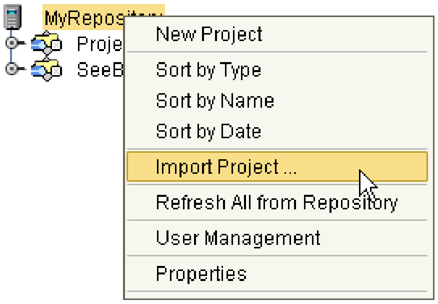
Repository URL: http://localhost/12000/MyRepository

4 On the **Login** screen, type your login ID and password, and then click **Login**.

5 When the Enterprise Designer has started, in the Enterprise Explorer, right click the **Repository** icon.

6 On the context menu, select **Import Project**. See **Figure 98 on page 110**.

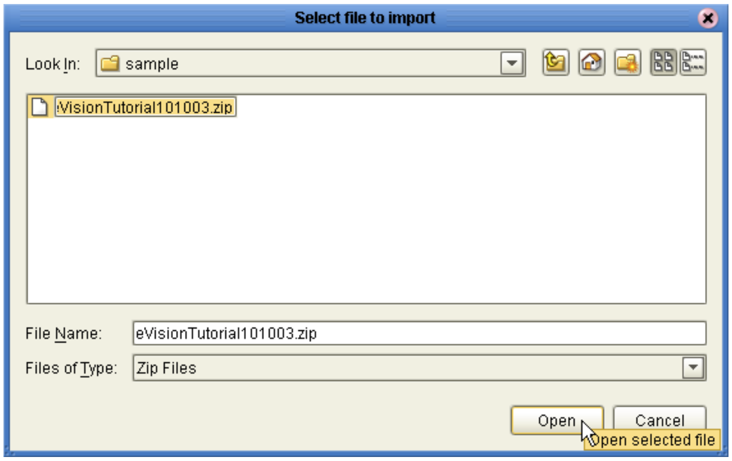
Figure 98 Import Project Menu



7 On the **Select file to import** dialog box, explore to the folder where you placed the extracted sample files.

8 Select the sample .zip file, **eVisionSampleComponents.zip**, and click **Open**. See Figure 99.

Figure 99 Selecting the Sample File to Import



File Name: eVisionTutorial101003.zip

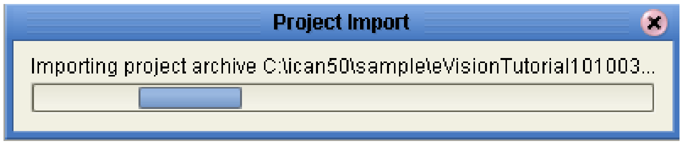
Files of Type: Zip Files

9 On the **File Destination** dialog box, select the **Import to a new project** radio button.

10 In the blank name box, type a new name for the project, for example, **eVisionSampleProject**.

11 Wait for the progress bar to stop. See Figure 100.

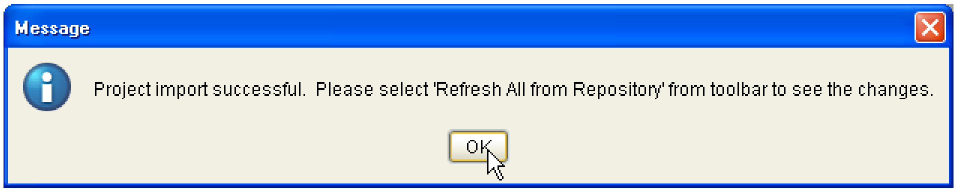
Figure 100 Import Progress Bar



Importing project archive C:\ican50\sample\eVisionTutorial101003...

When the import process is finished, a confirmation message will replace the progress bar. See **Figure 101 on page 111**.

Figure 101 Import Confirmation Message



Message

Project import successful. Please select 'Refresh All from Repository' from toolbar to see the changes.

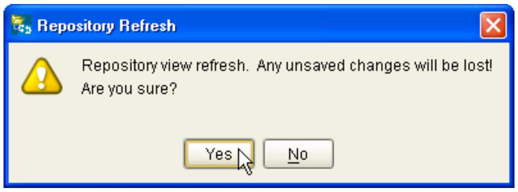
12 To dismiss the confirmation message, click **OK**.

13 Right-click the **Repository** icon.

14 On the context menu, select **Refresh All from Repository**.

15 Providing you have saved the changes made in any other active projects, on the **Repository view refresh** dialog box, click **Yes**. See Figure 102.

Figure 102 Repository Refresh Confirmation



Repository Refresh

Repository view refresh. Any unsaved changes will be lost!
Are you sure?

16 Wait for the progress bar to stop.

Importing the sample project is complete.

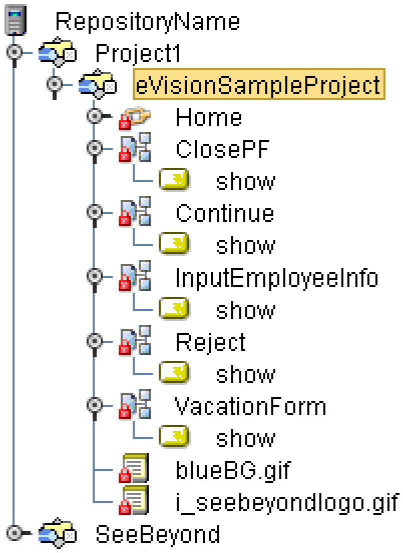
17 In the Project Explorer, double-click **eVisionSampleProject**.

The project will expand to reveal the components. Expand each component to view the contents.

In the Project Explorer, the sample project components should look the example shown in **Figure 103 on page 112** when it is finished.

Note: *Do not import the sample "Reference" project until you have finished creating the "SampleComponents" project.*

Figure 103 Sample Project Components in Enterprise Explorer



```
RepositoryName
  Project1
    eVisionSampleProject
      Home
      ClosePF
        show
      Continue
        show
      InputEmployeeInfo
        show
      Reject
        show
      VacationForm
        show
      blueBG.gif
      i_seebeyondlogo.gif
  SeeBeyond
```

10.1.4 Version Control: Checking Out the Project Components

Before you can manipulate the sample project components, you must complete the **Check Out** procedure using the Version Control system.

To Check Out the Project Components

Note: *Components must be checked out from the Version Control system one at a time.*

1. In the Enterprise Explorer, under **eVisionSampleProject**, right-click the **Page Flow** icon.
2. On the context menu, select **Check Out**.
3. Repeat the procedure to check out each component in the sample project.

10.2 Assembling the Sample Project Components

This section describes how to assemble the eVision components to create a Page Flow. Creating the Page Flow consists of the following procedures:

Step 1: Creating or importing a Project

Because the Project is already provided, you can skip this step.

Step 2: Create the Necessary Web Pages

Because the Web pages are already provided, you can skip this step.

Step 3: Create the Page Flow

In this procedure you create a new Page Flow, add the Page Flow elements, and then link them together to create a process. You also configure the Page Flow elements and links to process the data.

Step 4: Create a New Environment

In this procedure, you create the new run-time Environment for the Project.

Step 5: Create a Connectivity Map and Deployment Profile

The Connectivity Map enables the connections between the system components and the backend systems. The Deployment Profile is activated to put the finished Page Flow into production.

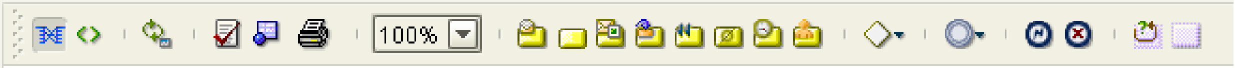
Step 6: Run and Test the System

The Bootstrap command reads the Deployment Profile, and executes the Page Flow.

10.2.1 Page Flow Designer Menu

In this section, you will use elements in the Page Flow Designer's menu to connect the various components in a Page Flow. These elements are described in detail in a previous chapter. See Figure 104.

Figure 104 The Page Flow Designer Menu



10.2.2 Creating the Page Flow

To Start a New Page Flow

1. In the Enterprise Designer, click the **Project Explorer** tab to switch to **Project** view.
2. In the Project Explorer tree, right-click **eVisionSampleProject**, and on the context menu, click **New**, and then select **Page Flow**.

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

Adding Page Flow Elements

To Add the Home Page Link and the While Loop Components

1. In the Project Explorer, expand the **Home Page** link, if necessary.

2 In the Project Explorer, drag the **Home Page** link Web service onto the canvas.

Place the **Home Page** link to the right of and below the **Start** element.

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

Place the **Business Rule** element to the right of and below the **Home Page** link.

4 From the Project Explorer, drag the **ClosePF.show** Page icon onto the canvas.

Place this Page to the left of and below the **End** element.

5 On the Page Flow Designer menu, drag a **While** element onto the canvas.

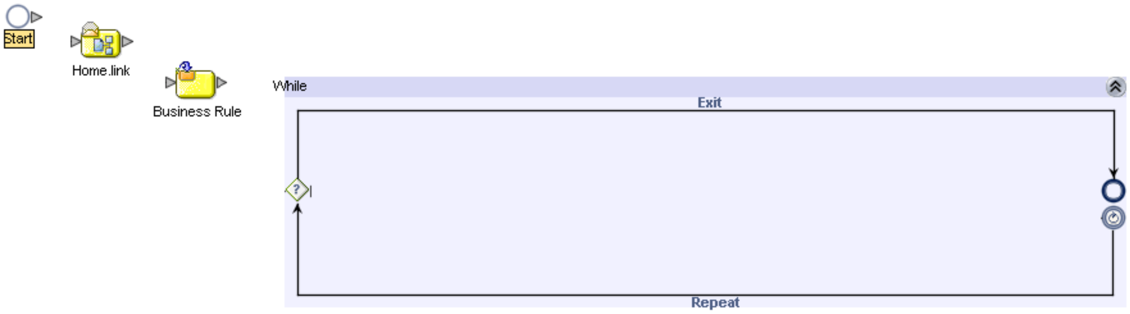
Place the **While** Loop in the center of the canvas.

6 Double-click the **While** Loop to expand it.

7 Check your progress.

The Page Flow should look like the example shown in Figure 105.

Figure 105 Starting the Page Flow



To Add the Page Flow Elements Inside the While Loop

1 In the Project Explorer, drag the **InputEmployeeInfo.show** Page icon onto the canvas.

Place the **InputEmployeeInfo.show** Page icon to the far left within the **While** Loop boundaries.

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

Place the **Decision** element to the right of, and aligned with the **InputEmployee.show** Page icon within the **While** loop.

3 On the Page Flow Designer menu, drag a **Business Rule** onto the canvas.

Place the **Business Rule** element to the right of and aligned with the **Decision** element within the **While** loop.

4 On the Project Explorer, drag the **VacationForm.show** Page icon onto the canvas.

Place this Page icon to the right of and aligned with the **Business Rule** element within the **While** loop.

5 On the Project Explorer, drag the **Continue.show** Page icon onto the canvas.

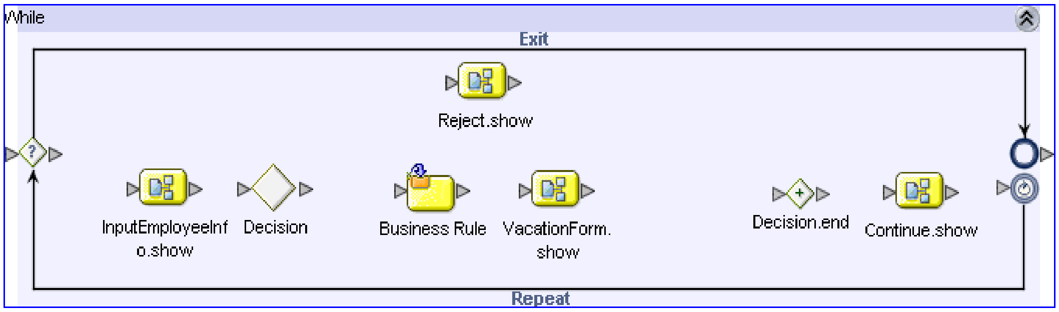
Place this Page icon to the right of and aligned with the **Decision.end** element within the **While** loop.

6 On the Project Explorer, drag the **Reject.show** Page icon onto the canvas.

Place this Page icon above the other elements and center it in **While** loop.

7 Check your progress. The **While** loop should look like the example shown in Figure 106.

Figure 106 Page Flow Elements in the While Loop



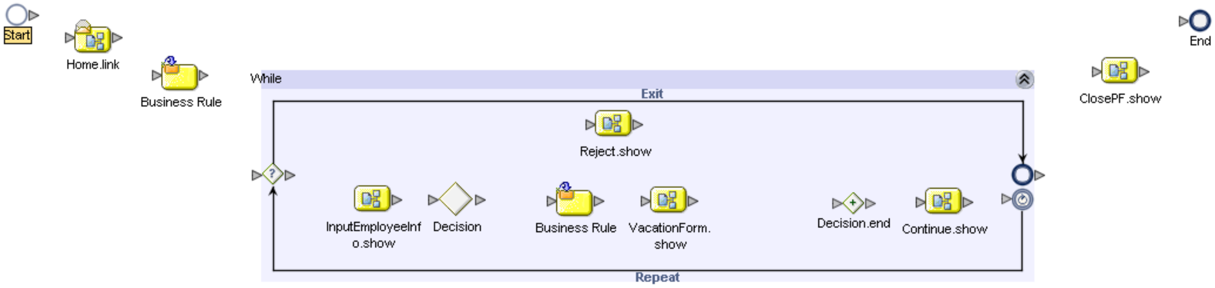
8 On the Project Explorer, drag the **ClosePF.show** Page icon onto the canvas.

Place this element to the right of, and outside the **While** loop. This element should be placed between the right boundary of the **While** loop and the **End** node.

9 Check your progress. The Page Flow should look like the example shown in Figure 107.

10 To adjust the size of the **While** loop, click and drag any element inside the loop boundaries.

Figure 107 Unconnected Page Flow Components



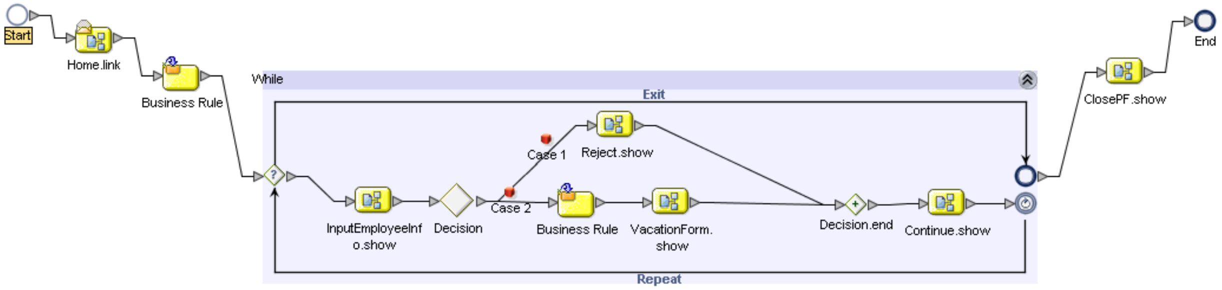
Connecting the Page Flow Components

To place a **Link** between elements, click the connector on the originating element and drag a connection line to the connector on the receiving element and release the mouse button

To Link the Page Flow Components

1. Hover over the arrow on the right edge of the **Start** icon until the pointer changes to the hand cursor.
2. Click and drag a connection line onto the receiving element and release the mouse button.
3. Proceeding from left to right, connect all the Page Flow components.
   - Create a branch link between the **Decision** element to the **Reject.show** page.
   - Create a branch link from the **Reject.show** page to the **Decision.end** element.
4. Check you progress. The Page Flow should look like the example shown in Figure 108.

Figure 108 Connected Page Flow Components



10.2.3 Configuring the Business Rule Elements

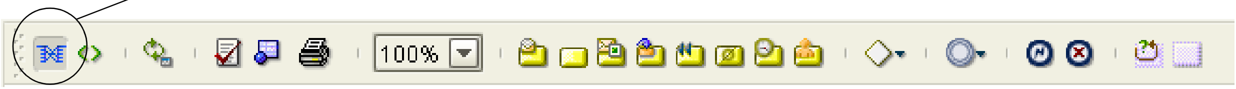
In the following procedures, you will use the **Business Rule Designer** to configure relationships between Input and Output Attributes. You will map outputs to inputs using an area called the *Mapper*. The Business Rule Designer window consists of three panes: **Output** (left), **Input** (right), and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

To Configure the Business Rules

1. To start the Mapper, on the Page Flow Designer toolbar, click the **Map Business Process Attributes** icon.

Figure 109 Starting the Business Rule Designer

Click **Map Business Process Attributes**



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

2. On the Page Flow Designer canvas, select the **Business Rule** that resides outside and to the left of the **While** loop.
3. In the Mapper **Input** pane, (the right pane) click **All**.

The Mapper **Output** and **Input** panes are populated.

4 In the Mapper **Input** pane, expand the **Continue.show.ouput** to expose the **radiogroup0** node.

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

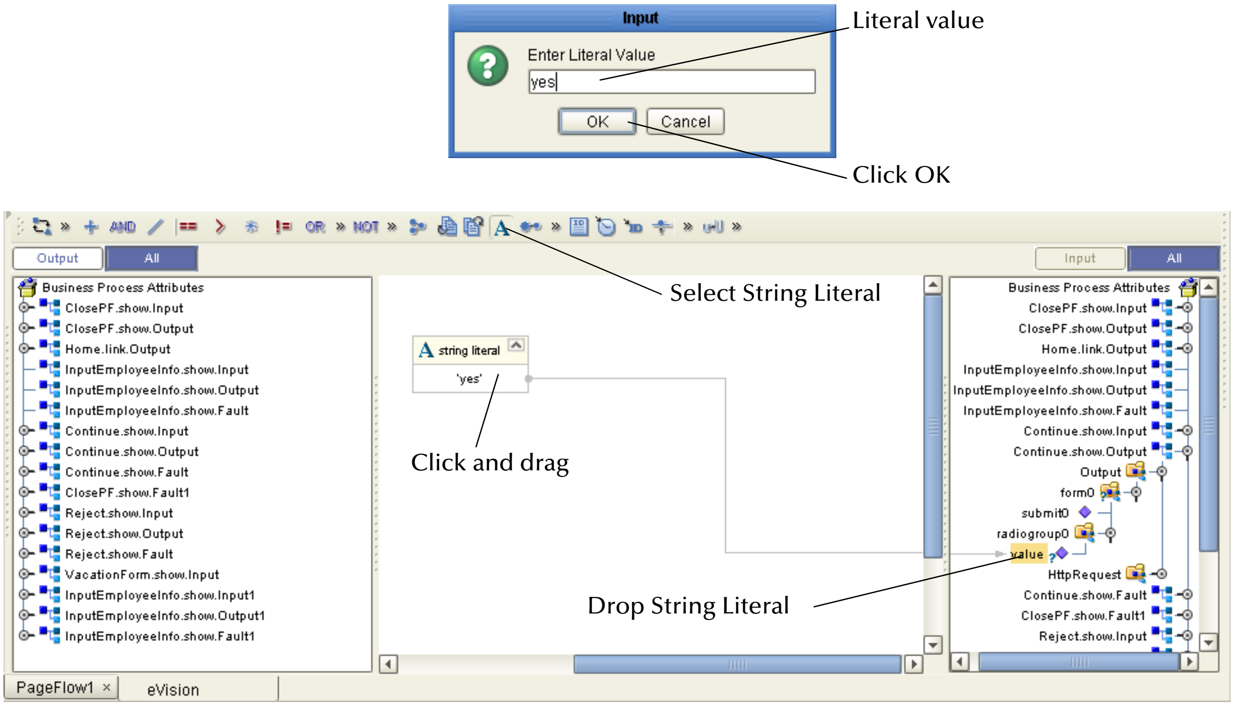
6 On the Business Rule Designer **Method** palette, drag the **String Literal** icon onto the Mapper canvas.

7 In the **Input** dialog box that appears, under the **Enter a Literal Value** heading, type **yes** (make sure the text is lower-case).

8 Click **OK**.

9 Select the **String Literal** container within the **"Yes"** box, drag it into the **Input** pane, and carefully drop is onto the **value** element text box. See Figure 110.

Figure 110 Mapping a String Literal



Input

Enter Literal Value: yes

Literal value

Click OK

Select String Literal

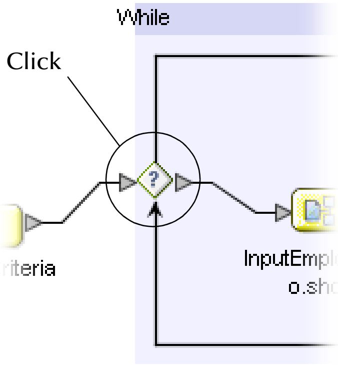
Click and drag

Drop String Literal

10.2.4 Configuring the While Loop

1 On the Page Flow Designer canvas, in the **While** loop, click the **While** icon. See **Figure 111 on page 118**.

Figure 111 While Loop Icon



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

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

4 On the Business Rule Designer **Method** palette, drag the **String Literal** icon onto the Mapper canvas.

5 In the **Input** dialog box that appears, under the **Enter a Literal Value** heading, type **yes** (make sure the text is lower-case).

6 Click **OK**.

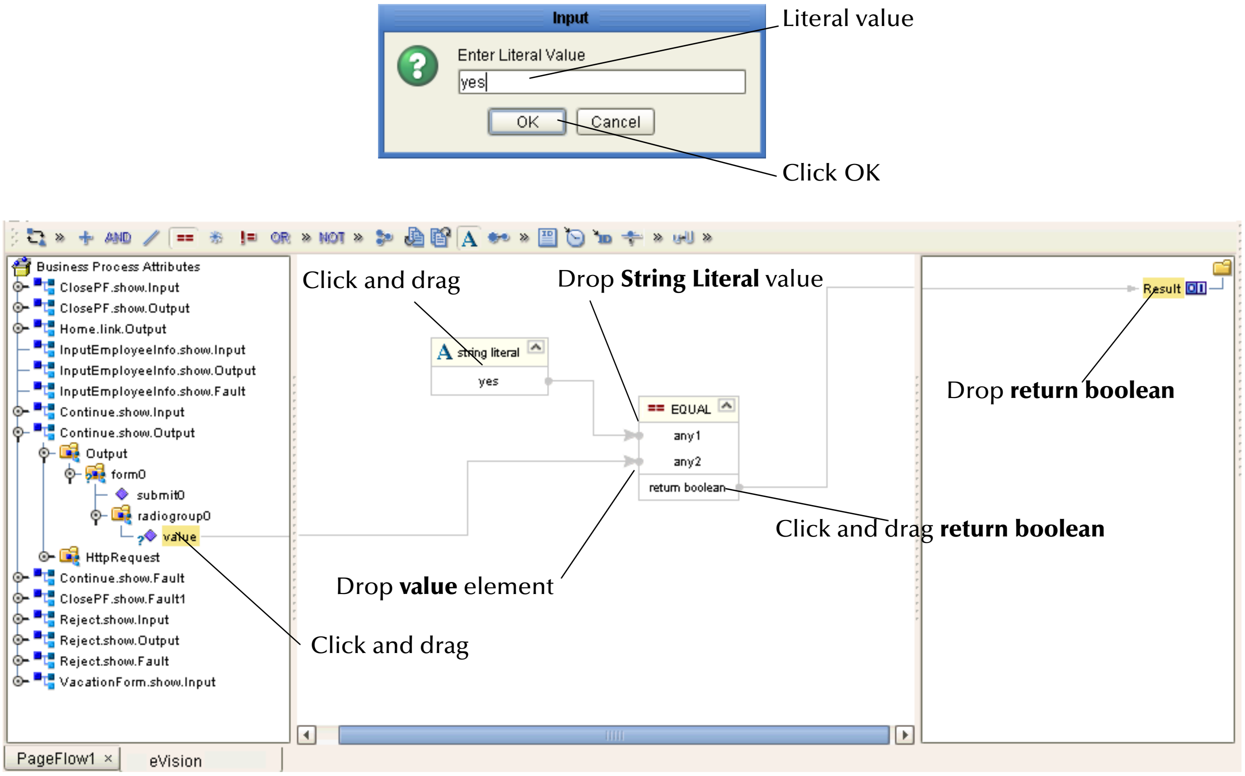
7 On the Business Rule Designer **Method** palette, drag the **EQUAL** icon (a “double-equal” sign) onto the Mapper canvas.

8 Click the **String Literal** container within the **“Yes”** box, and drag the cursor onto the **any1** connector node on the **EQUAL** container (the upper node on the left of the container).

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

10 Click the **EQUAL** container on the **return boolean** operator window, drag the **return boolean** element into the **Input** pane, and drop it onto the **Result** icon. See Figure 112 on page 119.

Figure 112 Mapping Within the While Loop



Literal value

Click OK

Click and drag

Drop **String Literal** value

Drop **return boolean**

Click and drag **return boolean**

Drop **value** element

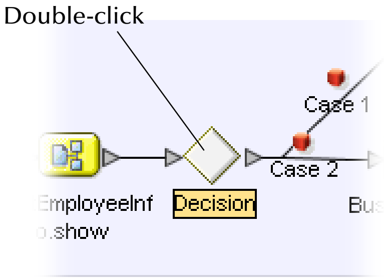
Click and drag

Configuring the Decision Logic

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

1 On the Page Flow Designer canvas, inside the **While** loop, double-click the **Decision Gate** element. See Figure 113.

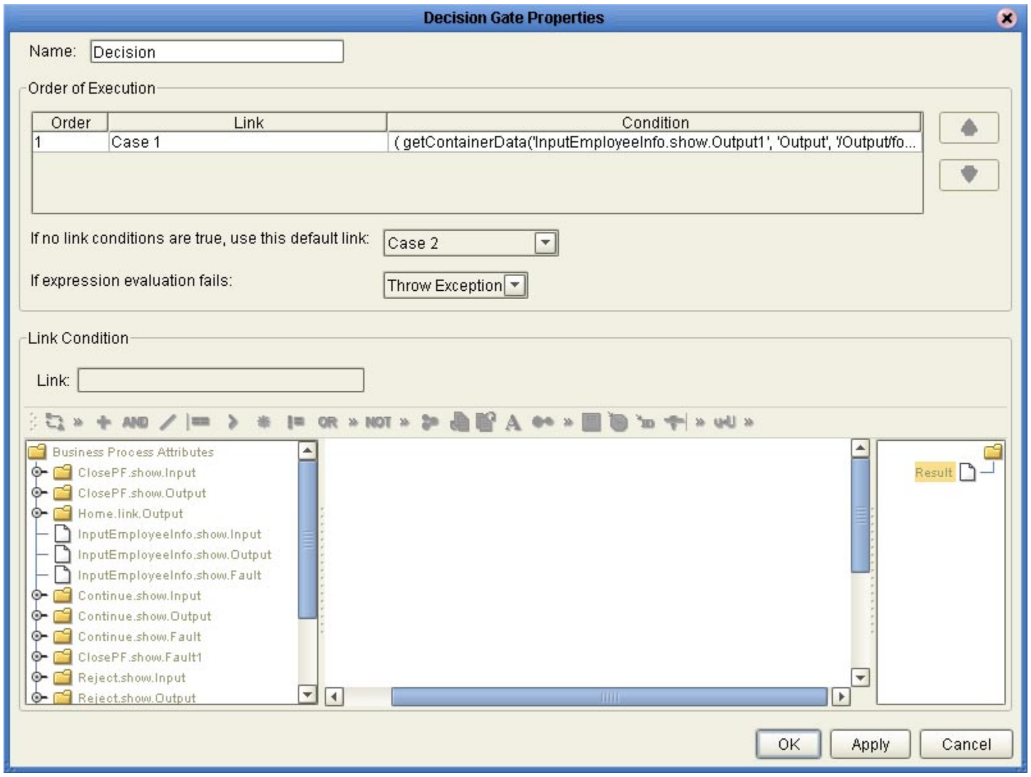
Figure 113 Decision Gate in While Loop



Double-click

The Decision Gate Properties window appears. See **Figure 114 on page 120**.

Figure 114 Decision Gate Properties Window



Decision Gate Properties

Name: Decision

Order of Execution

Order	Link	Condition
1	Case 1	(getContainerData('InputEmployeeInfo.show.Output1', 'Output', '/Output/fo...

If no link conditions are true, use this default link: Case 2

If expression evaluation fails: Throw Exception

Link Condition

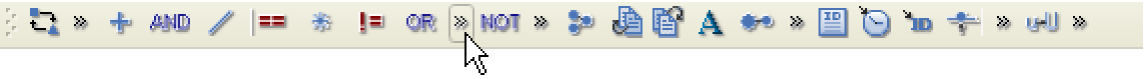
Link:

Business Process Attributes
- ClosePF.show.Input
- ClosePF.show.Output
- Home.link.Output
- InputEmployeeInfo.show.Input
- InputEmployeeInfo.show.Output
- InputEmployeeInfo.show.Fault
- Continue.show.Input
- Continue.show.Output
- Continue.show.Fault
- ClosePF.show.Fault1
- Reject.show.Input
- Reject.show.Output

2 On the **Decision Gate Properties** window, under the **Order of Execution** heading, click **Case1** to activate the Mapper.

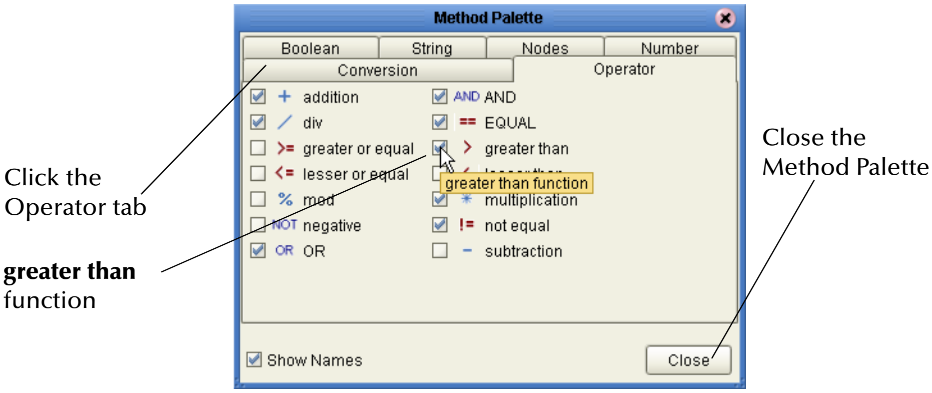
3 On the **Decision Gate Properties** window, to launch the **Method** palette, click the chevrons just to the right of the **OR** operator. See Figure 115.

Figure 115 Launching the Method Palette



4 On the **Method** palette, click the **Operator** tab to make the functions accessible. See Figure 116 on page 121.

Figure 116 Method Palette with Operator Tab



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

Click the Operator tab

greater than function

Close the Method Palette

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

6 On the **Method** palette, click **Close** to exit the palette and return to the **Decision Gate Properties** window.

Defining the Properties for Case 1

Case 1 tests for a valid employee ID number. There are no numbers greater than 5000. If a number equal to or greater than 500 is received, the user will get an error message and be prompted to try again.

To Define the Properties for Case 1

1 On the **Decision Gate Properties** toolbar, drag the **String Literal** icon onto the **Decision Gate Properties** Mapper.

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

If a number higher than 5000 is entered, the user will receive an error and be prompted to try again.

3 On the **Decision Gate Properties** toolbar, drag the **greater than** icon onto the Mapper in the **Decision Gate Properties** window.

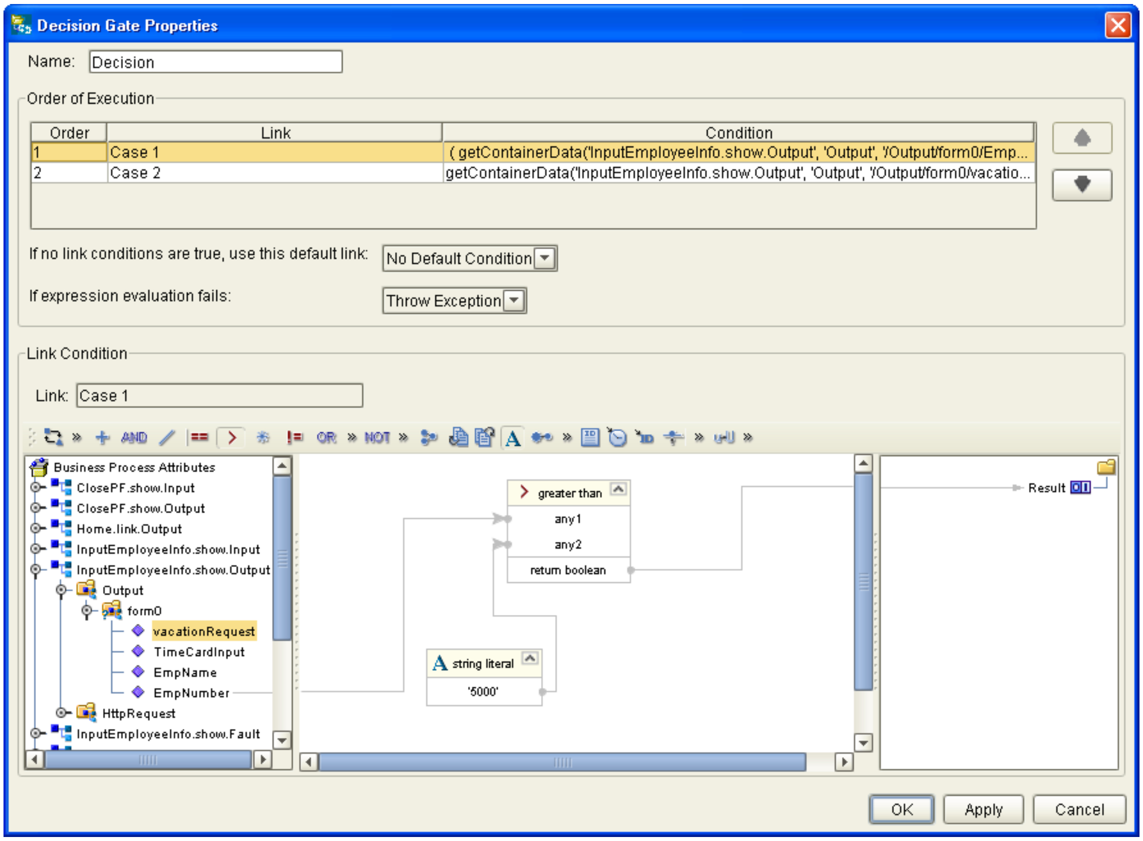
4 In the **Output** pane of the **Decision Gate Properties** window, expand the **InputEmployeeInfo.show.Output** node until the **EmpNumber** node is exposed.

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

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

7 Click the **greater than** container near the center of the **boolean** box, drag it into the **Input** pane, and drop it onto the **Resul**t icon. See Figure 117 on page 122.

Figure 117 Defining Decision Gate Properties: Case 1



Decision Gate Properties

Name: Decision

Order of Execution

Order	Link	Condition
1	Case 1	(getContainerData('InputEmployeeInfo.show.Output', 'Output', '/Output/form0/Emp...
2	Case 2	getContainerData('InputEmployeeInfo.show.Output', 'Output', '/Output/form0/vacatio...

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

If expression evaluation fails: Throw Exception

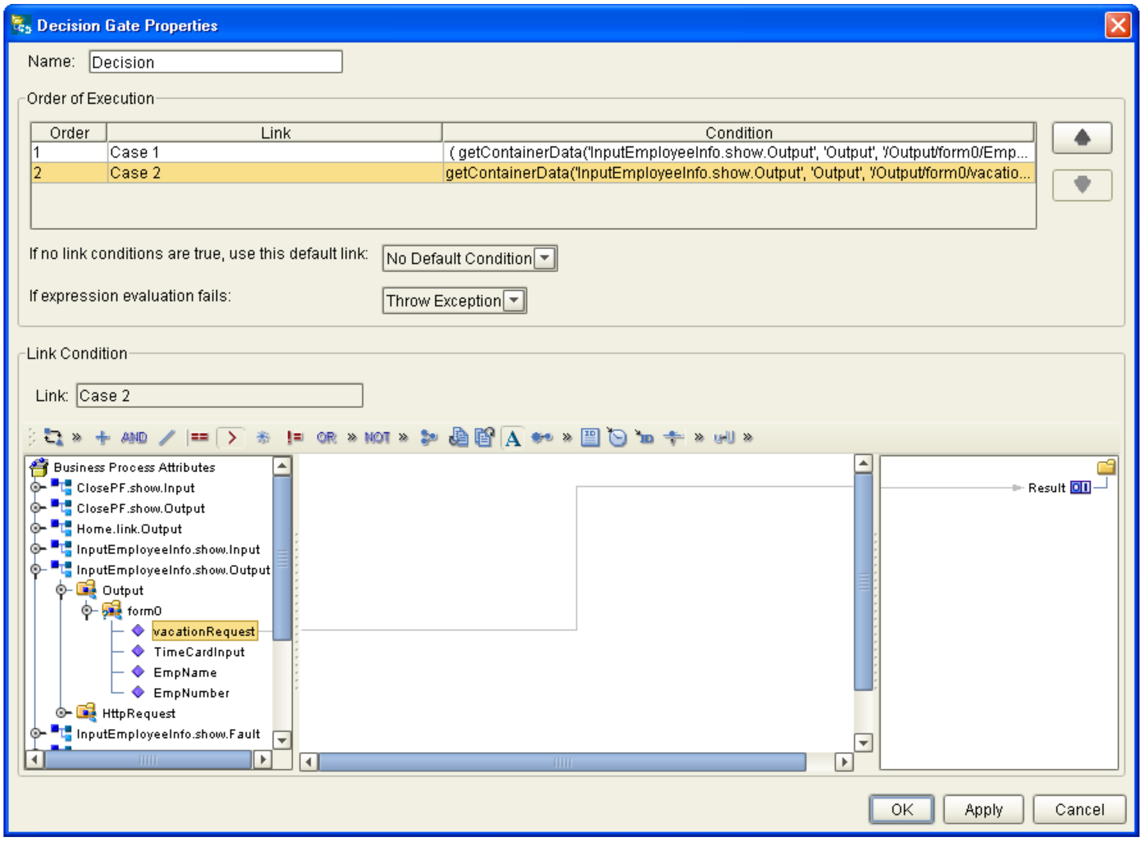
Link Condition

Link: Case 1

Defining the Properties for Case 2

1. On the **Decision Gate Properties** window, under the **Order of Execution** heading, click **Case2**.
2. In the **Output** pane, expand the **InputEmployee.show.Output** node until the **vacationRequest** node is exposed.
3. Click the **vacationRequest** node, drag it into the **Input** pane, and drop it onto the **Resul**t icon.
4. Click **Apply**.
5. To close the **Decision Gate Properties** window, click **OK**. See Figure 118 on page 123.

Figure 118 Defining Decision Gate Properties: Case 2



Decision Gate Properties

Name: Decision

Order of Execution

Order	Link	Condition
1	Case 1	(getContainerData('InputEmployeeInfo.show.Output', 'Output', '/Output/form0/Emp...
2	Case 2	getContainerData('InputEmployeeInfo.show.Output', 'Output', '/Output/form0/vacatio...

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

If expression evaluation fails: Throw Exception

Link Condition

Link: Case 2

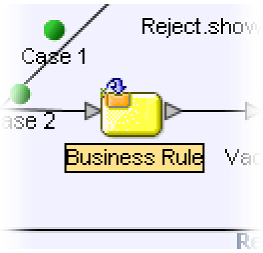
10.2.5 Mapping the Employee Name Output

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

To Map the Employee Name Output

1 On the Page Flow Designer canvas, inside the **While** loop, click the **Business Rule** element. See Figure 119.

Figure 119 Business Rule Element in the While Loop

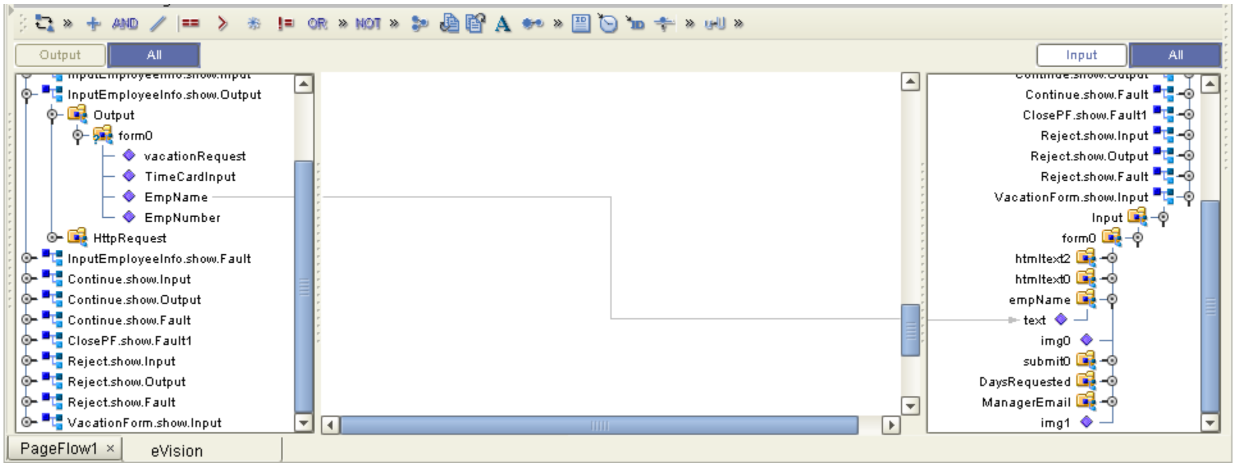


2 In the Mapper **Output** pane, expand the **InputEmployeeInfo.show.Output** node to to expose the **EmpName** node.

3 In the Mapper **Input** pane, expand the **Vacation.Form.show.Input** node to expose the **EmpName text** node.

4 Click the **Vacation.Form.show.Output** node, drag it onto the **Input** pane, and drop it onto the **text** node. See Figure 120.

Figure 120 Mapping Employee Name Output to Text Input



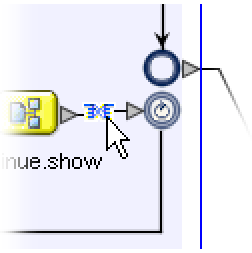
10.2.6 Mapping the Page Flow Attributes

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

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

2 Right-click the mouse and, on the context menu, click **Add Business Rule**. See Figure 121.

Figure 121 Adding a Business Rule



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

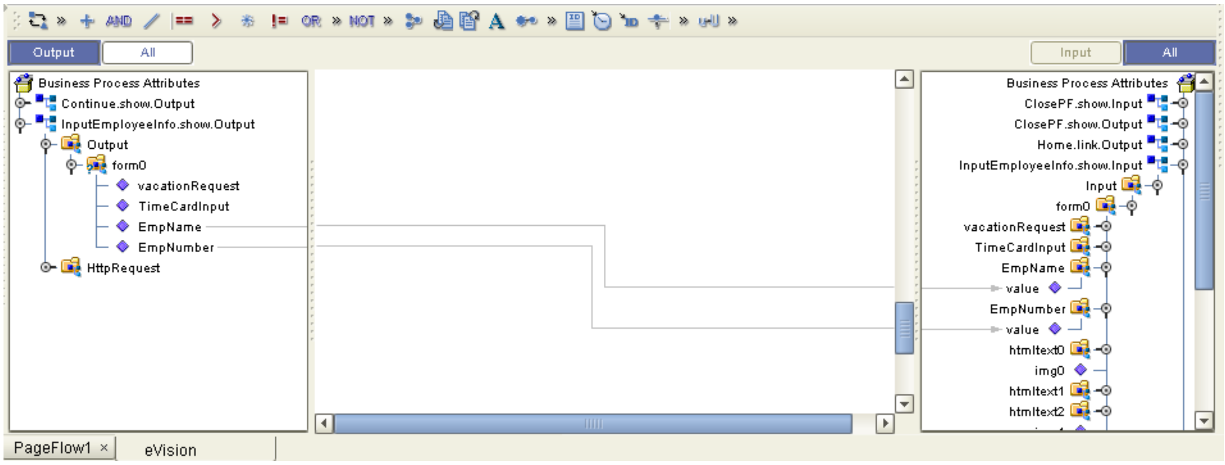
4 Expand the **InputEmployeeInfo.show.Ouput** node to expose the **EmpName** and **EmpNumber** nodes.

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

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

7 In the **Output** pane, click the **EmpNumber** node, drag it into the **Input** pane, and drop it onto the **value** node under the **EmpNumber** node. See Figure 122.

Figure 122 Output to Input Mapping



The mapping procedures are finished.

8 On the Enterprise Designer toolbar, click the **Save** icon.

This Page Flow is finished. The next procedure is to create a Connectivity Map.

10.2.7 Creating the Connectivity Map

The Connectivity Map comprises an **eGate Service** and a **Web Connector**. These components enable the project to run on the Logical Host.

To Create the Connectivity Map

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

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

2 On the **Connectivity Map Editor** (**CME**) toolbar, select the **Service** icon, and drag it onto the Enterprise Designer canvas.

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

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

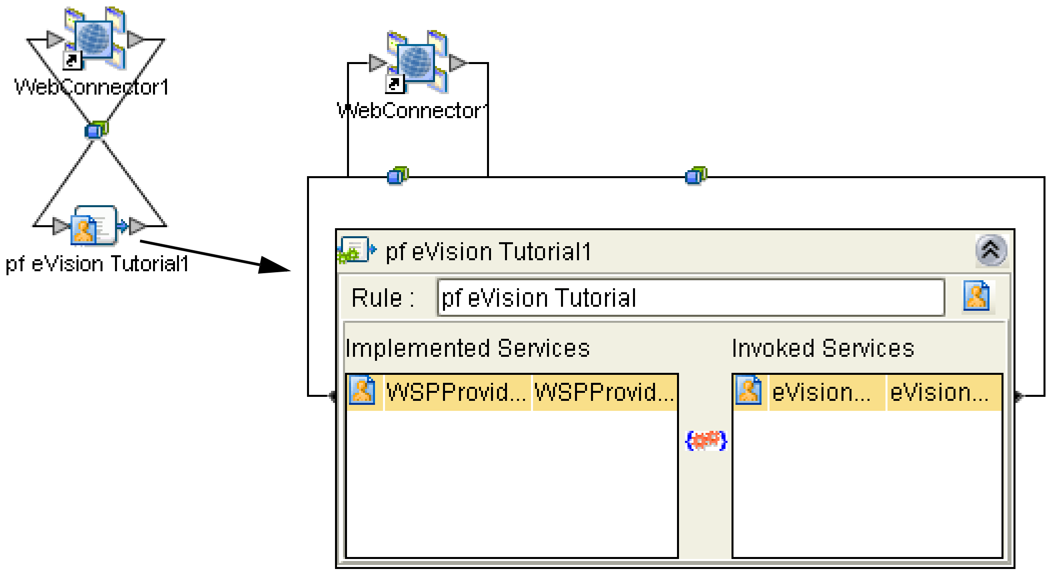
4 On the **Connectivity Map Editor** toolbar, drag a **Web Connector** icon onto the canvas.

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

6 Connect the arrows to the **Web Connector** icon.

The connectors will appear crossed, which is normal. See Figure 123.

Figure 123 New Connectivity Map



10.2.8 Creating the Environment

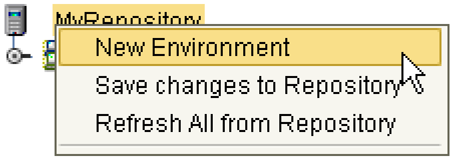
In this procedure, you will create the run-time environment for the eVision application which consists of:

- A Logical Host (the run-time platform)
- An Integration Server (provides run-time services for the application)
- An eVision External System

To Create the Environment

1. In the Enterprise Designer, click the **Environment Explorer** tab.
2. In the Enterprise Explorer, right-click the **Repository** icon and on the context menu, select **New Environment**. See Figure 124.

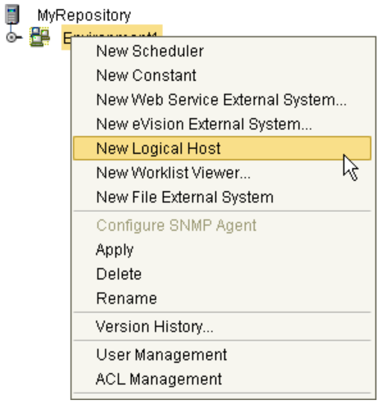
Figure 124 Creating a New Environment



The **Environment** name defaults to **Environment1**. You can accept the default name.

3 Right-click the new **Environment** and on the context menu, select **New Logical Host**. See Figure 125.

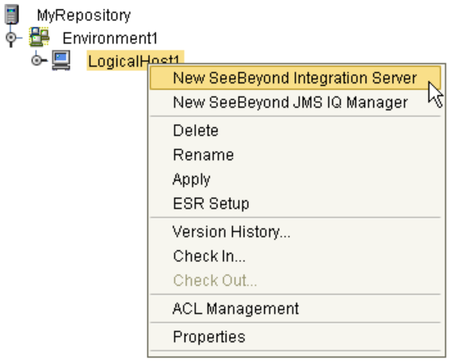
Figure 125 Creating a New Logical Host



The **Logical Host** name defaults to **LogicalHost1**. You can accept the default name.

4 Right-click the **Logical Host** and on the context menu, select **New SeeBeyond Integration Server**. See Figure 126.

Figure 126 Creating a New Integration Server



The **Integration Server** name defaults to **IntegrationSvr1**. You can accept the default name.

5 Right-click the **Environment** and on the context menu, select **New eVision External System**.

6 In the dialog box, type a name for the eVision External System, and **OK**.

7 On the Enterprise Designer toolbar, click **Save**.

10.2.9 Creating the Deployment Profile

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

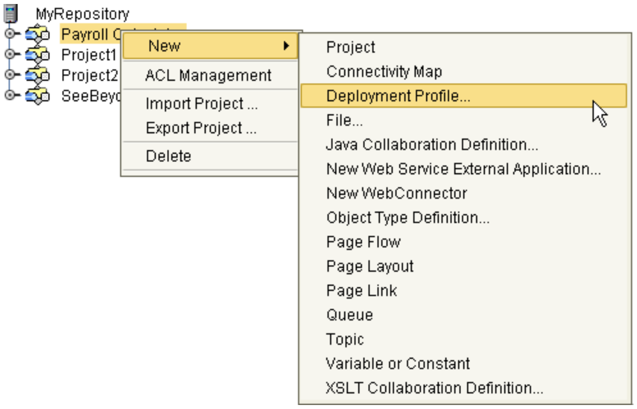
To Create the Deployment Profile

1 At the bottom of the Enterprise Explorer window, click the **Project Explorer** tab.

2 In the Project Explorer, right-click the **eVisionSampleProject** icon.

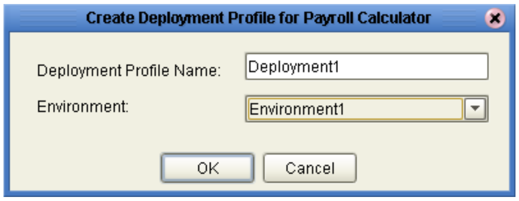
3 Click **New**, and on the context menu, select **New Deployment Profile**. See Figure 127.

Figure 127 Creating a new Deployment Profile



4 The **Create Deployment Profile** dialog box appears. See Figure 128.

Figure 128 New Deployment Profile Dialog Box


Create Deployment Profile for Payroll Calculator

Deployment Profile Name: Deployment1

Environment: Environment1

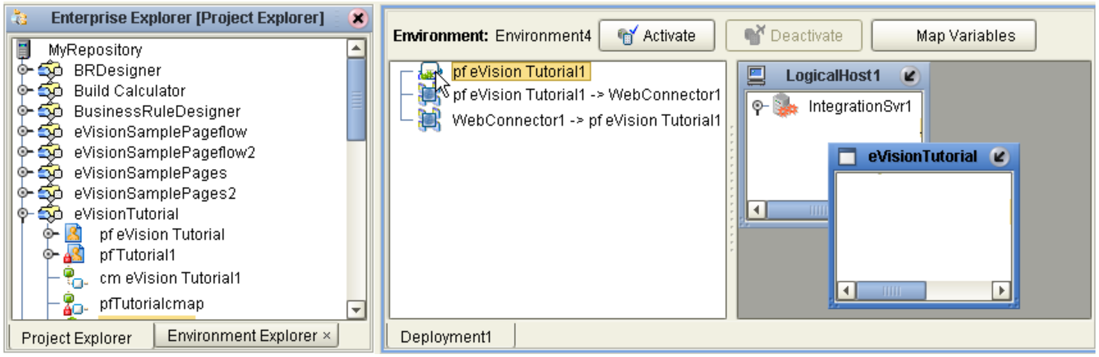
- The Deployment Profile defaults to **Deployment1**. You can accept the default name.
- The Environment defaults to **Environment1**. You can accept the default name, or, if there is more than one Environment, click the drop-down arrow, and select the Environment name you just created from the Environment list.

5 To finish, click **OK**.

10.2.10Configuring the Deployment Profile

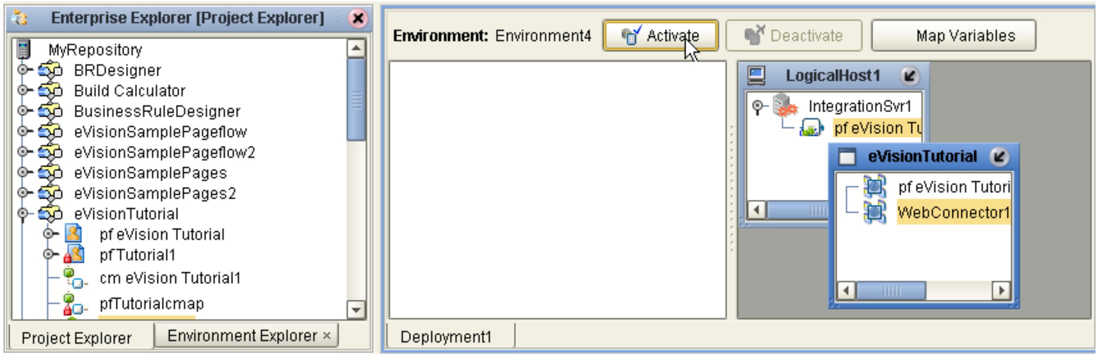
1 In the **Environment** canvas, click the **Page Flow** icon, drag it into the **Logical Host** window, and drop it onto the **Integration Server** icon. See Figure 129 on page 129.

Figure 129 Configuring the Deployment Profile



2 One at a time, drag both **Web Connector** icons onto the **eVision External System** window. See Figure 130.

Figure 130 Activating the Deployment Profile

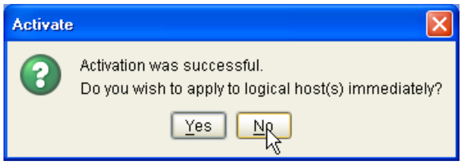


3 On the **Environment** canvas, click **Activate**.

4 On the **Activate** dialog box, click **No**.

Click **No** because the **Logical Host** is not yet running. If you click **Yes**, the operation will fail. You will start the Logical Host in another operation. See Figure 131.

Figure 131 Do Not Start the Logical Host

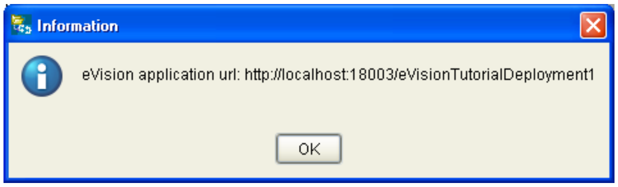


Activation was successful.
Do you wish to apply to logical host(s) immediately?

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

The **Information** window displays the URL for the eVision application. Remember the URL You will need to enter the URL on the address line in your browser to access the Web application. See Figure 132 on page 130.

Figure 132 eVision Project Startup URL



Information

eVision application url: http://localhost:18003/eVisionTutorialDeployment1

The URL will have the form:

http://<***servername***>:<***portnumber***>/<***projectname***><***deploymentprofilename***>.

For example:

http://localhost/18003/eVisionSampleProjectDeployment1

10.2.11Running and Testing the System

The Bootstrap process executes your **eVisionSampleProjec**t and starts the process of polling for input data. Bootstrap will pick up the deployment profile the first time it runs; after that, you select **reactivate** and click **Yes** to apply the most recent changes to the Logical Host.

How to Start the Logical Host

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

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

For example, the command

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

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

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

Logical Host Startup Parameters

The startup command syntax is as follows:

```
bootstrap arguments
```

Table 11 Command Arguments for bootstrap

Parameter	Description	Req/Opt
-d *debug*	Overrides bootstrap sequence. Displays all cached (default) argument values	Optional

Table 11 Command Arguments for bootstrap (Continued)

Parameter	Description	Req/Opt
-e *environment name*	The name of the Environment to which this Logical Host belongs.	Required (first time only)
-h *help*	Overrides bootstrap sequence. Displays the usage report.	Optional
-i *id*	The user ID used for accessing the Repository. Note that the user ID is the same as the username, and that the Administrator can set up more than one user ID.	Required (first time only)
-l *logicalhost name*	The name of this Logical Host.	Required (first time only)
-n *physical host name*	The name of this Physical Host.	Required (first time only)
-p *password*	The password used for accessing the Repository.	Required (first time only)
-r *repository URL*	The root URL for the Repository containing the Logical Host data.	Required (first time only)

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

A typical bootstrap command would look like the following:

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

Where:

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

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

repository_URL is the full URL of your Repository including the Repository name (for example, *http://localhost:10000/MyRep*),

username is your user name, and

password is your password.

10.2.12Starting the Logical Host

Before you begin

Make sure that the Integration Server on the Logical Host has been started. The Logical Host will not start if the Integration Server is not already running.

To start the Logical Host

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

   c:\ican50\logicalhost\bootstrap\bin

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

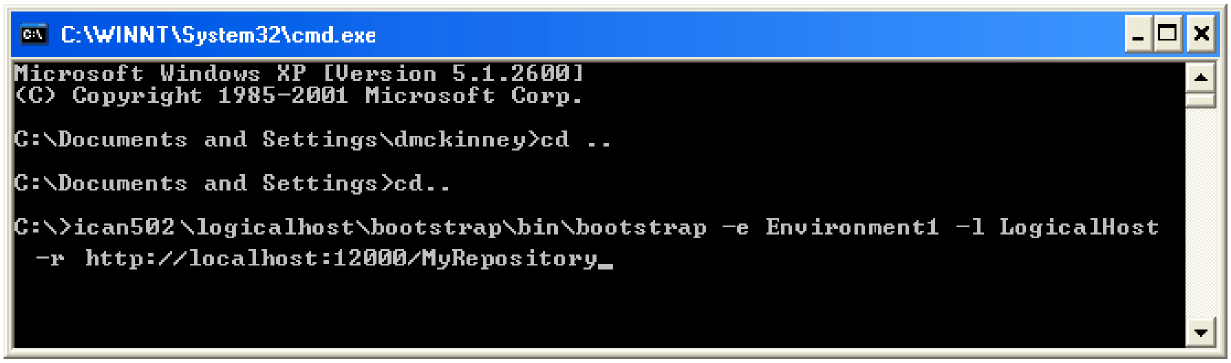
   - If you are starting the Logical Host for the first time, run the bootstrap command and include all of the required parameters described in **"Command Arguments for bootstrap" on page 130**.
   - An example of the bootstrap command in a Command Prompt is shown in **Figure 133 on page 132**.
   - If you are starting the Logical Host after having started it at least once previously, do the following:

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

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

Note: *On the Windows Operating System, the Bootstrap command is case-sensitive*

Figure 133 Invoking the Bootstrap Command



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\dmckinney>cd ..

C:\Documents and Settings>cd..

C:\>ican502\logicalhost\bootstrap\bin\bootstrap -e Environment1 -l LogicalHost
 -r http://localhost:12000/MyRepository_
```

10.3 Accessing the Web Application

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

To Access the Web Application

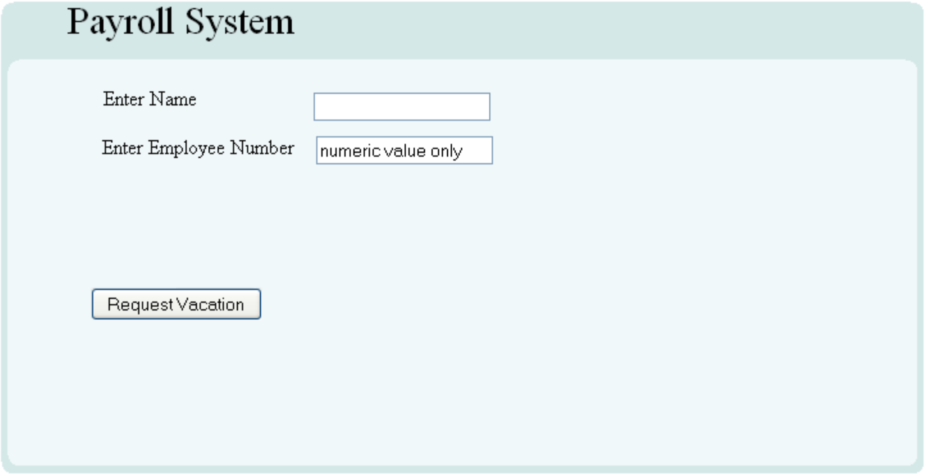
1. Make sure the Repository server is started and running.
2. Start your Browser.
3. On the **address** line, type the URL for the application, for example:

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

   portnumber is the port number assigned to the application, for example: **18003**.

The first page should look like the example shown in Figure 134.

Figure 134 Sample Web Application Startup Page



Payroll System

Enter Name

Enter Employee Number numeric value only

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

10.3.1 Importing the Working Sample Project

When you are satisfied with the "starter" Project, **eVisionSampleProjec**t, you can import the sample "reference" Project, **eVisionTutorial_sample**. This Project is more complex than the "starter" project, and it is completely finished and ready for deployment. After importing it, you can analyze its composition before deploying it onto a Logical Host. Once deployed, you can access it in your browser, and work through the finished application from a user's perspective. You can also add to and modify the Project when you feel you are ready.

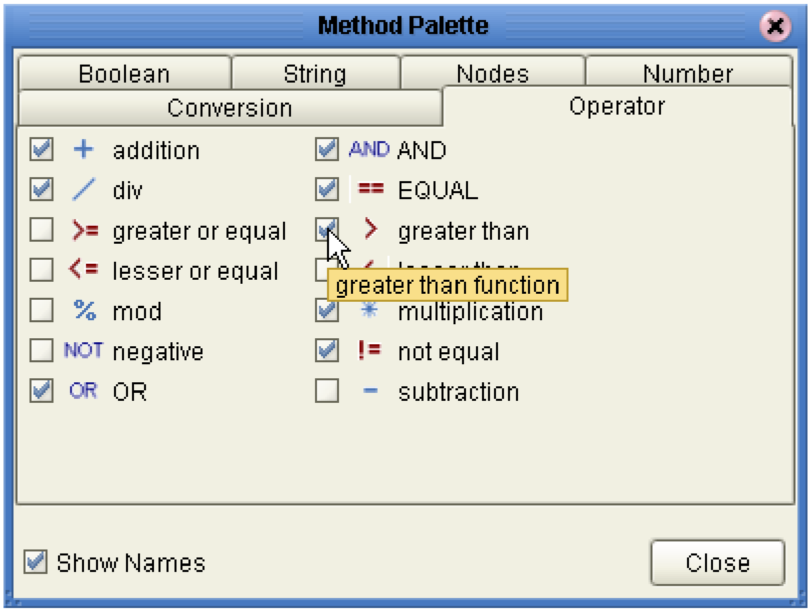
Important: *When you import **eVisionTutorial_sample**, you must give it the project name: **eVisionTutorial**. If you use another name, some of the links may break when you open it in the Page Flow Designer.*

Appendix A. Methods

Method Palettes

A.1 Operators

Figure 135 Method Palette: Operator tab



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

greater than function

☑ Show Names

Table 12 Operator Methods

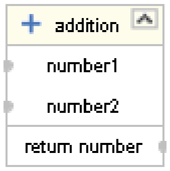
Method Box	Name	Description/Usage
	addition	Adds the value of *number1* to the value of *number2*, returns the sum.

Table 12 Operator Methods (Continued)

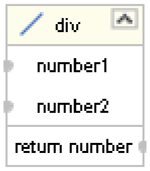
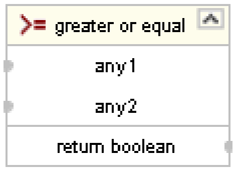
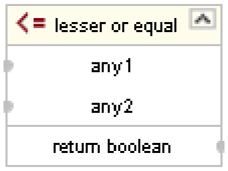
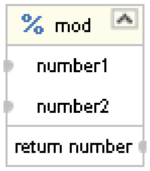
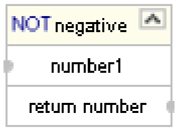
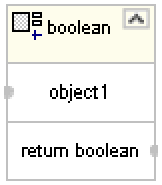
Method Box	Name	Description/Usage
	division	Divides the value of *number1* by the value of *number2*, returns the quotient.
	greater_or_equal	Returns Boolean true if *number1* is greater than or equal to *number2*; otherwise, returns Boolean false.
	lesser_or_equal	Returns Boolean true if *number1* is less than or equal to *number2*; otherwise, returns Boolean false.
	mod	Divides number1 by number2 and returns only the remainder.
	negative	Converts the input number to negative. Result is a negative number having the same absolute value as the input number.
	or	Returns Boolean false if both *boolean1* and *boolean2* are false; otherwise, returns Boolean true.

Table 12 Operator Methods (Continued)

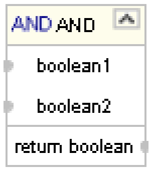
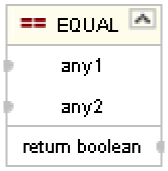
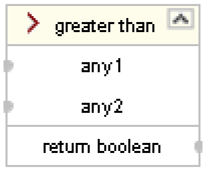
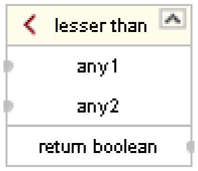
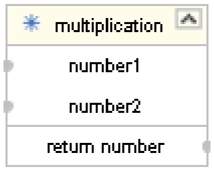
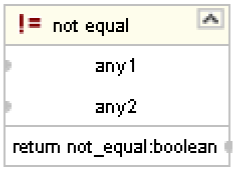
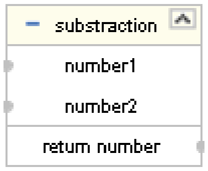
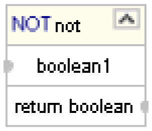
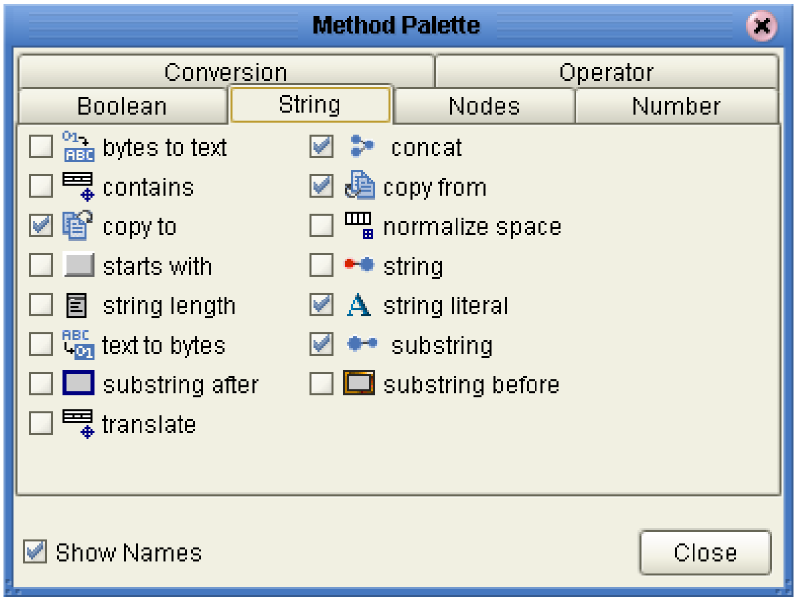
Method Box	Name	Description/Usage
 AND AND boolean1 boolean2 return boolean	and	Returns Boolean true if both *boolean1* and *boolean2* are true; otherwise, returns Boolean false.
 == EQUAL any1 any2 return boolean	equal	Returns Boolean true if *number1* is equal to *number2*; otherwise, returns Boolean false.
 > greater than any1 any2 return boolean	greater_than	Returns Boolean true if *number1* is greater than *number2*; otherwise, returns Boolean false.
 < lesser than any1 any2 return boolean	less_than	Returns Boolean true if *number1* is less than *number2*; otherwise, returns Boolean false.
 * multiplication number1 number2 return number	multiplication	Multiplies the value of *number1* by the value of *number2*, returns the product.
 != not equal any1 any2 return not_equal:boolean	not_equal	Returns Boolean true if *number1* is not equal to *number2*; otherwise, returns Boolean false.

Table 12 Operator Methods (Continued)

Method Box	Name	Description/Usage
	subtraction	Subtracts the numerical value of *number2* from the numerical value of *number1*, returns the difference.
	not	Returns the inverse of *boolean1*.

A.2 String

Figure 136 Method Palette: String tab



☐ bytes to text
☐ contains
☑ copy to
☐ starts with
☐ string length
☐ text to bytes
☐ substring after
☐ translate

☑ concat
☑ copy from
☐ normalize space
☐ string
☑ string literal
☑ substring
☐ substring before

☑ Show Names

Table 13 String Methods

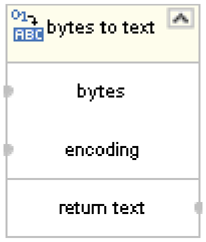
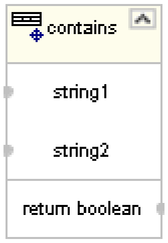
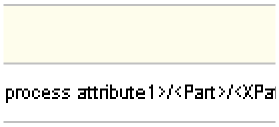
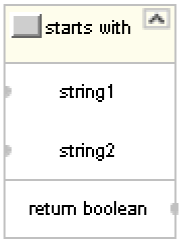
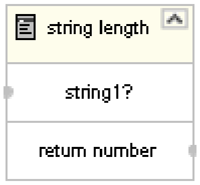
Symbol	Name	Description
	bytes to text	Decodes bytes into text using the specified encoding. If no encoding is specified, the platform's default encoding is used.
	contains	Returns true if the second string is contained within the first string, otherwise it returns false
	copy to	Allows you to type in the xpath expression for the destination of a copy operation. This is useful for entering xpath predicates. Note: This is for advanced users who are familiar with xpath and BPEL syntax.
	starts with	Returns true if the first string starts with the second string, otherwise it returns false
	string length	Returns the number of characters in a string

Table 13   String Methods (Continued)

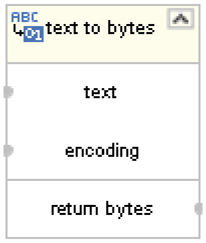
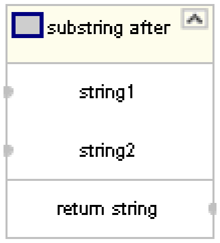
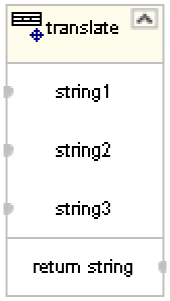
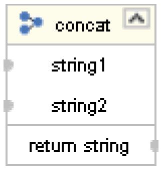
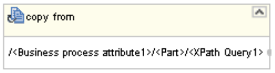
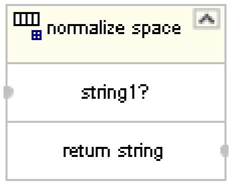
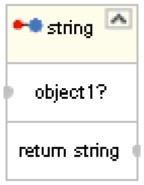
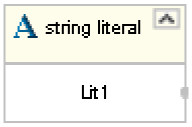
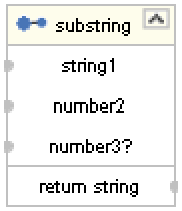
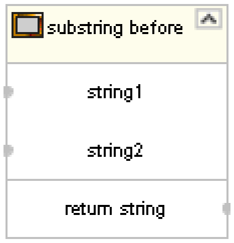
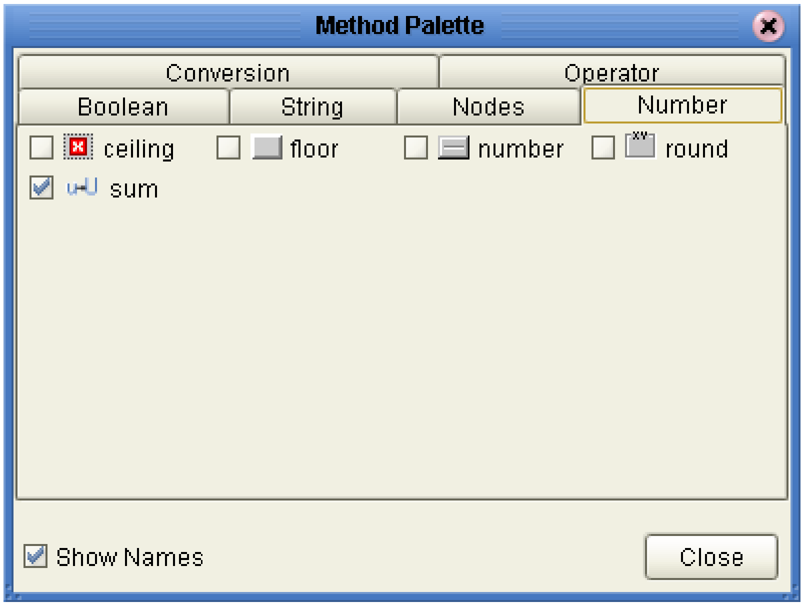
Symbol	Name	Description
 ABC 01 text to bytes text encoding return bytes	text to bytes	Encodes the input text into a sequence of bytes using the specified encoding. If no encoding is specified, the platform's default encoding is used
 	substring after	Returns the part of the string in the string argument that occurs after the substring in the substring argument
 	translate	Performs a character by character replacement. It looks in the value argument for characters contained in string1, and replaces each character for the one in the same position in the string2
 	concat	Returns the concatenation of all its arguments
 	copy from	Allows you to type in xpath expression for the source of a copy operation. This is useful for entering xpath predicates. Note: This is for advanced users who are familiar with xpath and BPEL syntax

Table 13 String Methods (Continued)

Symbol	Name	Description
	normalize space	Removes leading and trailing spaces from a string
	string	Converts the value argument to a string
	string literal	A sequence of characters of fixed length and content
 substring string1 number2 number3? return string	substring	Returns a part of the string in the string argument
	substring before	Returns the part of the string in the string argument that occurs before the substring in the substring argument.

A.3 Number

Figure 137 Method Palette: Number tab



□ ceiling □ floor □ number □ round
☑ sum

☑ Show Names

Table 14 Number Methods

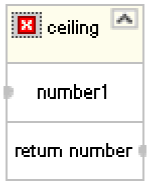
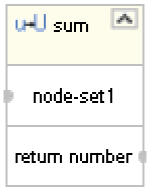
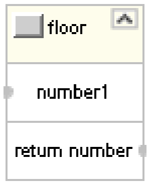
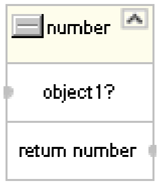
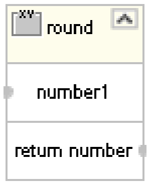
Symbol	Name	Function
	ceiling	Returns the smallest integer that is not less than the number argument
	sum	Returns the total value of a set of numeric values in a node-set

Table 14   Number Methods (Continued)

Symbol	Name	Function
	floor	Returns the largest integer that is not greater than the number argument
	number	Converts the value argument to a number
	round	Rounds the number argument to the nearest integer

A.4 Boolean

Figure 138 Method Palette: Boolean tab

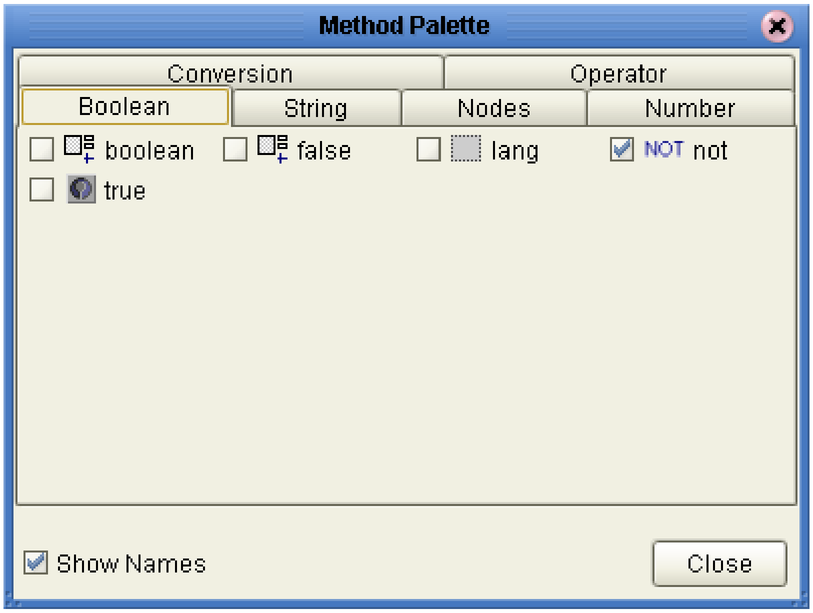


Table 15 Boolean Methods

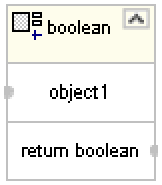
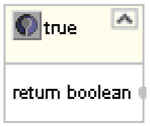
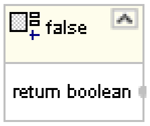
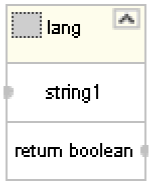
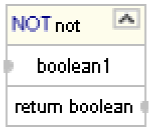
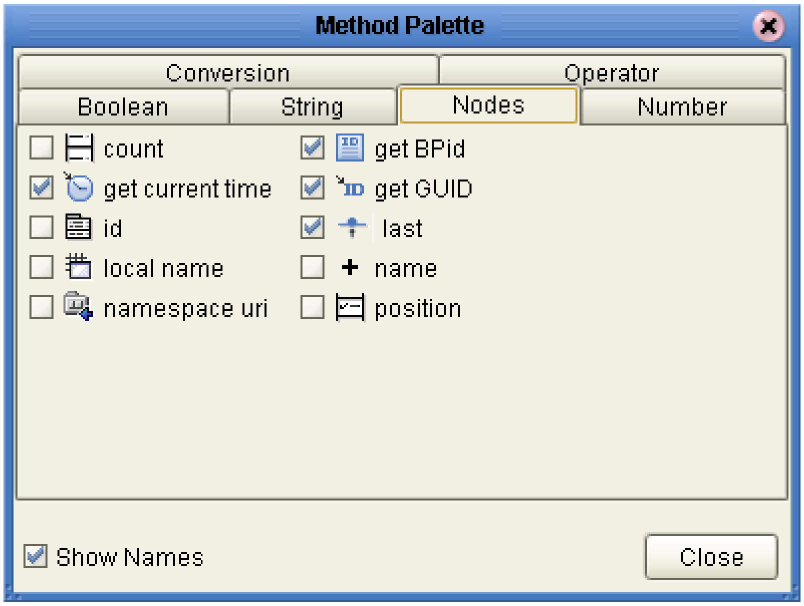
Symbol	Name	Function
	boolean	Converts the value argument to Boolean and returns true or false
	true	Returns true
	false	Returns false

Table 15 Boolean Methods (Continued)

Symbol	Name	Function
 lang string1 return boolean	lang	Returns true if the language argument matches the language of the xsl:lang element, otherwise it returns false
 NOT not boolean1 return boolean	not	Returns true if the condition argument is false, and false if the condition argument is true

A.5 Nodes

Figure 139 Method Palette: Nodes tab



Method Palette

Conversion | Operator

Boolean | String | Nodes | Number

☐ count
☑ get current time
☐ id
☐ local name
☐ namespace uri

☑ get BPid
☑ get GUID
☑ last
☐ name
☐ position

☑ Show Names

Close

Table 16 Nodes Methods

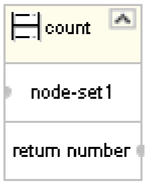
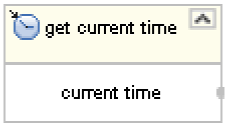
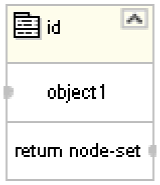
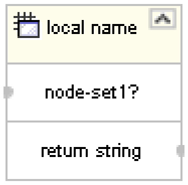
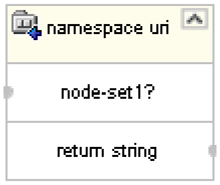
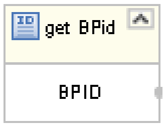
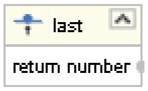
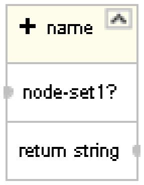
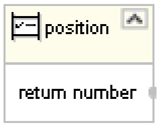
Symbol	Name	Function
	count	Returns the number of nodes in a node-set
	getCurrentTime	Gets the current time in ISO 8601 format (e.g. 2003-08-15T02:03:49.92Z).
	id	Selects elements by their unique ID
	local name	Returns the local part of a node. A node usually consists of a prefix, a colon, followed by the local name
	namespace uri	Returns the namespace URI of a specified node
	getBPId	Gets the business process instance ID.

Table 16   Nodes Methods (Continued)

Symbol	Name	Function
 get GUID GUID	getGUId	Gets a randomly generated globally unique ID.
 last return number	last	Returns the position number of the last node in the processed node list
 name node-set1? return string	name	Returns the name of a node
 position return number	position	Returns the position in the node list of the node that is currently being processed

Index

A

adding page flow elements **113**
archive files **106**
assembling the eVision components **112**

B

Back to Editor button **53**
blank canvas **67**
Bootstrap Command **130**
  Example **132**
bootstrap command **132**
border attributes **52**
Bottom Align tool **31**, **82**
box attributes **52**
Business Rule Designer **116**
Business Rule Designer Output pane **118**

C

canvas **62**
Center Align tool **31**, **82**
Check Box Group object **43**
check out the Project components **112**
class
  applying **58**
class attributes **58**
Class field **58**
class name
  selecting **58**
classes
  list of **58**
Command button **46**
command line arguments **130**
Command Prompt **132**
component palette **29**
Component Properties sheets **44**
configuring a Deployment Profile **128**
configuring Business Rule elements **116**
configuring the Decision logic **119**
connecting the page flow components **115**
context menu **65**
conventions
  path name separator **13**
  Windows **13**
Copy Component tool **31**
creating a Connectivity Map **125**
creating a Deployment Profile **127**
Creating a Group of Radio Buttons **49**
creating a Logical Host **127**
creating an Environment **126**
creating the page flow **113**
css folder **54**
custom style sheet
  applying **51**
Cut Component tool **31**

D

Decision Gate element **119**
Decision Gate Properties window **119**
Delete component **31**
design canvas **29**
document
  conventions **13**
downloading the sample project **107**
drag-and-drop techniques **29**
dynamic content **36**
dynamic tables **36**

E

Edit Options dialog box **48**
element class **57**
Enterprise Designer menu **82**
Enterprise Explorer tree **65**
Environment
  Bootstrap **130**
EQUAL icon **118**
eVision **12**
  default.css **51**
  described **12**
eVision Applications
  described **15**
eVision Studio
  overview **15**
  using **62**
eVision Studio archive file **107**
eVision Studio GUI
  described **29**
eVision Studio toolbar **30**, **82**
eVision Studio tools
  Bottom Align **31**
  Center Align **31**
  Copy Component **31**
  CSS **32**
  Cut Component **31**
  Delete **31**

- described **31**
- Horizontal Spacing **31, 82**
- Left Align **31**
- Paste Component **31**
- Preview **32**
- Right Align **31**
- Top Align **31**
- Vertical Spacing **32, 82**

eVision Style Editor **52**
eVision.css **51**
eVisionSampleComponents.zip **110**
eVisionSampleProject **110**
execution **19**

F

font decoration **52**
font type **52**
Form Object palette **32**
Form Objects **34, 43**
Form Objects palette **29, 75**
Form Objects title bar **34, 75**

G

graphic image
- adding to the Repository **39**

graphic images
- importing **41**

Graphical User Interface Components
- described **32**

graphical user interface components **19**
Group of objects
- moving **83**

GUI Component Selection Tables **33**
GUI Components
- placing **70**

GUIC
- property sheet **33**

H

Hidden object **43**
Home page
- designating **59**

horizontal element alignment **52**
Horizontal Line object **35**
horizontal lines
- resizing **84**

HTML Object palette **32**
HTML Objects title bar **34**
HTML Objects, **34**
HTML Text icon **74, 75**
HTML Text object **35**

I

Image icon **38, 71, 73**
image map
- creating and linking **38**

Image Map object **35**
Image object **36**
Importing a CSS **54, 59, 69**
importing the sample project **108**
installing eVision Studio **20**
Internet **19**
intranet **19**

J

Java classes **33**
JSP code **33**

L

Left Align tool **31, 82**
Link object **34**
linking to another page **38**
Logical Host
- Starting Manually on Windows **130**

M

making a Drop-Down list component dynamic **49**
manipulate **30**
Mapper **117**
mapping a String Literal **117**
mapping Page Flow attributes **124**
Method palette **117**
monitoring **19**
Multiple Entries in a Drop-Down List **48**

N

New Page Layout context menu **66**

O

object
- moving **83**

objects
- aligning **81**
- manipulating **30, 83**
- placing **30**
- resizing **82**

Operator tab **120**

P

page design
  previewing **29**
Page Flow **12**
Page Flow Components **107**
Page Flows **19**
  execution **19**
  model design **19**
  monitoring **19**
Page format
  controlling **52**
page format
  controlling **51**
Page Layout Designer **29**
  described **19**
Page Layout Designer canvas **30**
Page Layout Designer components
  detailed **33**
Page Link Wizard **59**
page links
  creating **58**
Page Properties sheet **44**
pagination images **36**
Password icon **78**
Password object **43**
Paste Component tool **31**
Pre-built components **29**
pre-built components **19**
pre-configured Web page components **107**
Preview CSS button **53**
Preview icon **49**, **82**
Preview Page tool **32**
Project icon **65**

R

Radio Button Group **49**
Radio Button object **43**
repeating node **36**
Repository
  saving to **82**
Repository icon **65**
Reset Button object **43**
return boolean operator **118**
Revert button **53**
Right Align tool **31**, **82**

S

sample application **106**
sample project .zip files **106**
sample project components **106**
Save CSS button **53**
Selection Box object **44**
Start the Logical Host **130**
String Literal container **117**
String Literal icon **117**
Style Editor **53**
style sheet
  applying **55**
  customizing **44**, **51**
  elements **52**
  importing **53**
Style Sheet Explorer **55**
Style Sheet Selector **56**
Submit Button icon **80**
Submit Button object **43**, **44**
Switch object **35**

T

Table **35**
Table object **35**
table pagination **36**
Text Area object **43**
Text Box icon **76**, **77**
Text Box object **44**
text-based property sheet **29**
text-bearing components **29**
Top Align tool **31**, **82**

U

Upload Button object **43**

V

Value entry dialog box **46**
vertical element alignment **52**

W

Web Connector icon **125**
Web interface objects **33**
Web-centric component **33**
Windows
  The Bootstrap command is case sensitive on Windows **132**
writing conventions **13**
WYSIWYG **19**