SeeBeyond ICAN Suite

eInsight Enterprise Service Bus User's Guide

Release 5.0.4

Version 20040604101213.

Contents

Chapter 3

Chapter 4

List of Figures

List of Tables

Chapter 1

Introduction

The *eInsight Enterprise Service Bus User's Guide* provides instructions and background information for all users of the SeeBeyond Technology Corporation™ (SeeBeyond™) eInsight Enterprise Service Bus™ (eInsight ESB) application. This chapter introduces you to this guide, its general purpose and scope, and its organization. It also provides sources of related documentation and information.

In this chapter

1.1 Overview

This guide explains how to use eInsight ESB. This user's guide includes information on the following topics:

- Working with modeling elements and the Business Process Designer.
- Creating and configuring business process models.
- Setting up the eInsight ESB database for Persistence and Monitoring.
- Creating and running the sample project.
- Using Task Assignment with the User Activity and Work List Manager.

1.2 Intended Audience

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

1.3 Contents of This Guide

This document includes the following information:

- **Chapter 1, "Introduction"** provides an overview of this document's purpose, contents, writing conventions, and supported documents.
- **Chapter 2, "Introduction to eInsight ESB and the SeeBeyond Integrated Composite Application Network Suite"** describes SeeBeyond's product suite and how it works with eInsight ESB.
- **Chapter 3, "Designing Business Process Models"** describes business process elements and procedures related to building a business process model.
- **Chapter 4, "Configuring Business Process Models"** discusses configurations and options.
- **Chapter 5, "Exception Handling"** explains the concept of exception handling and how to configure various methods of handling errors.
- **Chapter 6, "Deploy Business Process Models"** discusses deployment of the business process model.
- **Chapter 7, "Persistence and Monitoring"** describes these features and how to set up the required database instances.
- **Chapter 8, "Implementation"** gives a step-by-step example of a simple implementation.
- **Chapter 9, "Workflow Solutions With the User Activity"** gives a step-by-step example of a Task Assignment implementation.
- **Appendix A, "Method Palette"** describes the methods available from the Business Rules Designer.
- The **"Glossary"** defines eInsight ESB specific terms.
- **"Index"**

1.4 Writing Conventions

The following writing conventions are observed throughout this document.

Table 1 Writing Conventions

Text	Convention	Example
Button, file, icon, parameter, variable, method, menu, and object names.	**Bold** text	▪ Click **OK** to save and close. ▪ From the **File** menu, select **Exit**. ▪ Select the **logicalhost.exe** file. ▪ Enter the **timeout** value. ▪ Use the **getClassName()** method. ▪ Configure the **Inbound** File eWay.
Command line arguments and code samples	`Fixed` font. Variables are shown in ***`bold italic`***.	`bootstrap -p` ***`password`***
Hypertext links	**Blue** text	For more information, see **"Writing Conventions" on page 16**.

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.

1.5 Supporting Documents

The following SeeBeyond documents provide additional information about the SeeBeyond Integrated Composite Application Network Suite:

- *SeeBeyond Integrated Composite Application Network Suite Primer*
- *eInsight Enterprise Service Bus Installation Guide*
- *eGate Integrator for eInsight Enterprise Service Bus User's Guide*
- *eGate Integrator System Administrator Guide*
- *eGate Integrator Deployment Guide*

Chapter 2

Introduction to eInsight ESB and the SeeBeyond Integrated Composite Application Network Suite

This chapter provides an overview of the SeeBeyond Integrated Composite Application Network (ICAN) Suite and explains how eInsight ESB interacts with the other suite components.

2.1 The SeeBeyond ICAN Suite

SeeBeyond delivers the ICAN Suite, which allows companies to rapidly assemble enterprise scale, end-user applications built on existing systems and infrastructure. ICAN is an application-level network that unifies connectivity among people, application systems and devices in different locations and across organizations.

The notion of business services facilitates the implementation of extended applications. Service-oriented architectures clarify design and enable reuse by sharing logic and data among different client systems and users.

2.2 Summary of Features

eInsight ESB provides the following additional features:

- Streamlines basic integration services needed by small to medium size enterprises in a cost-effective way as eInsight ESB creates business logic that helps organizations integrate and orchestrate Web services as well as proprietary or legacy applications
- Allows business analysts to model, in a graphical drag and drop environment, the business processes that their department or even entire company performs.
- eInsight ESB is tightly integrated with the ICAN Suite and leverages over 80 packaged SeeBeyond eWay™ Intelligent Adapters that provide out-of-the-box connectivity to a variety of systems, applications, databases and legacy technologies.

- Provides graphical transformation capabilities to drag-and-drop and visualize the mapping of data transformations between the systems being integrated.
- Resource Management - eInsight ESB 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 security module fulfills security needs such as authentication and authorization access to eInsight ESB functions.
- Repository storage and access - The setup, component, and configuration information for the elements of a Project, including business process and related eInsight ESB components, are stored in the Repository.
- Deployment abilities - Deployment profiles contain the information necessary to activate a Project business processes and associated components. When a deployment profile is activated, active business processes are made available as Web Services and published to SeeBeyond's UDDI Registry.
- Monitoring - The Enterprise Manager provides web-based monitoring abilities to observe and correct business process activity.
- Connectivity Mapping - The Connectivity Map maintains the relationships between eInsight and other system components. The Connectivity Map specifies the topology of services that will be invoked, by:
  - Identifying the order and nature that services are invoked.
  - Depicting relationships between the components, including the publish/subscribe information for data routing.
  - Defining the partners fulfilling the services that are invoked.
- Version Control - This feature maintains a history of business process versions, through a check-in and check-out process.
- Impact Analysis - Impact Analysis allows you to view how changes to one component or business process will impact other components or business processes of a Project or all Projects in the Repository.
- Import and Export of Business Process Models - The ability to import and export business process models makes it possible to recreate the processes on other systems or to reuse processes that may be similar in other areas.

2.3 Overview of eInsight ESB

eInsight Enterprise Service Bus (ESB) is a streamlined, distributed integration platform that combines Web services support, transformation and content-based routing. eInsight ESB is a rapidly implemented version of eInsight Business Process Manager.

The Enterprise Service Bus edition provides a lower-cost alternative to deploy integrated business processes as composite applications built on a services-oriented architecture. eInsight Enterprise Service Bus is sold standalone and supports

SeeBeyond applications and eWay Intelligent Adapters, which can be purchased separately as plug-ins to eInsight ESB to access external applications and data.

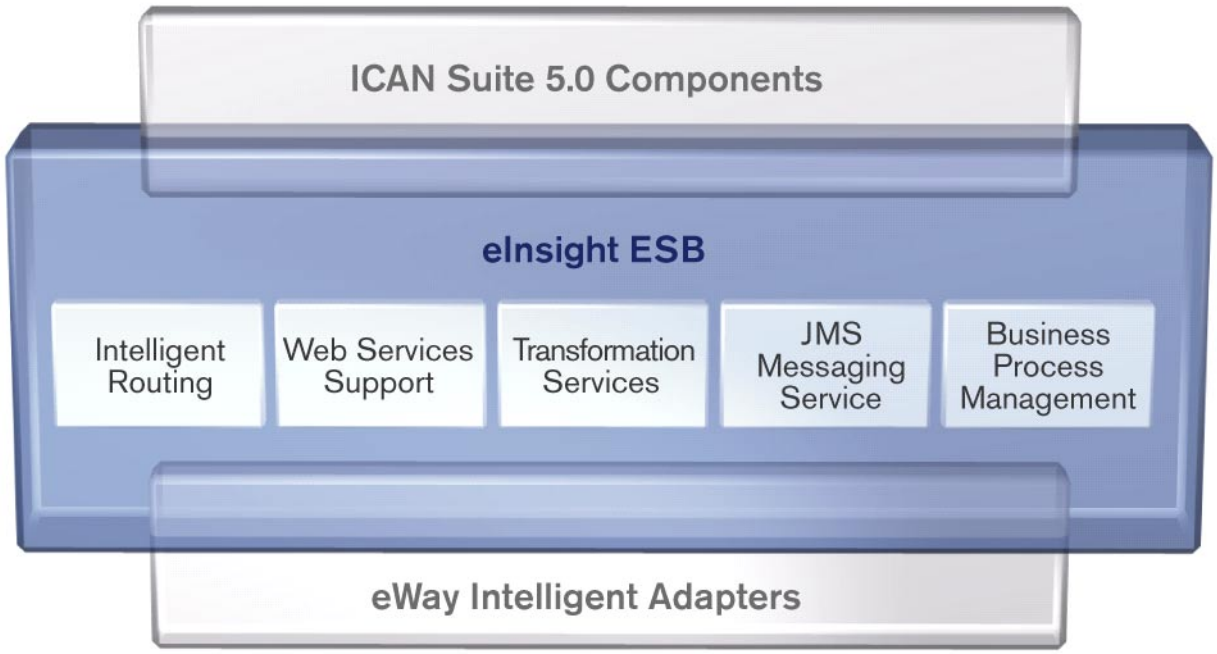
2.3.1 SeeBeyond ICAN Suite Integration

eInsight ESB is tightly integrated with the SeeBeyond® Integrated Composite Application Network™ Suite and is compatible with the other ICAN suite products (also, see), including:

- eVision Studio
- ePortal Composer
- eBAM Studio
- eTL Integrator
- eView Studio
- SeeBeyond eWay™ Intelligent Adapters

Thus, as your organization's integration needs grow, you can leverage and reuse the eInsight ESB integration components that you have already deployed.

Figure 1 eInsight ESB and the ICAN Suite



2.3.2 eInsight ESB Architecture

eInsight ESB speeds the design and deployment of business processes by providing an open process modeling environment using BPMN (Business Process Modeling

Notation) for the graphical notation of a business process and by automatically generating the BPEL4WS code needed to implement the business process across all of the participating Web services (e.g. for applications and business partners).

Business analysts drag and drop components into the process model and then specify the additional flow control and business rules that manage what services are called and when. eInsight ESB supports importing and exporting BPEL4WS to share processes with third party tools.

The technologies that carry out eInsight ESB business process functions are entirely based on industry standards. eInsight ESB uses the following standards:

- Business Process Modeling Notation (**BPMN**), from the Business Process Management Initiative (BPMI) standards body, provides a standard graphical view for Business Process Execution Language for Web Services (**BPEL4WS**).
- Web Services Business Process Execution Language (**BPEL4WS**) is the underlying code generated when creating a business process.
- Web Services Description Language (**WSDL**) is an XML-based language used to define Web services and describe how to access them.
- J2EE Connector Architecture (**JCA**) provides a mechanism to access external applications and data. The JCA engine is implemented as a standard JCA 1.5 module and it plugs into the SeeBeyond Integration Server.

2.3.3 Business Process Modeling and Design

You can use eInsight ESB to streamline operations by creating business logic that helps you reach outward to include customers and trading partners. Using eInsight ESB to implement business process management removes inefficiencies by orchestrating a unified work flow. This flow can include multiple systems/users, therefore extending to customers.

eInsight ESB allows you to model the business processes that your department or even your entire company performs on a regular basis. The tools provided allow for various scenarios and events that may take place in your process.

2.3.4 Business Process Designer

The Business Process Designer serves as the front-end design tool used to create a visualization of your business workflow and increase understanding of the business processes involved. As the business user, you are able to integrate logic into the business process. The Business Process Designer includes the following major areas:

- **Enterprise Explorer** - Displays a hierarchical representation of all the business process models and related components. This view shows you what is currently being displayed on the modeling canvas.
- **Business Rule Designer** - Allows you to graphically configure relationships between Input and Output Attributes.

2.3.5 Design Phase Overview

The basic steps that you will perform to design a business process model are as follows:

1. Plan and design a model that represents a business process taking place in your company.
2. Set up a Project and related components necessary to your business process model.
3. Create the new business process model in the eInsight Business Process Designer, using activities, links, decision and exception handling logic, and any other elements that express the actual business process.
4. Validate, generate and save the business process code to the Repository, where business processes configuration and deployment information is stored.
5. Create a Connectivity Map to configure the relationship between your components.
6. Select or create an Environment where your business process will run.
7. Select or create a Deployment Profile and activate it to complete the deployment process.

2.3.6 Runtime Phase Overview

Once all of the design phase tasks are complete and the system is running, you can monitor and manage business process activity and the overall Project with the Enterprise Manager. These tasks are only available if you use a database.

eInsight ESB Engine

The eInsight ESB Engine provides process coordination that enables the execution of business processes, activities and tasks. During the runtime phase, the eInsight ESB engine:

- Receives messages that instantiate business process instances.
- Writes monitoring, persistence and recoverability data to a database (if available).

See **on page 1** for a detailed view of the eInsight ESB architecture.

eInsight ESB Database

eInsight components connect to the optional database to provide:

- **Persistence** - The eInsight Engine writes instance data to a database to ensure that data is able to persist in the system.
- **Recoverability** - Using a database allows you to recover data from the last state of consistency.
- **Monitoring** - Instance data is written to a database and then read by the Enterprise Manager to provide current and historical system information.

2.3.7 Real-Time Business Process Monitoring and Alerting

Once the automated business processes are up and running, the Enterprise Manager provides a dashboard delivering instant visibility into the state of each business process activity. This allows organizations to monitor all aspects of both internal processes and those involving B2B trading communities. Processes are normally monitored for exceptions using business rules that alert managers as needed. This proactive alerting enables managers to respond instantly to business events such as increased customer demand, inventory shortages and quality problems.

Chapter 3

Designing Business Process Models

You can use eInsight ESB to configure the components depicted by each activity in your business process models. This chapter provides the background information you need to create and understand business process models.

3.1 Overview

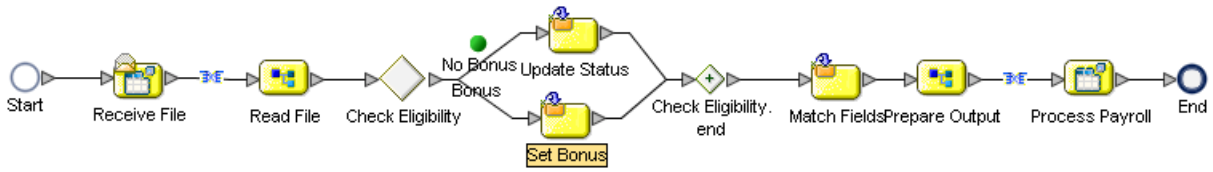
Topics in this chapter are:

3.2 Building a Business Process Model

A *business process* is a collection of actions that take place in your company, revolving around a specific business practice. These processes can involve a variety of participants and may include internal and external computer systems or employees. In eInsight ESB, you create a graphical representation of the business process called a *business process model*.

A business process modeled in eInsight ESB may look something like Figure 2.

Figure 2 Sample Business Process Model



Add a Business Process to your Project

Adding a business process to your Project provides an empty modeling canvas for you to work with and allows you to name the business process. Before you create a model of your business process, you must add a new business process to your project.

1. From the Enterprise Explorer pane, right-click on a **Project**.
2. Select **New** and **Business Process** from the context menu.
3. Enter a new name for your business process.

Note: *See the eGate Integrator for eInsight Enterprise Service Bus User's Guide for more information about creating a new Project.*

3.2.1 Modeling a Business Process

To model a business process, drag and drop modeling elements on the Business Process Designer, and then link these components to reflect the logical flow of the business process. eInsight ESB provides the tools you need to quickly develop business process models, including graphic editing tools to help you adjust, size, and align model components.

Business Process Designer

Once you create a new business process, you will build your model in the Business Process Designer (as shown in Figure 3). The Business Process Designer is the area in the Enterprise Designer where you view, create, and edit your business process models.

You can also drag and drop many other ICAN component operations from the Project Explorer directly onto the Business Process Designer or an activity. Some component operations that you can use in a business process include:

- File eWay: Read and Write operations
- Object Type Definition (OTD): Marshal and Unmarshal operations
- Java Collaboration Definition operations
- eVision Pages and PageFlows

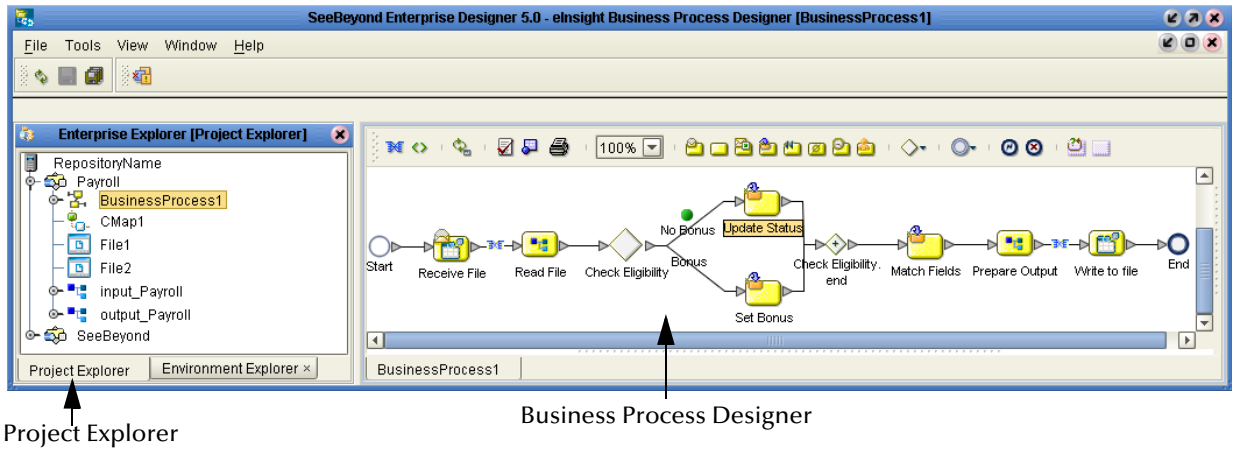
Create a Business Process Model

Begin designing your business process model by dragging and dropping modeling elements onto the Business Process Designer from the toolbar.

The **Start** and **End** Activity appear on the blank Business Process Designer, by default. There is only one starting point for any business process model. There can be multiple end points.

1. Drag the appropriate modeling elements to your blank business process to the Business Process Designer (Figure 3).

Figure 3 Business Process Model



2 Draw links between the modeling elements to show the process flow (Figure 3)

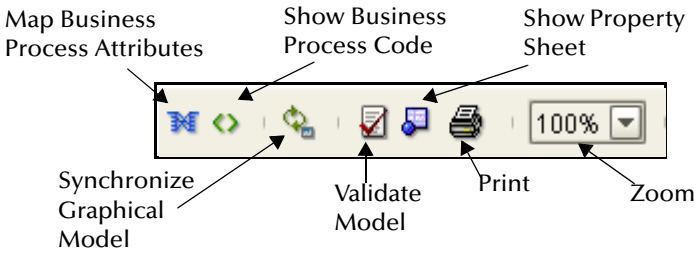
3 Select **Save** to save your changes to the SeeBeyond Repository.

This will validate your business process, generate the code to run it and save your changes to the SeeBeyond Repository.

3.3 Using the Business Process Designer

Using the Business Process Designer is very similar to any of the other ICAN Suite interfaces. When you create a new Business Process, you see the Business Process Designer and the a new Business Process Designer toolbar appears, as shown in Figure 4.

Figure 4 Business Process Designer Toolbar Options



- **Map Business Process Attributes** - Selecting this icon reveals the Business Rules Designer in the lower portion of the Business Process Designer.
- **Show Business Process Code** - You can toggle this icon to see and edit the corresponding Business Process Execution Language (BPEL) code.
- **Synchronize Graphical Model and Business Process Code** - This icon will update the business process code on demand. The business process code is also synchronized when the model is saved.

- **Validate Business Process Model** - Click this icon to check for any errors in your Business Process Model.
- **Show Property Sheet** - This icon shows the Property Sheet for the modeling element that is selected.
- **Print** - You can print the model from the toolbar. This options also allows you to control the scale of the printed model.
- **Zoom** - Controls the view size of the model and is available from the toolbar.

3.4 Modeling Elements

eInsight ESB provides a palette of modeling elements to assist you in customizing your business process model. The Business Process Designer is where the you create the business process flow. Like other objects, business processes appear in the Enterprise Explorer.

Elements from the Enterprise Explorer can either be dropped onto empty canvas or onto an Activity. Many elements provide custom settings so that you can model every detail of your process. Each business process model you create consists of some or all of the elements as described in the following sections:

Activity

There are several different kinds of activities you can include in a business process model. Table 2 shows examples of each of the different kinds of activities described below.

Add an activity

1. To add an activity to the modeling canvas do the following:
2. Click an activity from the eInsight ESB toolbar or the Project Explorer list, then drag and drop it where you want it.
3. Click on the activity name and begin typing to rename it from the default. The activity name must contain at least one character (A-Z, a-z, or 0-9), it must start with a letter or an underscore (_) and it may contain spaces.

   The selected activity appears on the modeling canvas.

Link modeling elements

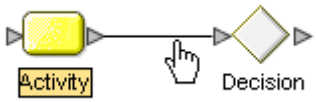
eInsight ESB supports orthogonal and diagonal link styles – this setting applies to all links in a model and is an automated application of the style.

To link modeling elements

1 Move your cursor over the connector portion of your modeling element.

2 Hold the cursor over the outside edge of the modeling element until it changes from the arrow pointer to a hand (see Figure 5).

Figure 5 Link Example



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

To change link style

1 Right-click the business process.

2 From the context menu, select Toggle Link Style.

Figure 6 Orthogonal Link Style

Table 2 Activity Elements

Button	Command	Function
	Start Node	The Start Node is a modeling element indicating the start of the process. This element appears in the Business Process Designer, by default, when you create a new business process model. A Start Node can only link to a **Receive Activity**.
	Link	Links indicate the flow of the business process by connecting activities together. eInsight ESB ensures the model is being properly linked because it does not allow invalid links to connect. Links can also accept business rules. A link with a business rule is marked with a blue icon.

Table 2 Activity Elements

Button	Command	Function
	End Node	The modeling element indicating the completed state of a business process. This element appears in the Business Process Designer, by default, when you create a new business process model.
	Receive Activity	This Activity type is used to indicate the invocation of a business process or to wait for the arrival of an inbound message. The Receive Activity represents the actual method by which a business process is initiated.
	Activity	An Activity is a step in the business process in which eInsight ESB invokes a Web Service or an ICAN component. Depending upon the configuration of the component, a response may or may not be required.
	Reply Activity	The Reply Activity allows a business process to respond to the external system or user that originally invoked the business process. 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 business process, the Reply uses information that correlates the message in the calling system. A Reply acts as the last step in a business process in which the business 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.
	Business Rule Activity	This Activity is used to map and manipulate data in the Business Rules Designer. You can also add a Business Rule to some links for the same purpose.
	Compensate	This 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 Activity	This activity allows data to pass through without any changes.
	Wait Activity	The Wait activity will delay the process for a set period of time.
	User Activity	This activity is used to represent and configure a step in a business process that requires human intervention.

Branching Activities

Branching Activities are objects you add to your business process models to specify the logical flow of information. eInsight ESB provides three different kinds of Branching Activities—Decisions, Event Based Decisions, and Flow.

Add a Branching activity

To add a Branching Activity to the modeling canvas:

1. Click on the Branching Activities toolbar icon and select the type of Branching Activity you would like to add.
2. Click on your choice and drag it from the menu to the Business Process Designer canvas.

   The selected Branching activity appears on the modeling canvas.

Table 3 Branching Activities

	Decision	A Decision allows one of several possible paths to execute, based on expression logic. This element is used to create complex expressions that determine the path of the business process. It also contains the expression and connection names. Decisions allow you to define expressions that are evaluated to determine the proper business process flow. Expressions are built using the Business Rules Designer interface and Business Process Attributes.
	Event Based Decision	Multiple inbound messages can be juxtaposed against one or more timeout conditions, to allow the type of message received to determine the appropriate business process path.
	Flow	Allows you to specify one or more business process paths to be performed concurrently.

Intermediate Events

Intermediate Events are those activities that can receive a Business Process. Some intermediate events handle exceptions that may occur during your business process or compensate for exceptions that occur.

Add an Intermediate event

To add an **Intermediate event** to the modeling canvas:

1. Click on the **Intermediate event** toolbar icon and select the type of **Intermediate event** you would like to add.
2. Click on your choice and drag it from the menu to the Business Process Designer canvas.

Table 4 Intermediate Events

	Timer Event	A Timer Event is set upon Activities, sets of Activities or a business process as a whole to ensure that process(es) complete within given amount of time. Timeout conditions also allow you to design the process that takes place after a timeout condition takes place. This modeling element is used with Event Based decisions only.

Table 4 Intermediate Events

	Message Event	This is similar to a Receive Activity, but it occurs only in the middle of a process. Each of these elements can be a different message. This modeling element is used with Event Based decisions only.
	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 business process. The components within that scope will throw the exceptions when things go wrong and the exception handler will automatically initiate the appropriate process to handle the problem.
	Catch All Exceptions	This exception handler is configured to handle un-named exceptions that occur in a scope or across a Business Process.
	Compensation Handler	Used when something in a business process fails and requires a rollback of upstream activities (like money has to be returned to the customer account). On an automatic basis in the business process, upstream steps in the business process 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	This element exists in case you want to create an error along a certain business process path.
	Terminate	This element allows you to terminate an entire business process, before it reaches an end node.

Scope

The behavior for one or more activities can be defined by a scope. A scope can provide exception handlers, event handlers and a compensation handler. The exception handlers for the scope can be used to catch the faults caused by the possible exception responses.

Table 5 Scope Element

	Scope	The Scope element allows you to apply exception handling to a set of sequential or simultaneous steps in a business process.

While

This modeling element makes it possible to have repeating or looping logic inside of a business process.

Table 6 While Element

	While	This allows you to create a looping process within a business process (for instance, a negotiation process may take several weeks, but the manager wants to review the daily status). The loop continues until the negotiation is complete, and then the business process continues.

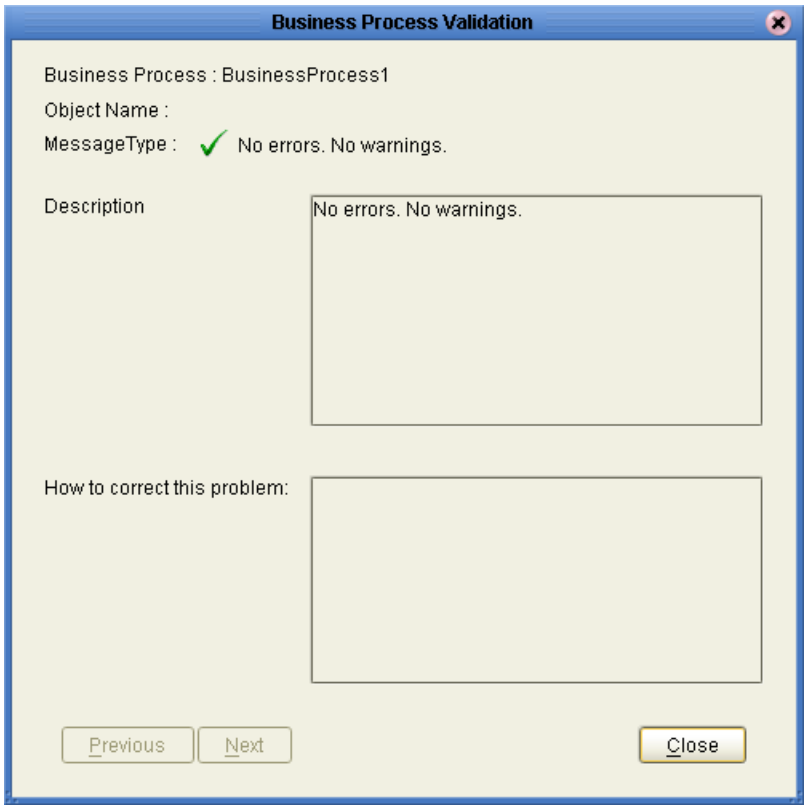
3.4.1 Validating a Business Process Model

After creating a business process model, you can check to see if there are any errors or warnings. Errors appear for activities that are not connected or an incorrect number of output links from an activity. Warnings appear when there is a problem, but it is not critical enough to stop the Business Process.

To check the business process for errors or warnings

- On the toolbar, select **Validate Business Process Model.**
  - If an error or warning is encountered, a message box displays more information about the error or warning.
  - If there are multiple errors or warnings, an option to view the **Next** error displays for each additional error or warning.
  - If there are no errors or warnings, a message appears stating so, as shown in **Figure 7.**

Figure 7 Validate Business Process Model



3.4.2 Saving a Business Process Model

Even if a business process model is not complete and/or contains errors, you can save it as a "work in progress" and return to it later.

To save a business process model

1 Do one of the following:

- Pull down the **File** menu and select **Save**
- Press **Ctrl+s** on the Keyboard
- On the toolbar, click **Save**.

Chapter 4

Configuring Business Process Models

This chapter provides the background information you need to configure business process models.

4.1 Overview

Most of the advanced modeling elements and some of the basic modeling elements allow you to configure settings that customize your business processes. Topics in this chapter are:

- **Configuring Modeling Elements** on page 33
- **Business Process Properties** on page 36

4.2 Configuring Modeling Elements

Some modeling elements have configuration options. This section describes those elements and how to configure each option.

4.2.1 Business Rule Activity

The Business Rule Activity is used to map and manipulate data in the Business Rule Designer. You can also add a Business Rule to some links for the same purpose.

Business Rule Links

You can configure logic in a Business Rule Activity or add a Business Rule to a link.

Add a Business Rule Activity

1. Select the Business Rule Activity from the Business Process toolbar.
2. Drag the Business Rule Activity to the Business Process Designer.
3. Click the **Map Business Process Attributes** icon on the toolbar.

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

Add a Business Rule to a Link

1. Right-click on a link that you have created.
2. Select **Add Business Rule**.
3. Click the **Map Business Process Attributes** icon on the toolbar.

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

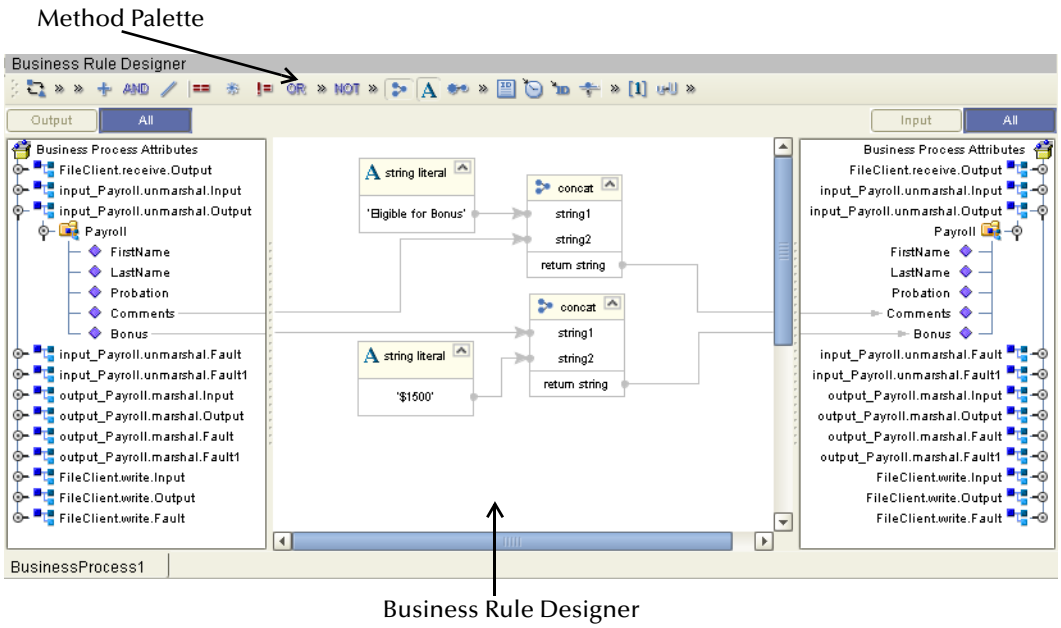
Business Rule Designer

The Business Rule Designer allows you to configure relationships between Input and Output Attributes. Some attributes are automatically created for each activity when you drag and drop a component on the Business Process Designer (as shown in Figure 8).

The Business Rules Designer appears when you click the **Map Business Process Attributes** icon (as shown in Figure 8). It is active when you:

- Add or select a link with a Business Rule.
- Add or select a Business Rule activity.

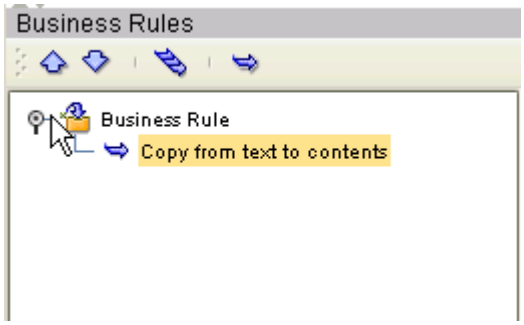
Figure 8 Business Rule Designer



Business Rule Editor

The Business Rule Editor allows you to have an advanced view of the business rules for your business process (See Figure 9).

Figure 9 Business Rule Editor

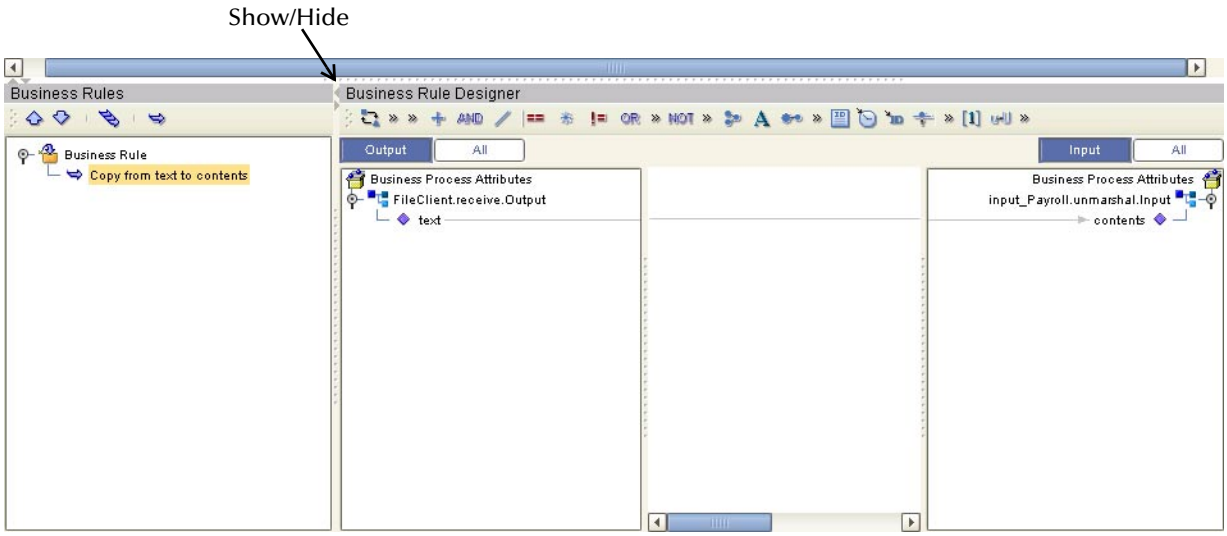


To Show/Hide the Business Rules Editor

From the Business Rule Designer view, you can access the Business Rule Editor.

1 Click the top gray triangle to the left of the Business Rules Designer title bar to show the Business Rules Editor (see Figure 10).

Figure 10 Show/Hide Business Rules Editor

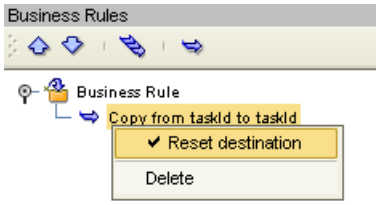


2 Click the top gray triangle to hide the Business Rule Editor.

To Remove a Rule

1 Right-click a rule, as shown in Figure 11.

Figure 11 Delete Rule



2 Select **Delete** to remove the rule.

Use of Reset Destination

Reset Destination allows eInsight ESB to reset a Business Process Attribute to an empty state before performing a mapping. Since this can create performance overhead, the default setting is off.

When a Business Process Attribute with repeating nodes is continually re-used (as occurs with a Business Process that uses a Loop Activity), it may be necessary to reset the Attribute value to an empty state.

When eInsight ESB populates these nodes during the looping process, it overwrites the data in the Business Process Attribute. If the Attribute contains more information than will be overwritten, there is leftover data in the node. In this case, extraneous information appears that does not reflect the current intended value of the Attribute. In addition, it is important to activate the Reset Destination option on the first rule, to ensure that the first action in mapping process resets the Attribute to an empty state.

Activate Reset Destination

The Reset Destination feature can be set in the Busines Rules Editor.

1 From the Business Rule Editor, select the first rule.

2 Right-click the first Rule and select Reset Destination.

The option now appears with a check to indicate that Reset Destination is activated for the selected rule.

The Reset Destination option is also used when creating a business process that includes a User Activity inside of a While Loop. The purpose of the Reset Destination option is to create an output container. See **"Configure a User Activity inside a While Loop" on page 101** for specific instruction on how to set this option.

Method Palette

Use the Method Palette in the Business Rule Designer (as shown in Figure 8) to configure data passed between input and output nodes. You can drag and drop a method from the method palette to the Business Rules Designer and then configure the method.

See **"Method Palette" on page 133** for more information about each method available in the Method Palette.

4.3 Business Process Properties

Each Business process has a set of properties that you can change and create. These properties provide rapid creation and deletion of business process attributes. eInsight ESB uses this information to automatically create the appropriate business process attributes and input/output structures, for use in the Business Rule Designer.

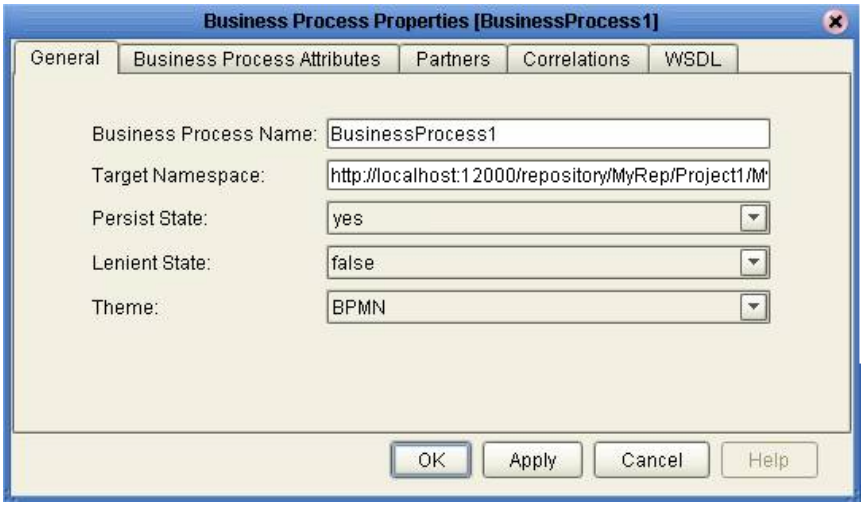
Edit Business Process Properties

1 Right-click on the **Business Process** you want to edit.

2 Select **Open Business Process Properties Sheet**.

The **Business Process Properties** dialog appears as shown in Figure 12.

Figure 12 Business Process Properties: General Tab



4.3.1 General

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

Edit General Business Process Properties

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

- **Business Process Name -** Change the default name.
- **Target Namespace** - address of the business process.
- **Persist State** - See **"Configure Persistence for the Business Process" on page 67** for more information.
- **Lenient State** - The Lenient State property specifically applies to projects that are imported from ICAN 5.0.0, to ICAN 5.0.4, or business processes from other third-party vendors. These projects do not contain the updated optional node assignments and will throw an exception which kills the process instance. The values are:
  - **true:** Adds the attribute *sbynruntime:processLenient="true"* to the BPEL Process tag. This in turn causes any copy/write activity, that would throw an exception, to be skipped. A "false" is returned as an evaluation of the condition that has thrown a fault, overriding the user settings that may have been set for the switch block (by the user through the decision gate mapper).
  - **false:** No attribute is added.

The default property is false. If the Lenient State is not set, the Lenient flag on the individual copy statement, if present, will have the same effect.

- **Theme -** The default Theme is BPMN. Select Custom 1 for a different look.

2 Click **OK** to Save your changes and exit the **Business Process Properties** dialog box.

4.3.2 Business Process Attributes

Business Process Attributes are data values used by a business process. They make it possible to share data between activities in a business process as well as move data to and from the components that implement those activities. Complex structures such as OTDs and Collaborations are represented automatically in the Enterprise Explorer and are available for use in your business process.

Some examples of Business Process Attributes are:

- customer names
- addresses
- order quantities
- item descriptions

Business Process Attributes are used to pass values between the business process and external sources. Business Process 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 business process and is included in the business process output.

eInsight ESB 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 business process attributes.

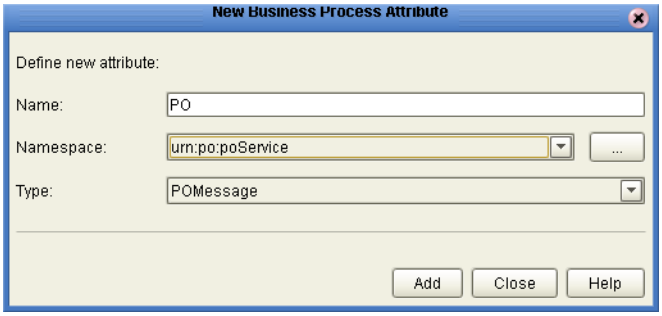
Create New Business Process Attribute

1 Select the **Business Process Attributes** tab (see Figure 14).

2 Select **New** to add a **New Business Process Attribute**.

The **New Business Process Attribute** dialog box appears as in Figure 13.

Figure 13 New Business Process Attribute



3 From this dialog, complete the following information:

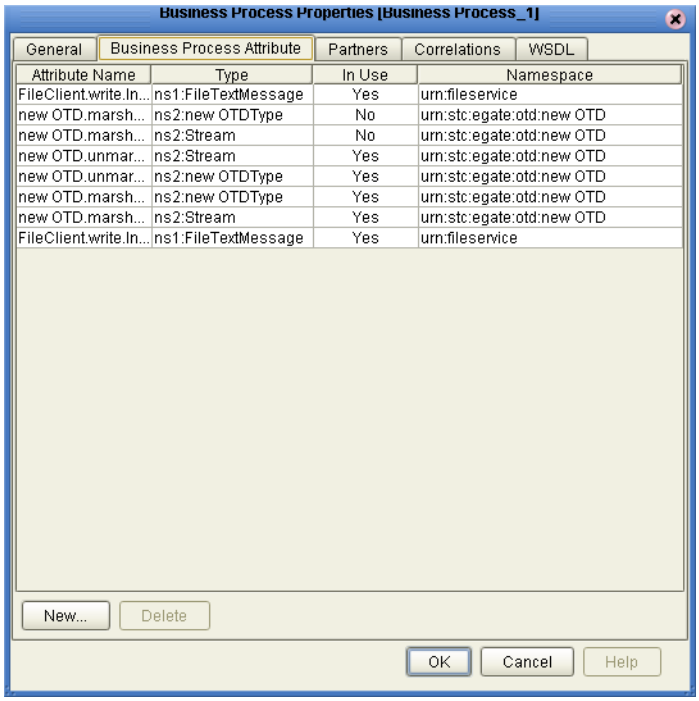
- Enter a **Name** for the attribute.
- Select or browse for an existing **Namespace.**
- Select an available **Type** for your attribute.

4 Click **Add** to Save the attribute or Close to return to the **Business Process Properties** dialog box.

Edit a Business Process Attribute

1 Select the **Business Process Attributes** tab (see Figure 14).

Figure 14 Business Process Properties: Business Process Attributes Tab



2 Select an existing attribute and:

- **Rename**: Select and double click the attribute name to rename it.

Note: *Some attributes cannot be renamed.*

- **Delete**: Select Delete to remove the attribute.

3 Click **OK** to Save your changes and exit the **Business Process Properties** dialog box.

4.3.3 Partners

The Partner is an abstracted identification for an external system that will appear in the Binding dialog within the Connectivity Map Editor. Multiple activities can use the same external system – hence, multiple Activities may have the same Partner. By

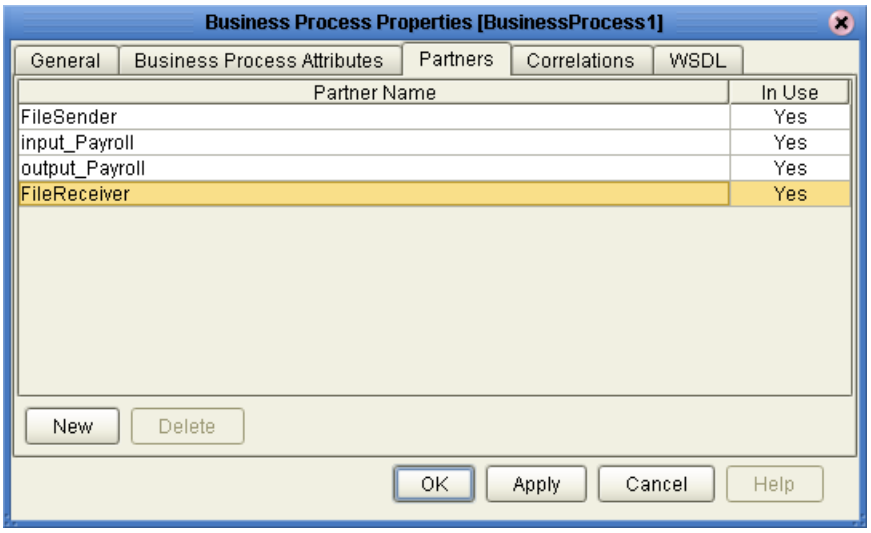
default, eInsight ESB assigns this identification to speed up and automate the model development.

When creating a Business Process that will be used as a sub-process, you need to create a partner and associate it with the receive or receive/reply pair. See **"Sub-Processes" on page 48** for more information.

Create New Partner

1 Select the **Partner** tab (see Figure 15).

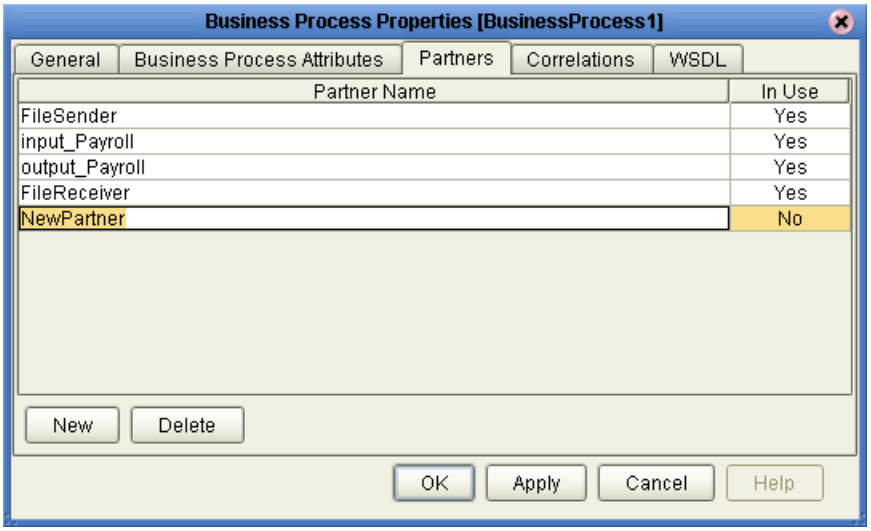
Figure 15 Business Properties: Partner Tab



2 Select **New** to add a **New Partner**.

The **New Partner** is added to the Partner list, as shown in Figure 16.

Figure 16 New Partner



3 Click the Partner name to rename the Partner.

Delete a Partner

You can only delete a Partner that is not in use.

1 Select the **Partner** tab (see Figure 15).

2 Select the Partner name that you want to remove.

3 Select **Delete** to remove the Partner.

The **New Partner** is removed from the Partner list.

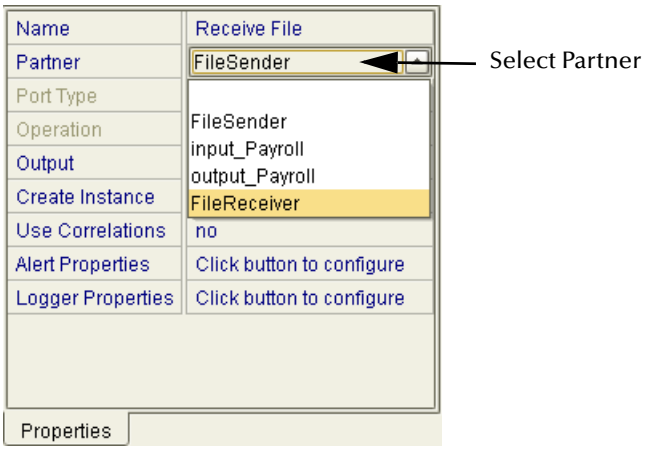
Select a Partner for an Activity

1 Select an activity from the Business Process.

2 Click the Show Property Sheet icon from the eInsight ESB toolbar.

The activity's property sheet appears, as shown in Figure 17.

Figure 17 Activity Properties



3 Click the Partner field and select or change the default Partner from the drop-down list.

4.3.4 Message Correlation

In some situations, you want to impose conditions on a set of messages, process a group of messages together or make a decision contingent on the receipt or non-receipt of all messages of a certain type. These situations can be handled using Message Correlation.

You can use Message Correlation when you want to sort messages into separate groups and execute your business rules on a group-by-group basis rather than an message-by-message basis. Messages in the same group are linked.

Groups are stored in memory, to facilitate processing each one as a unit. eInsight ESB refers to these groups as containers. As eInsight ESB retrieves a message, it correlates the received message against a business process instance. If found, eInsight ESB stores the message in the container for that business process. Otherwise, it will instantiate a new instance of that business process.

Message Correlation Example

In this example, the business process, expects to receive three course grades. The courses are Math, English and Computer Science. Each message contains the course grade, the course type, and a Correlation ID to indicate where this message belongs.

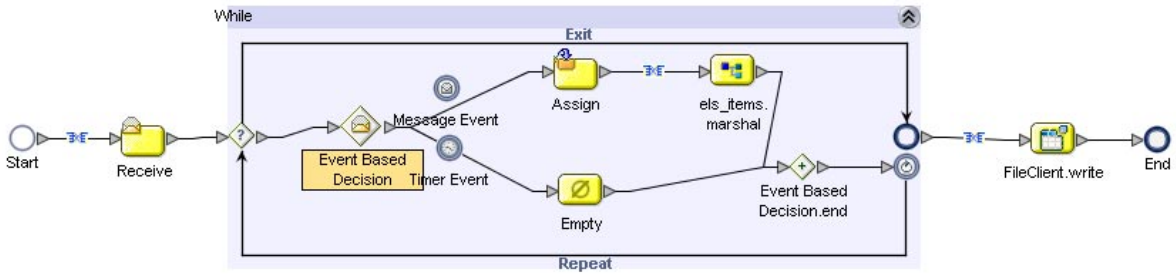
A new message arrives with a Correlation ID of 101. The first thing eInsight ESB does is correlate that message to see if there is a match on the newly arrived message. Since this is the very first message, there is no match and a new instance is created. The second message has a Correlation ID equal to 101 and is forwarded to the same business instance as above. The third message has a Correlation ID of 102. Thus it is forwarded to a new business instance and so on.

This process can continue based on conditions set by the user. This is based on Count or time expiration. A sample business process is shown below for Message Correlation. The first business process uses a File eWay to read a DTD based message, unmarshal it and then invoke the second business process passing in the unmarshaled message.

Figure 18 Message Correlation: First Business Process

The second business process receives the unmarshaled message using Event Based Decision and Timer Events. The Event Based Decision and Timer Events are in a While loop. The While continues to loop until either a count has been reached or time has expired. When a messages are received, they are stored in containers.

Figure 19 Message Correlation: Second Business Process



The Timer Event is used to set the expiration time. If time expires, then the loop condition is set to false to terminate the loop. If a message is received, then message counter is incremented and if the maximum number of messages have been received, then loop is terminated. At the end, the date is written to a file.

4.3.5 Correlation Keys and Sets

Configuring Correlation consists of the following steps:

- **"Create a Correlation Key" on page 43**

  A *correlation key* is a value that you can assign to a business process, like a Purchase Order number. The correlation key provides a way to associate and route information about specific business process 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 business process.

- **"Add Correlation Sets" on page 44**

  Correlation sets are groups of properties shared by all messages in the group. A correlation set matches messages and conversations with a business process instance. For example, you may wish to assign a Purchase Order number and an invoice number to a transaction, so that all information about the purchase and payment are associated.

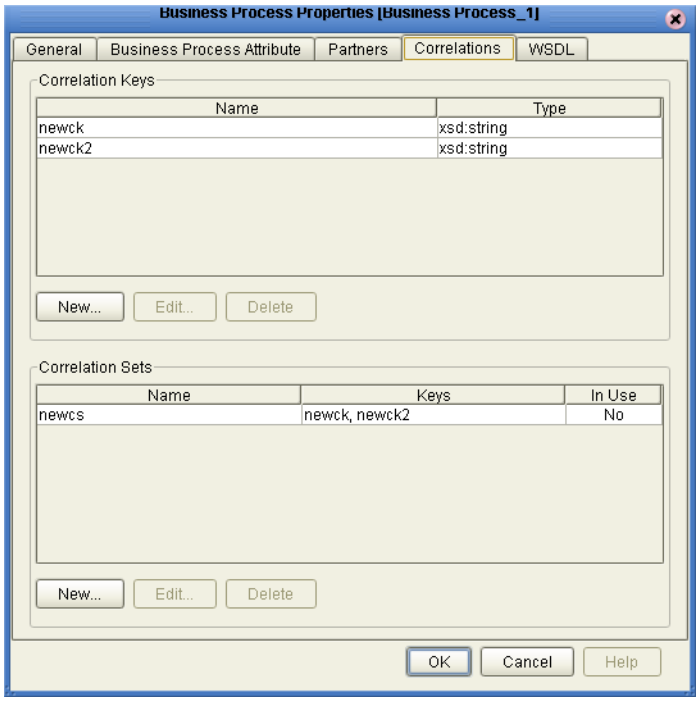
- **"Bind Sets of Correlations to Activities" on page 45**

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

Create a Correlation Key

1 Select the **Correlations** tab (see Figure 20).

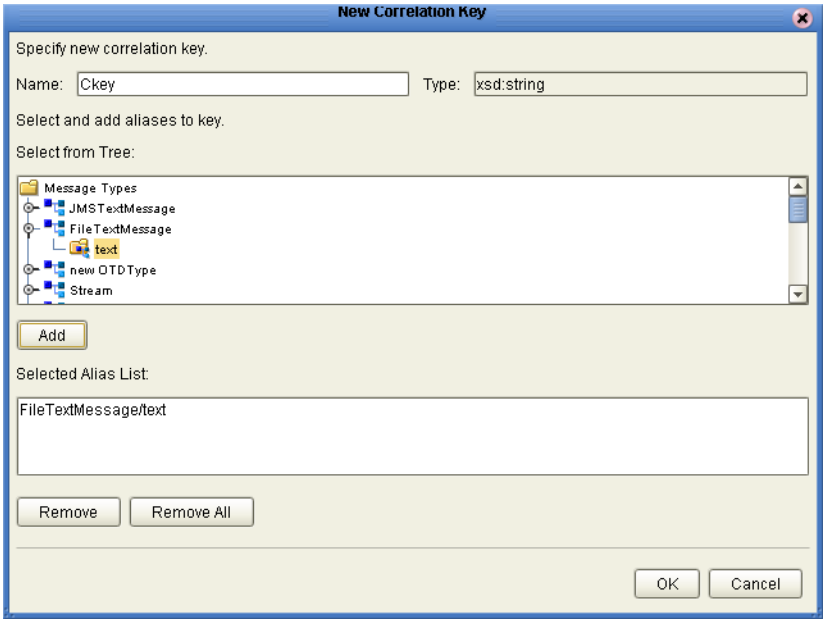
Figure 20 Business Process Properties: Correlations Tab



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

The **New Correlation Key** dialog appears as shown in Figure 21.

Figure 21 New Correlation Key Dialog Box



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

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

B 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 business process.

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

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

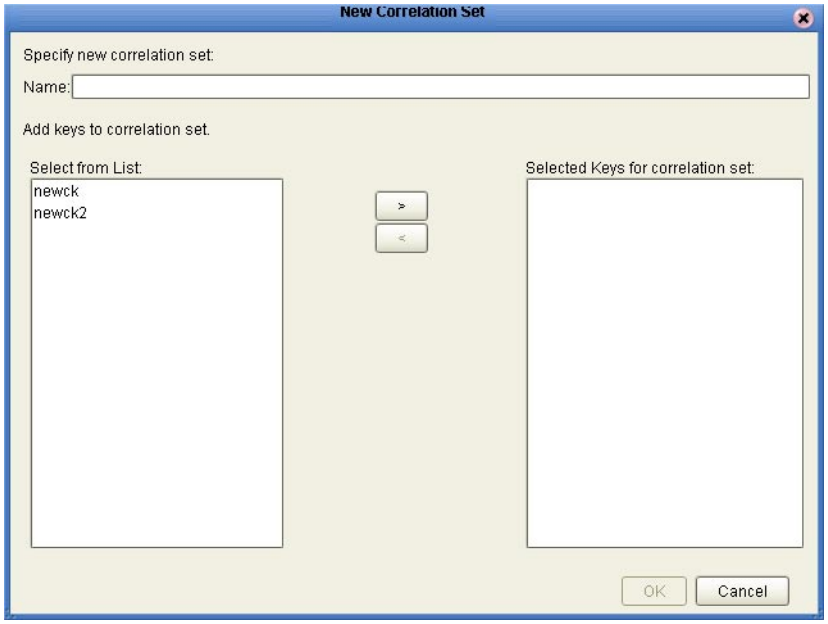
Add Correlation Sets

1 Select the **Correlations** tab (see Figure 20).

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

The **New Correlation Set** dialog appears as shown in Figure 22.

Figure 22 New Correlation Set Dialog Box



3 From the New Correlation Set dialog box:

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

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

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

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

Bind Sets of Correlations to Activities

1 Select an Activity.

2 Select **Show Property Sheet** from the toolbar.

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

4.3.6 WSDL Files

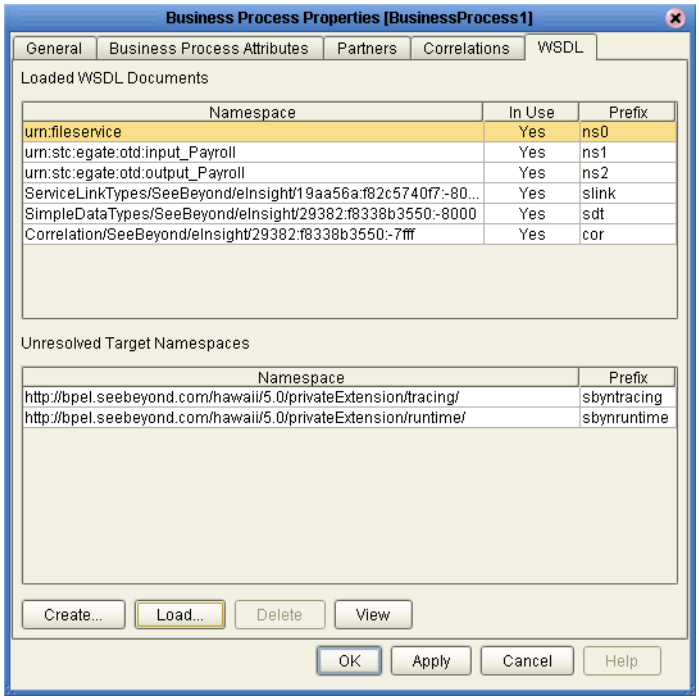
WSDL files are used to invoke and operate Web services. WSDL files can be used for web services on the Internet and/or to access and invoke remote applications and databases.

The WSDL tab is available from the Business Process Properties dialog. From this dialog, you can upload a WSDL file that represents predefined Business Process Attributes for use in your business process.

Upload a WSDL File

1 Select the **WSDL** tab (see Figure 23).

Figure 23 Business Process Properties: WSDL Tab



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

The **Load WSDL** dialog box appears, as shown in Figure 24.

Figure 24 Load WSDL dialog box

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

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

B Enter the path to the WSDL file.

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

4 Click **UPLOAD**.

Note: *eInsight ESB does not create the full WSDL Input message if the required inputs/ leaf nodes are not mapped in the Business Rule Designer. The part of the message which has not been mapped is not generated. This can cause errors at runtime.*

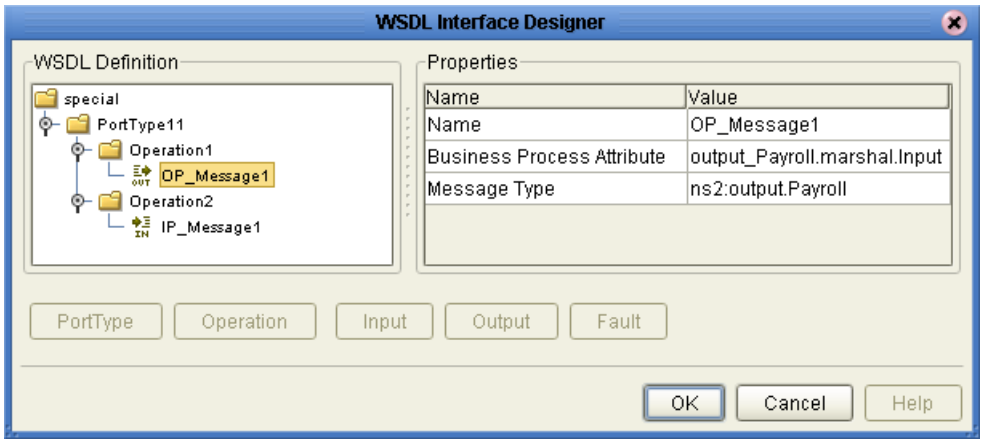
WSDL Interface Designer and Viewer

Create a New WSDL File

1 Click **Create** to create a new WSDL file.

2 The **WSDL Interface Designer** appears, as shown in Figure 25.

Figure 25 WSDL Interface Designer



3 Specify the following:

- PortType

- Operation
- Input and/or Output and/or Fault

4 Click on each WSDL definition to select the Business Process Attribute and Message Type that you will use.

Note: *If you accept the defaults for WSDL creation, the file is unique.*

Edit a WSDL file

1 Select the WSDL tab from the Business Process Property sheet.

2 Select a WSDL file from the list and click Edit.

3 The WSDL Interface Designer appears, as shown in Figure 25.

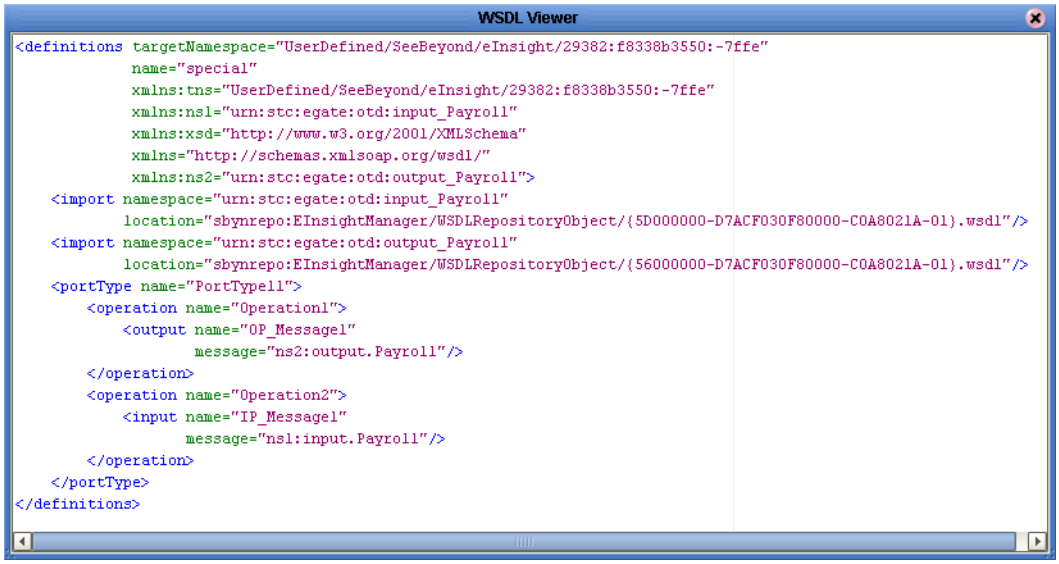
View a WSDL file

1 Select the WSDL tab from the Business Process Property sheet.

2 Select a WSDL file from the list and click **View**.

3 The WSDL Viewer appears, as shown in Figure 26.

Figure 26 WSDL Viewer



4 From the WSDL Viewer you can copy all or part of the WSDL code to the clipboard, and paste it to a text file.

You cannot edit from the WSDL Viewer.

4.3.7 Sub-Processes

Sub-processes are deployed business processes that can be invoked within the same Integration Server. When a sub-process is dropped into a business process, the Business Rules Designer is used to configure the input and output attributes for the sub-process.

Sub-processes are necessary when using the User Activity. See **Create the Sub-Process** on page 116 for a detailed example of a sub-process.

Sub-Process Overview

- Create a Business Process (this is the parent process). Create an Empty Activity as a placeholder for the sub-process.
- Create a second Business Process (this is the sub-process). Create an empty Receive Activity at the beginning of the Business Process and Reply Activity at the end.
- Create or upload a WSDL file that represents the inputs and outputs of the sub-process.
- Create a Partner from the Business Process Property Sheet: Partner tab.
- In the sub-process, open the properties of the initial Receive and select the appropriate Partner, portType, and operation. Do the same for the Reply.

Note: *If the 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.*

- Generate the Business Process Code and save the sub-process.
- Return to the parent process from the Project Explorer, drag the first operation under the sub-process to the sub-process placeholder in the parent business process.

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

- Perform any input/output mappings using the Business Rule Designer, in the parent business process.

Note: See **Create the Sub-Process** *on page 116* for a detailed example of a sub-process.

4.3.8 Repeating Nodes

For Web Services/components that contain repeating nodes, the Business Rule Designer displays repeating nodes within the input/output Attributes for each Activity. Repeating nodes contain the repeating icon. For direct node mapping, repeating nodes are used to dynamically populate 1-n values based on the runtime data.

Mapping Elements within Repeating Nodes

1. From the Method Palette, in the Business Rule Designer, select the **copy-from** method and drop it into the operator canvas.

   This action opens the **copy-from** entry box.
2. Enter the XPath expression and link it to the input of the appropriate activity.

The link from the appropriate output is implied within the XPath expression.

Using Predicates With Repeating Node Values

The XPath predicate functionality allows you to isolate particular elements within repeating nodes at runtime. The predicate functionality allows you to design conditional mappings within a business rule when using Business Process Attributes that contain repeating values.

When assigning values in the Business Rule Designer, you can access the *predicate* feature by selecting repeating nodes or elements of a Business Process Attribute and selecting "New Predicate" from the right-click menu. You use the Predicate Editor to create the conditions. From the Business Rules Designer, you can then map the associated repeating node values (at that point, the condition is in effect for that mapping).

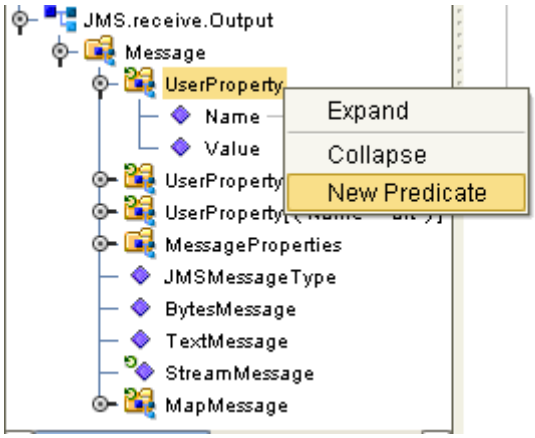
The existence of the condition will appear to the right of the repeating node or element for which the condition has been developed. At runtime, the design condition is used to select the correct element and performs the mapping, as designed.

To create a new predicate

1 From the Business Rule Designer, right-click on a repeating node.

2 Select New Predicate, as in Figure 27.

Figure 27 New Predicate

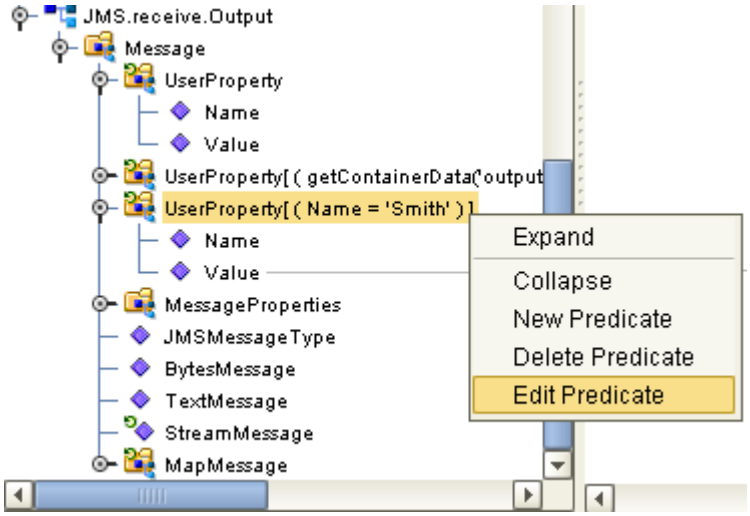


The Predicate Editor opens.

To edit a predicate

1 From the Business Rule Designer, right-click the existing predicate, as shown in Figure 28.

Figure 28 Edit Predicate



2 Select Edit Predicate from the menu.

The Predicate Editor opens.

To delete a predicate

1 From the Business Rule Designer, right-click the existing predicate.

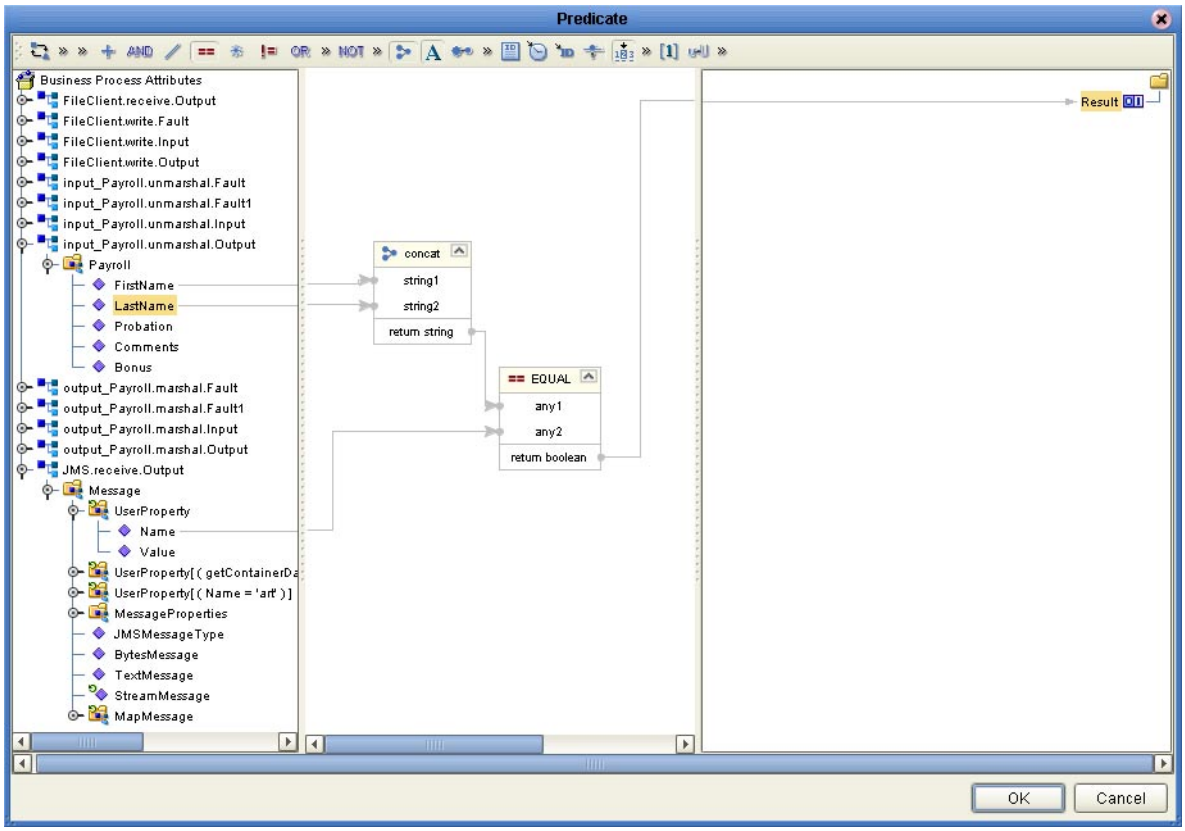
2 Select Delete Predicate from the menu.

The Predicate condition is removed.

Predicate Example

The most common use of the predicate functionality will be to create a condition using either runtime Business Process Attribute values or fixed values in an expression, and then create an appropriate mapping for when that condition is found to be true.

Figure 29 Predicate Editor

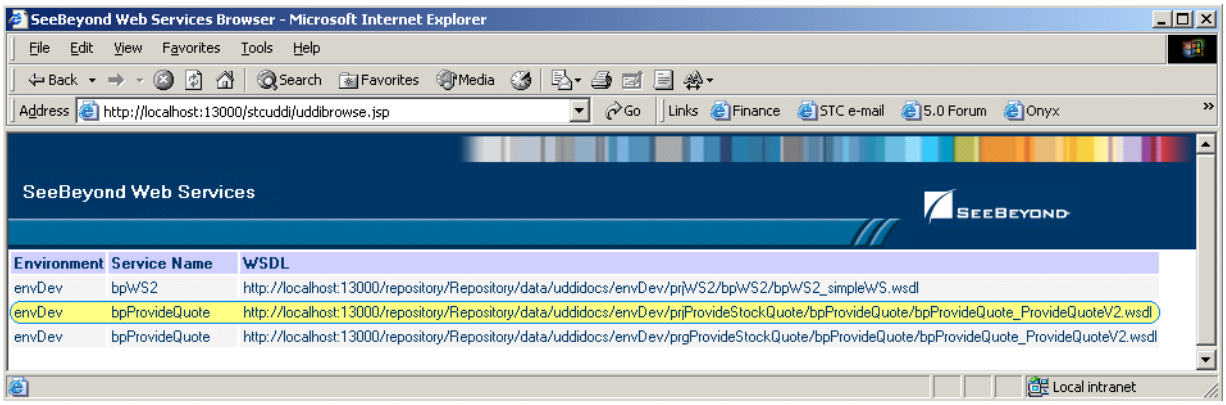


For instance, Figure 29 illustrates a condition in which the names in one Business Process Attribute must match the “name” in the JMS Message User Property. When the condition is found to be true, the appropriate mapping for “value” will then take place. In addition, only the appropriate value for “value” will be mapped from the series of name/value pairs.

4.3.9 Business Processes as Web Services

A Business Process can be exposed as a Web Service. Business Processes can also invoke other external web services. The Web Services created are available to be invoked by any Web Service client. Upon activation, the WSDL for the business process is placed in the ICAN Repository and accessible from a UDDI browser, as shown in **Browse Web Services** on page 52.

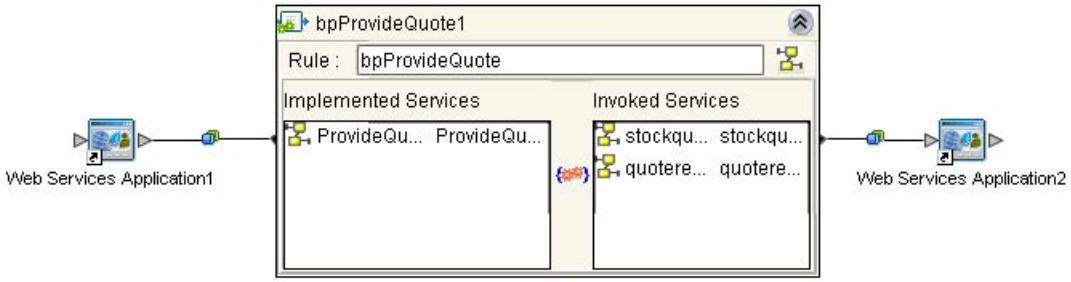
Figure 30 Browse Web Services



Web Service Application

Use the Web Service Application icons in the Connectivity Map to map a business process to a web service, as shown in Figure 31.

Figure 31 Web Service Application



Chapter 5

Exception Handling

This chapter explains the concept of exception handling and how to configure various methods of handling errors.

5.1 Overview

eInsight ESB ensures the integrity of critical business transactions and long-lived processes as they flow between multiple applications and multiple enterprises via automated exception handling capabilities. Exceptions can be automatically handled so that every process either completes or is successfully backed out using compensating transactions to ensure consistent information flows between systems.

5.1.1 Scope and Process Level Exceptions

In eInsight ESB, Exception Handling allows one or more components to throw an exception that is caught by eInsight ESB within a Scope or at the process-level. Scope allows you to define a range for handling exceptions. The range of the Scope can span one or more Activities in the business process. When your exceptions handler is not attached to a Scope, the Exception Handling is at the process level.

You can configure eInsight ESB to catch all exceptions or certain exceptions that you specify. The elements that you use to configure Exception Handling in your model are:

- Catch Named Exceptions
- Catch All Exceptions

Exception Handling Configuration

Exception handlers are configured to catch errors that are thrown by components and/or Web Services. These systems can be configured to publish one or more exceptions.

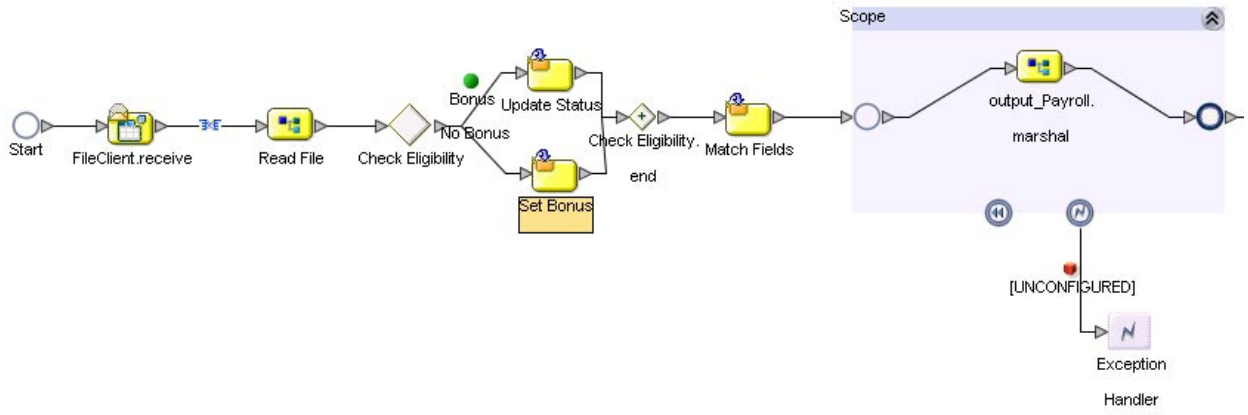
Catch Named Exception

To catch a named exception, there must be a defined fault in the WSDL file for your business process. You can use defined faults or create a WSDL file that includes faults with the WSDL Interface Designer (see **WSDL Interface Designer and Viewer** on page 47).

1 Drag a Scope element to the Business Process Designer.

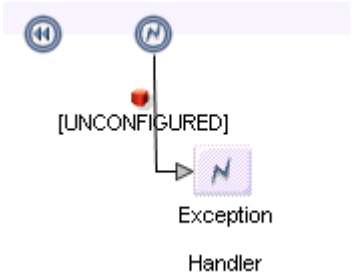
2 Drag one or more activities into the Scope and connect the Scope to the rest of your business process, as shown in Figure 32.

Figure 32 Build an Exception Handler



3 Drag the **Catch Named Exception** activity onto the Exception icon of the **Scope** for which the Exception Handler applies. See Figure 33.

Figure 33 Named Exception Handler



4 Select the Exception Handler activity and then click the **Show/Hide Property Sheet** from the Business Process Designer toolbar.

The property sheet appears on the right of your screen, as shown in Figure 35.

Figure 34 Named Exception Properties

Exception Name	ns2:MarshalException
Output	output_Payroll.marshal.Fault1
Alert Properties	Click button to configure
Logger Properties	Click button to configure

5 In the Exception Handler properties, double-click the empty fields to reveal a drop-down list and configure the following:

- The **Exception Name** – which is the runtime value for the exception that will be passed from the component to eInsight ESB at runtime.

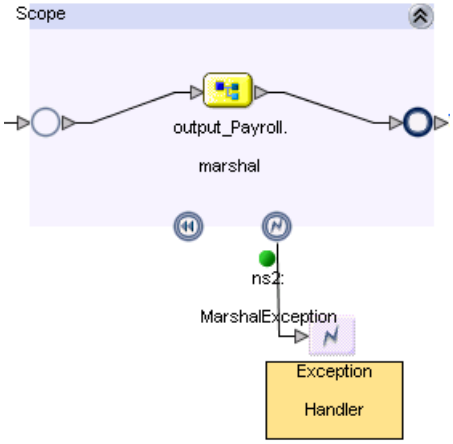
- The **Output** – which is the output Attribute that contains the runtime name of the thrown fault.

The fault name is auto-populated with values based on the components (and the associated WSDL files) in the Business Process Designer. If the drop-down lists are unpopulated, then there is no WSDL in use with defined exceptions. In this case, you must:

- Load a WSDL file with defined faults. See **"WSDL Files" on page 45** for more information.
- Create a WSDL file with defined faults. See **"WSDL Interface Designer and Viewer" on page 47** for details.

6 Close the Property Sheet by clicking the **Show/Hide Property Sheet** on the Business Process Designer toolbar.

Figure 35 Configured Exception



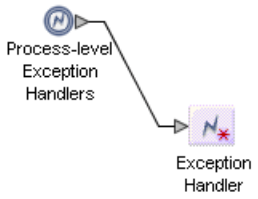
Once the Exception Handler is configured, the red icon will turn green and the Exception name appears on the link. See Figure 35 for an example of the configured Exception Handler.

Catch All Exceptions

To use **Catch All Exceptions:**

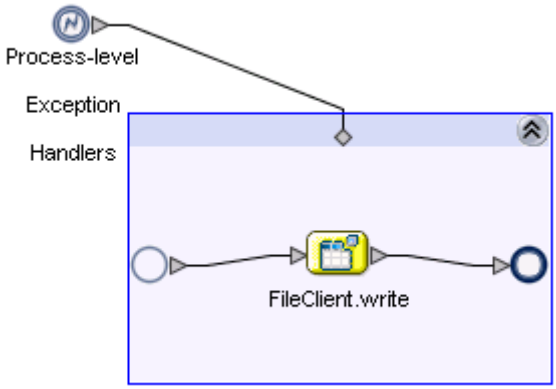
1 Drag the **Catch All Exceptions** activity to a **Scope** in the Business Process Designer or onto the canvas, as shown in Figure 36. This will capture any and all exceptions that occur.

Figure 36 Catch All Exceptions (Process-Level)



2 Double-click the Exception activity to configure the Exception Handler, as shown in Figure 36.

Figure 37 Configure Catch All Exceptions

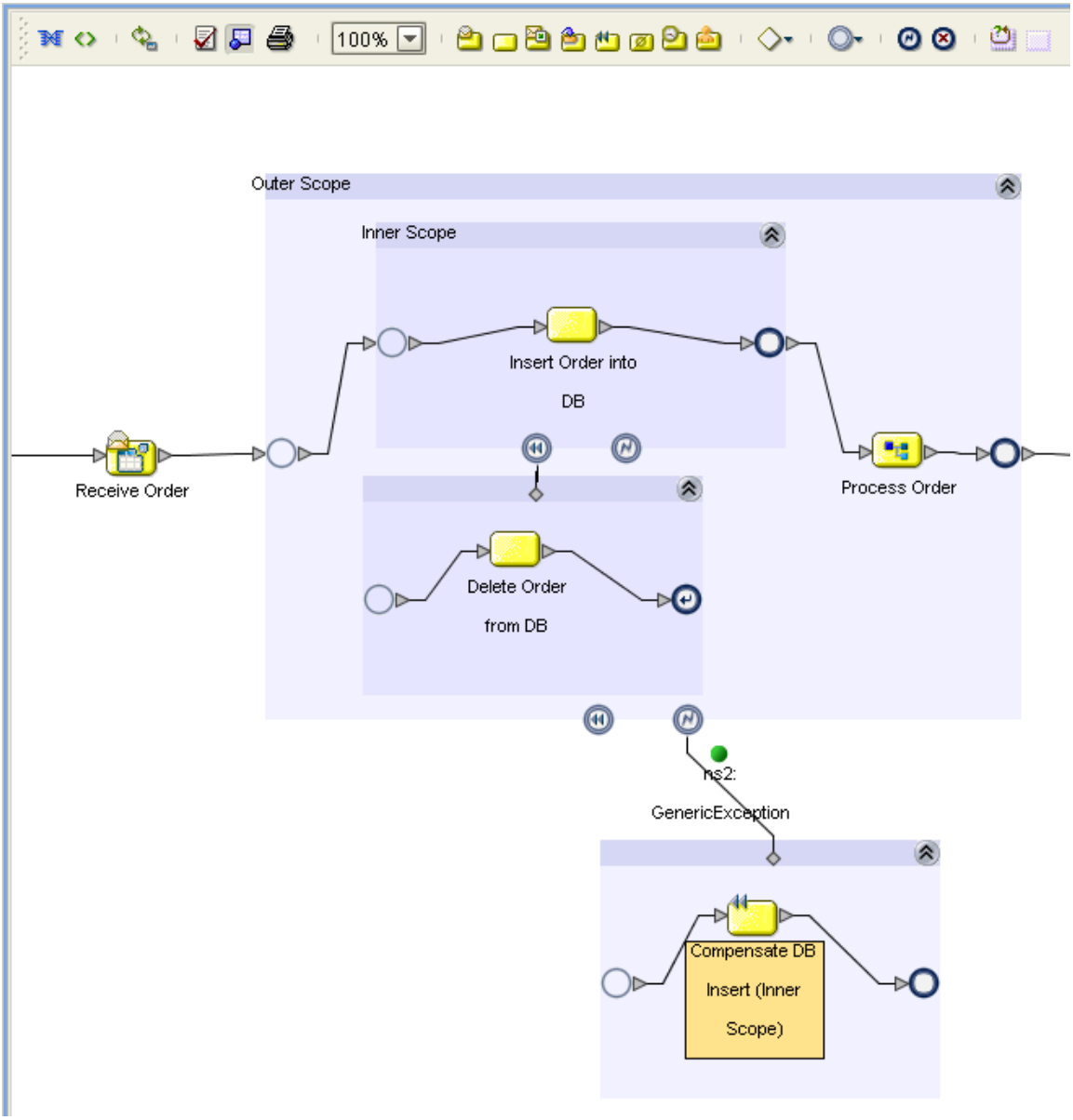


You can configure the Exception Handler to perform an action when an exception is encountered. In Figure 36, the Exception Handler has been configured to write the exception to a file.

5.1.2 Compensation Overview

Compensation Handlers allow you to define processes to compensate previously executed system interactions. This can be used in conjunction with Exception Handling logic when a compensating transaction needs to be invoked. The logic for compensating a transaction can be simple or complex, but either way, it will be defined as a business process within the Compensation Handler.

Figure 38 Example of Compensation Handling



Compensation you to create the process flow for executing complex compensations. Exception Handlers for parent scopes invoke the correct Compensation Handlers in the appropriate order.

Using the Compensation Activity

The Compensation Activity, which is a modeled as a "Throw", is used in an Exception Handler to initiate the compensation process. Compensation is always used with the Scope and Exception Handling elements. See Figure 38 for an example of a configured Compensation Handler. Refer to this figure when reviewing the following steps:

1 Create a scope ("Outer Scope")

2 Create another scope inside of the Outer Scope (“Inner Scope”)

3 Add a Compensation Activity to the Inner Scope

4 Add an Exception Handler to the Outer Scope

5 Add a Throw to the Exception Handler

6 Configure the Exception, see **Catch Named Exception** on page 53.

7 Configure the properties of the compensation Throw to call the name of the Inner Scope. See **Configure the Compensation Activity** on page 58 for details.

Note: *Although it’s not visible, the entire business process exists as a scope. This allows a user to create a single scope within a business process and design a compensation handler for that scope. In this case, the user will drop the exception handler at the business process level.*

Configure the Compensation Activity

1 Select a Compensation Activity.

2 Click the **Show Property Sheet** toolbar button.

The Property Sheet for the Compensation Activity appears on the right.

Figure 39 Compensation Activity Properties

3 Enter the name of the Scope where the compensation takes place.

Chapter 6

Deploy Business Process Models

6.1 Overview

Topics in this chapter are:

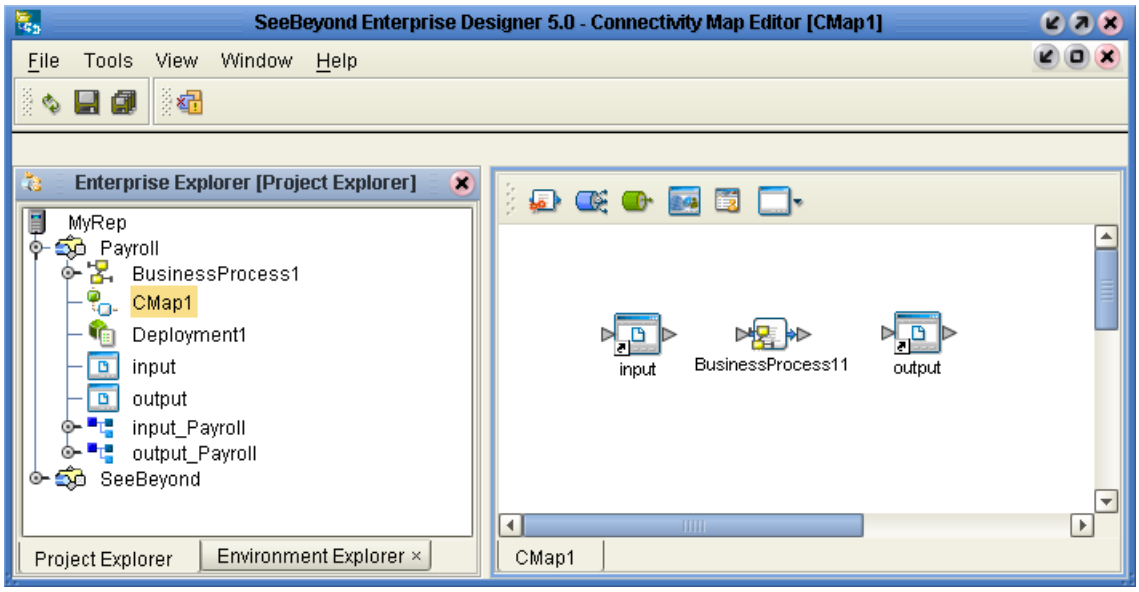
6.1.1 The Business Process and the Connectivity Map

The Connectivity Map represents connection information in the ICAN Suite. The flow is represented at a higher level than in the Business Process Model. eInsight ESB also uses the information in the Connectivity Map to establish and maintain connections to systems for the correct step in a business process.

Creating the Connectivity Map that Includes a Business Process

1. Drag the desired Business Process from the Enterprise Explorer to the Connectivity Map Editor.
2. Add the external systems and components to the Connectivity Map Editor, as shown in Figure 40.

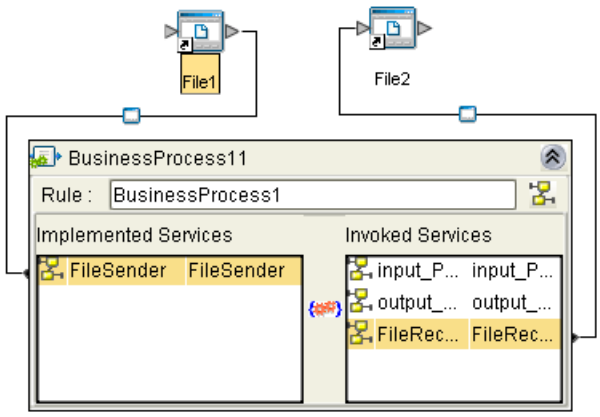
Figure 40 Connectivity Map with Business Process



Open the Business Process to Connect the Activities

1. Double-click on the Business Process in the Connectivity Map to open the Binding Dialog.
2. Connect the Activities to the appropriate component, as shown in Figure 41.

Figure 41 Connectivity Map: Business Process Binding

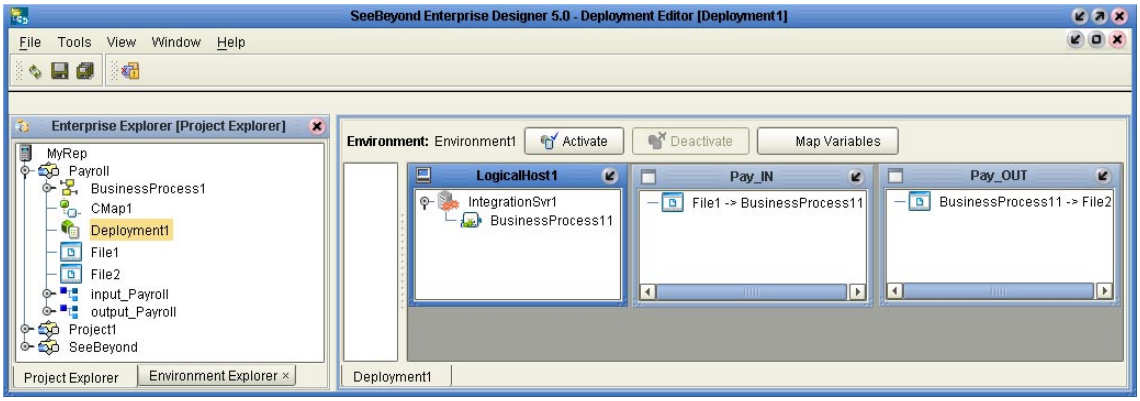


- Receive Activities appear in the left pane.
- Invoke and Reply Activities appear in the right pane.

6.1.2 Deploy a Business Process

A business process is like any other ICAN Suite component. After creating your Environment, Logical Host and other necessary hosts, create a Deployment Profile.

Figure 42 Deployment Profile



The deployable components along with the Business Process from the Connectivity Map will appear in your Deployment Profile. Drag and drop the Business Process to the desired Integration Server. Activate your Deployment Profile to complete the deployment of the components to the target hosts. For more information about Deploying a Project, see the *eGate Integrator User's Guide*.

Chapter 7

Persistence and Monitoring

This chapter discusses Persistence and Monitoring. Both of these features require that you have a database available.

7.1 Overview

eInsight ESB contains database scripts to create the eInsight ESB database schema. The database schema allows you to collect and persist data from your business process. Once the data is persisted, you can also use the Enterprise Manager to monitor business processes. See **Database Support** on page 27 for information about supported databases.

The main areas to be configured in order to have persistence are:

- **eInsight Engine Configuration** on page 62
- **Run Database Scripts** on page 64
- **Configure Persistence for the Business Process** on page 67

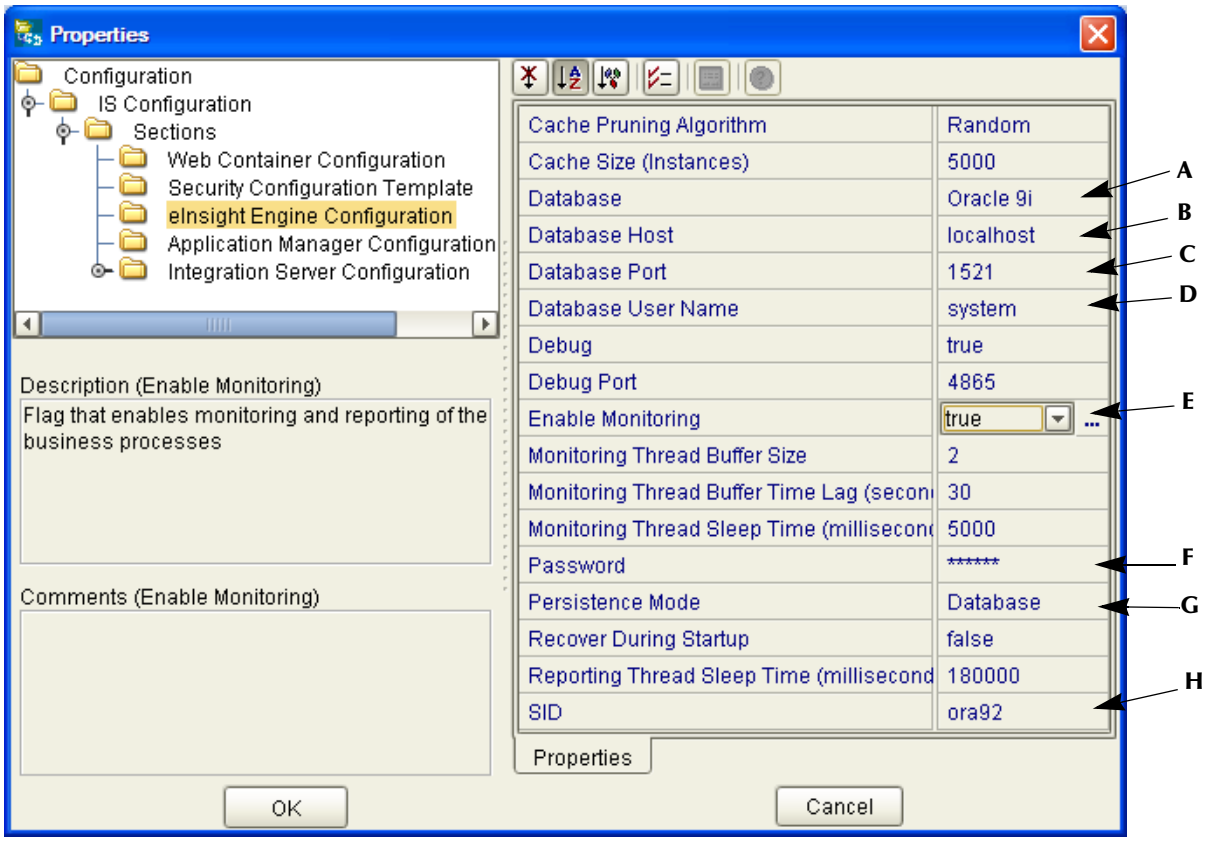
7.1.1 eInsight Engine Configuration

To configure the eInsight Engine

1. Right-click the Integration Server (IS) in your Environment from the Environment Explorer and open the tree structure that contains properties.
2. Navigate to, and select the eInsight Engine.

   The properties dialog appears for the eInsight Engine, as shown in **Figure 43 on page 63**.

Figure 43 eInsight Engine Configuration



3 Configure the following to connect to a database:

- A Database - Select Sybase 12.5, Oracle 8.1.7, Oracle 9i, DB2 8.1or SQL Server 2000.
- B Database Host - Enter the name of the machine where your database resides.
- C Database Port - Enter the port number to connect to your database.
- D Database User Name - Enter the User Name for your database.
- E Enable Monitoring - Select True to use Monitoring.
- F Password - Enter the password for your database user.
- G Persistence Mode - Select Database to use persistence.
- H SID - Enter the database name or SID.

Note: *The other eInsight ESB settings can be left as default, or you can change the options to your preference.*

7.1.2 Create the eInsight Database

To create the runtime recoverability database schema, you can run a database script that is automatically installed with eInsight ESB.

Configure database connection

1. Expand the SeeBeyond folder in the Project Explorer.
2. Expand the eInsight folder that is located under the SeeBeyond folder.
3. Expand the Run Database Scripts folder.
4. Right-click the Database Scripts folder and choose Properties.
5. Enter your database configuration information.

Note: *If you are using a Sybase, DB2 or SQL Server database, you need to have your Database Administrator modify the scripts before you execute. Enter the database user and password that has privileges to your Sybase, DB2 or SQL Server database. See* **View or Modify Database Scripts** *on page 65* *for more information.*

Database Connection Information

Oracle

```
Database Type: oracle
Database DriverName:  oracle.jdbc.driver.OracleDriver
Database URL: jdbc:oracle:thin:@<host>:<port>:<SID>
Database User ID: <userid>
Database Password: *****
```

Sybase

```
Database Type: sybase
Database DriverName:  com.sybase.jdbc2.jdbc.SybDriver
Database URL: jdbc:sybase:Tds:<host>:<port>/<dbname>
Database User ID: <userid>
Database Password: *****
```

SQL Server

```
Database Type: sqlserver
Database DriverName: com.SeeBeyond.jdbc.sqlserver.SQLServerDriver
Database URL:
jdbc:SeeBeyond:sqlserver:<host>:<port>;DatabaseName=<dbname>
Database User ID: <userid>
Database Password: *****
```

DB2

```
Database Type: db2
Database DriverName: com.SeeBeyond.jdbc.db2.DB2Driver
Database URL: jdbc:SeeBeyond:db2://
<hostname>:<port>;DatabaseName=<dbname>;collectionId=JDBCPKG;packageN
ame=JDBCPKG;embedded=true;createDefaultPackage=TRUE
Database User ID: <userid>
Database Password: *****
```

Run Database Scripts

The database user that executes these scripts must have permission to create/delete tables.

Install and Run Database Scripts

1. Expand the SeeBeyond folder in the Project Explorer.
2. Expand the eInsight folder that is located under the SeeBeyond folder.
3. Expand the Run Database Scripts folder.
4. Select the appropriate database install file for Oracle, DB2, SQL Server or Sybase.
5. Right-click the file associated with the appropriate database (Oracle Install, SQL Server Install, DB2 Install or Sybase Install) and select **Run**.

Note: *DBA or Sysadmin/DB2 will have to create the database DB2 instance on the server. The user will need privileges to create objects in* ***createddatabase*** *(tables, indexes, sequences objects etc).*

Uninstall Scripts

1. Expand the **SeeBeyond** folder in the Project Explorer.
2. Expand the eInsight folder that is located under the **SeeBeyond** folder.
3. Expand the **Run Database Scripts** folder.
4. Select the appropriate database file for Oracle, SQL Server, DB2 or Sybase.
5. Right-click the file associated with the appropriate database (Oracle Uninstall, SQL Server Uninstall, DB2 Uninstall or Sybase Uninstall) and select **Run**.

View or Modify Database Scripts

You may open the database scripts and view them within eInsight ESB. You may also modify and run the modified scripts from eInsight ESB. Consult your databse administrator when making changes to the database scripts. You may wish to download the scripts and modify them outside of the product. See **"Download Database Scripts"** for more information.

To View or Modify Database Scripts

1. Expand the **SeeBeyond** folder in the Project Explorer.
2. Expand the eInsight folder that is located under the **SeeBeyond** folder.
3. Expand the **Run Database Scripts** folder.
4. From the **Run Database Scripts** folder, select the script you will modify.
5. Right-click and select **Open** to view the script.
6. If you wish to make changes, the scripts are editable.
7. **Run** the script, see **"Run Database Scripts"**.
8. **Save** or **Discard** your changes.

Note: *You are prompted to Save or Discard your changes when you close the script. To keep the original scripts, you may want to discard your changes, otherwise select Save.*

Modify Database Fields Lengths

You may need to modify database scripts to accomodate larger field lengths in your tables. You may experience errors if your data exceeds the size allowed by the field.

1. Expand the **SeeBeyond** folder in the Project Explorer.
2. Expand the eInsight folder that is located under the **SeeBeyond** folder.
3. Expand the **Run Database Scripts** folder.
4. From the **Run Database Scripts** folder, select the script you will modify.
5. Right-click and select **Open** to view the script.
6. Find the appropriate field(s) and change the field length.

   For example, from **varchar(255)** to **varchar(4000)**.
7. **Run** the script (see **"Run Database Scripts"**) to make changes to the database.
8. **Save** or **Discard** your changes.

Note: *You are prompted to Save or Discard your changes when you close the script. To keep the original scripts, you may want to discard your changes, otherwise select Save.*

7.1.3 Download Database Scripts

This is an alternative method to **"Run the Business Process Database Script" on page 68**. You may wish to download the database scripts and run them outside of the ICAN environment.

Download the compressed script files

To create the runtime recoverability database schema, you can download a Database Script that is automatically installed with eInsight.

1. Expand the SeeBeyond folder in the Project Explorer.
2. Expand the eInsight folder that is located under the SeeBeyond folder.
3. Expand the Download Database Scripts folder.
4. Select the appropriate database file: Oracle, SQLServer, DB2 or Sybase.
5. Right-click the file associated with the appropriate database (Oracle.zip, SQLServer.zip, DB2.zip or Sybase.zip) and select Export.
6. Unzip the database script to a local folder.

Included in the zipped file are:

- install_db.bat - This script will create the tablespace, users, tables, stored procedures, and any initial value.
- uninstall_db.bat - This script reverses what the install_db.bat script creates (drops tables and users, deletes stored procedures).
- database specific sql scripts - These scripts are called by the install_db.bat and uninstall_db.bat commands (such as, create_tables.sql, drop_tables.sql, etc.)

- A Readme.txt file with additional instructions, specific to your database application.

7 Follow the specific instructions in the Readme.txt file, for your database.

To execute Database Scripts

1 Open a command window and navigate to the directory where script is located.

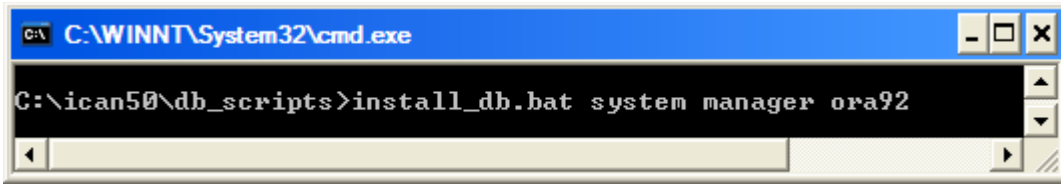
Important: *The database user that executes these scripts must have permission to create tables.*

2 Enter the following at the command prompt, as shown in Figure 44:

```
install_db <user> <password> <tns>
```

- <user> is the database username
- <password> is the database user password
- <tns> is the database or tns name

Figure 44 Install_db.bat



Note: *The default user and password created from these scripts is "einsight". You can modify the user, password, disk space allocated for tables, and user permissions. The table and column definitions should not be modified.*

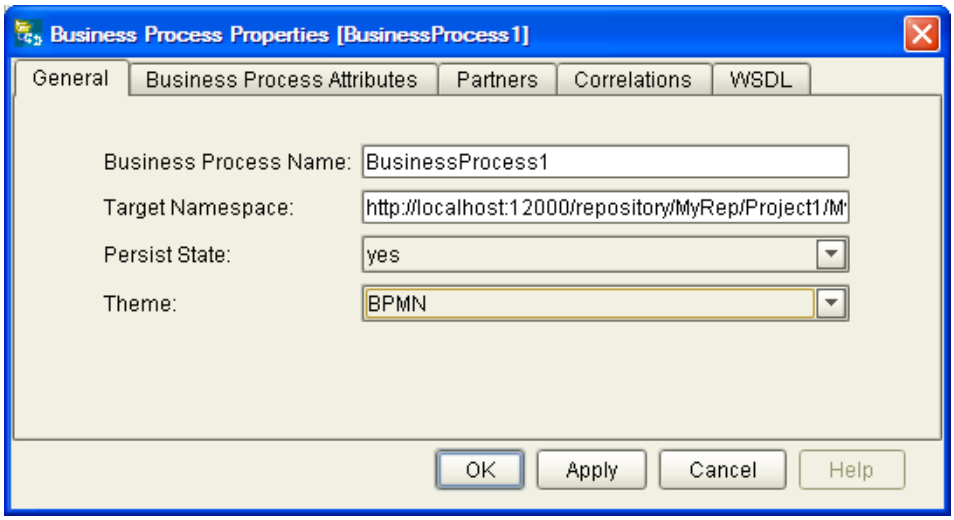
7.1.4 Configure Persistence for the Business Process

1 Right-click your Business Process and select Open Property Sheet.

2 Select Yes for the Persist State option, as shown in Figure 45.

You set the persistence state for individual business processes. The default setting is no for Persist State.

Figure 45 Business Property Sheet



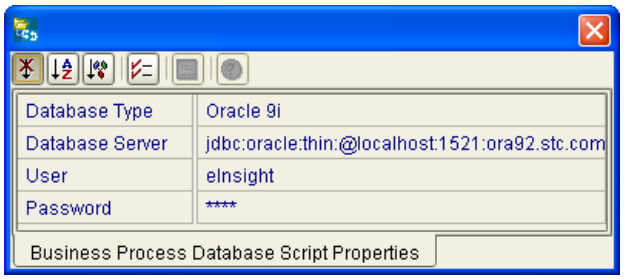
3 Click the Save All toolbar icon.

This creates a Database Install Script option under your Business Process.

Configure connection information

1 Right-click the Database Install Script under the Business Process and select Properties.

Figure 46 Properties of Database Scripts



2 Enter the connection information for your database (as shown in Figure 46).

Run the Business Process Database Script

1 From the Project Explorer, expand your Business Process.

2 Expand the Database Scripts folder.

3 From the Database Scripts folder, right-click the appropriate database and select Run.

The scripts complete the database creation process.

Uninstall Script for the Business Process

1 From the Project Explorer, expand your Business Process.

2 Under the Business Process, expand the Database Scripts folder.

3 Right-click the appropriate uninstall script and choose Run.

7.2 Monitoring

The Enterprise Manager allows users to quickly identify problems with components or systems. From the Enterprise Manager, you can double-click on business process components to go directly to the problem.

From the Enterprise Manager (Monitoring interface), you can:

- Filter the list of displayed instances to quickly identify exceptions.
- Navigate to particular versions of a Business Process to monitor the progress of instances.
- Use a Web based interface allows users to securely access the monitoring environment over the internet.

Setup Monitoring

Once persistence is configured, you can use the Enterprise Manager to monitor your business process instances. It is important that you use the following procedures to ensure the proper view of the business process appears in the monitor.

In order to view the special tools for the business process state diagram in the Enterprise Manager, you must have the Enterprise Manager Plug-in file uploaded and installed. This file contains the Adobe SVG Plug-In, which allows you to see the graphic model in the Enterprise Manager. See the *eGate Integrator System Administration Guide* for information. The SVG Plug-In is not required to use monitoring.

To monitor a new business process

1 Open and save each business process.

Note: *If your business process has any User Activities or special OTDs, you should also open and close each of these to make sure they will register with the monitor.*

2 Right-click each business process and select Check In.

3 Right-click each business process and select Check Out.

4 Rerun the database scripts (see **Run Database Scripts** on page 64).

5 Save the business process.

6 Activate the deployment profile.

7 Launch the Enterprise Manager.

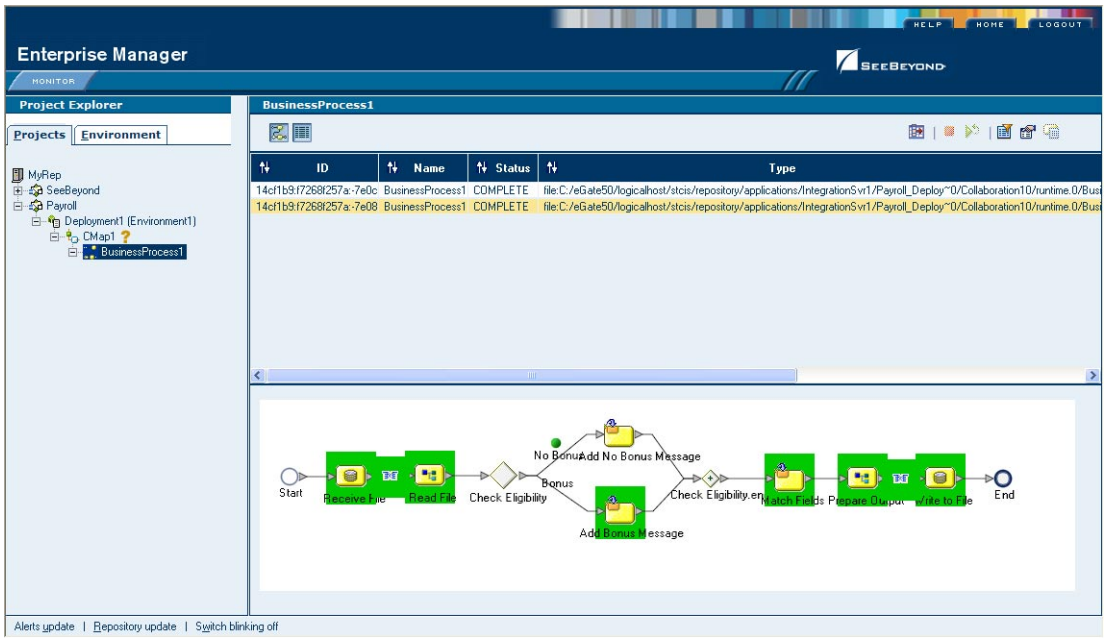
8 Select the Home tab.

9 Select the Monitor icon to bring up the tree structure which allows you to navigate through projects or environments.

10 Select the Projects tab.

11 Navigate to the correct Project/Deployment Profile/Connectivity Map, and select the Business Process name.

Figure 47 Monitor View



To monitor a modified business process

1 Check Out the Business Process (if necessary).

2 Modify the business process.

3 Click Save to save the changes.

4 Right-click the business process and select Check-In.

5 Right-click the business process and select Check-Out.

6 Run the database script (see **Run Database Scripts** on page 64).

7 Activate the deployment profile.

To monitor an imported project

1 Import the project.

2 Check Out the business process(es).

3 Open and save each business process.

Note: *If your business process has any User Activities or special OTDs, you should also open and close each of these to make sure they will register with the monitor.*

4 Right-click each business process and select Check In.

5 Right-click each business process and select Check Out.

6 Rerun the database scripts (see **Run Database Scripts** on page 64).

7 Save the business process.

8 Activate the deployment profile.

9 Launch the Enterprise Manager.

10 Select the Home tab.

11 Select the Monitor icon to bring up the tree structure which allows you to navigate through projects or environments.

12 Select the Projects tab.

13 Navigate to the correct Project/Deployment Profile/Connectivity Map, and select the Business Process name.

7.2.1 Monitoring Options

Using the monitor console, you can view and interact with Business Processes instances in both Project and Environment views.

- In Project view, you can start, stop and set the cache.
- In Environment view, you can start, stop, and examine Business Process Instances using an exact graphical model of the business process.

Business Process Monitoring Tools

Interactive monitoring tools allow you to control the view of Business Process Instances, and manage the display of Instance details. The monitor console provides a suite of controls arrayed in toolbar format across the top of the Details window. Before you start monitoring, you should be well acquainted with these tools. This page describes the tools, in left-to-right order, as they appear in the console.

Controlling the Business Process display mode

The display of Business Process Instances and the Instance List in the console viewer are controlled by the buttons described in the following table. (These buttons are located in the upper left of the Details window.)

Table 7 Monitor: Display Mode

	**Hide Business Process** hides the rendered image of a Business Process Instance in the **Details** window.
	**Show Business Process** renders the image of a Business Process Instance in the **Details** window.
	**Show Instance List** displays the attributes of the current Business Process Instance in list format, and adds the tools described in the following table.

Table 7 Monitor: Display Mode (Continued)

	**Hide Instance List** hides the attributes of the current Business Process Instances, and removes the instance tools from the interface.

Controlling the display of instance data

When the monitor is in Instance monitoring mode, you can manipulate the view of Instance data using the buttons described in the following table. (These buttons are located in the upper right of the Details window).

Table 8 Monitor: Display Instance Data

	**Choose Preferences** allows you to add, move, and sort the columns in the Business Process Instance.
	**Start** starts a stopped Business Process Instance.
	**Stop** stops a Business Process Instance.
	**Filter Instances** allows you to set criteria to display a specific instance or group of Instances.
	**Business Process Instance Attributes** displays the attributes of an Instance when the instance is selected in the **Instance List**.
	**Activity Details** allows you to see the details of an Activity. Defines a step within a particular Business Process. when the Activity is selected in the **Instance List**.

Note: *Please see the Enterprise Manager Help for more information about monitoring business processes.*

7.3 Logging

The eInsight Engine coordinates all business process-related activity of a deployed project. The engine runs within the SeeBeyond Integration Server.

7.3.1 Setting Log Levels

You cannot set the log level of the eInsight Engine from the ICAN Monitor. Instead, you must perform the following steps.

Set eInsight Log Level

1. Using a text editor, open the **log4j.properties** file in the ***ICAN-root*/logicalhost/logconfigs/IS_*integration-server-name*** directory.
2. Add the following line:

   ```
   log4j.category.com.stc.bpms.bpelImpl=<loglevel>
   ```

   The values for <*loglevel*> are:

   - Debug
   - Information
   - Warning
   - Error
   - Fatal

Chapter 8

Implementation

The two case studies in this chapter are designed to illustrate functionality, in addition to showing working examples of business process implementations.

8.1 Overview

Implementing a business process is translating the vision of the business user into a functioning system. You implement a business process model by using modeling components. Business process modeling components are mostly pre-configured but some may require modification.

This chapter provides three ways to learn about implementing a project. Depending on your needs, you can:

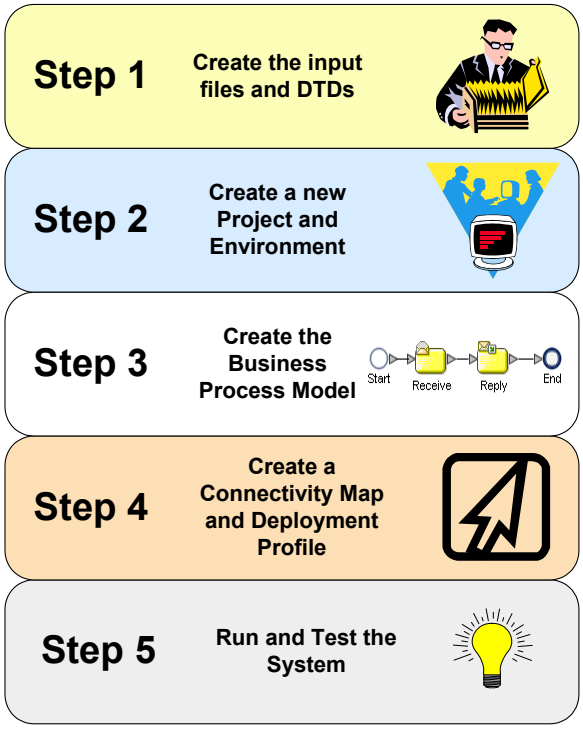
- Create the end to end sample from scratch: **"Case Study: Payroll Processing" on page 76**
- Import the end to end sample and run it: **"Import the End to End Sample" on page 94**
- Import and run a sample that demonstrates the correlation feature: **"Import the Correlation Sample" on page 95**.

8.1.1 Road Map

Each type of implementation involves a different approach, however, there are certain similarities. To give you an overview of the complete process, the following implementation road map contains high-level steps for this implementation. The road map is further refined and given more detail in the case study that immediately follows.

Figure 48, illustrates the major steps in the integration process for this implementation.

Figure 48 Integration Road Map



Step 1: Create the Input Files and DTDs

The first step in this implementation requires that you create two input files for the system. In an actual implementation, your input files may come from an external system.

The first step also entails creating your Document Type Definition (DTD) files. The DTD file tells the system which elements it should expect from the input files and how to format the output data.

Step 2: Create a New Project and Environment

In the second step of this implementation, you will create a new Project where your business process will reside and a new Environment for your Project.

Step 3: Create the Business Process Model

In this step, you will create a new business process, add the modelling elements and link them together. You will also configure the modelling elements and links to process the data.

Step 4: Create a Connectivity Map and Deployment Profile

When you create the Connectivity Map, you are making the connections between the system components and the external systems. You will also create a Deployment Profile that you will activate when you put the system into production.

Step 5: Deploy and Test the System

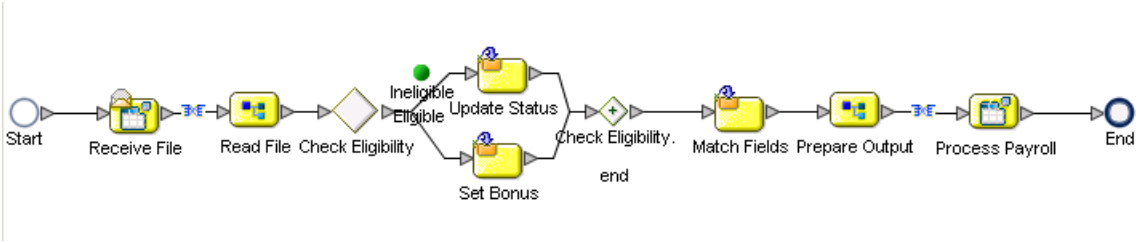
To run the system, you must invoke the Bootstrap. This action will pick up your Deployment Profile and execute your business process. Once the system processes your input files, an output file is created. To verify that this implementation has completed properly, check the output file.

8.2 Case Study: Payroll Processing

This case study begins with a description of the scenario and then shows how to set it up. The case study discussed in this chapter illustrates a simplified implementation of payroll processing. In this case, eInsight ESB receives payroll data as XML files.

Once eInsight ESB has received the data, a check is made to see if the employee is eligible for a bonus, if they are, the bonus is set. Finally, the payroll is processed and a message added to the paystub, indicating whether a bonus was paid. Figure 49 shows the components involved in the business process implementation.

Figure 49 Business Process Model



1. The first File eWay picks up the input XML files containing the employee's information from a local folder on your computer. The payroll information is used to start a business process instance. eInsight ESB retrieves the information and uses it to execute the decision logic.
2. eInsight ESB uses the decision logic information it contains to check the employee's probation status and continues along one path or the other, depending on that status.The decision logic determines whether the employee is eligible for a bonus, and then moves forward to the next activity in the business process based on the result.
3. If the employee is eligible for a bonus, the next activity is **Set Bonus**; if the employee is not eligible, the next activity is **Update Status**.

- Let's assume the employee is not eligible for a bonus because they have been employed for less than three months. The Probation status is *Yes*, therefore eInsight ESB proceeds to the corresponding activity, **Update Status,** in the business process.

Once the Comment and Bonus fields are updated, eInsight ESB moves forward to the next activity in the business process—**Match Fields**.

- Otherwise, the employee is eligible for a bonus and eInsight ESB uses the information to verify eligibility. When the **Set Bonus** activity is finished, eInsight ESB moves forward to the next activity in the business process—**Match Fields**.
- The **Match Fields** activity uses the Business Rule function to match the data fields in your input file to the data format of your output file.

4 eInsight ESB then proceeds to the **Prepare Output** activity and finally the **Process Payroll** activity. **Process Payroll** is a File eWay that performs two functions: it sends a status report to the payroll system, and also writes the data to the output file.

5 eInsight ESB has performed the final activity in the business process and completes successfully.

8.2.1 Before You Begin

To complete this exercise, you need to have the following:

- ICAN 5.0 products installed:
  - eGate Integrator
  - eInsight Enterprise Service Bus
  - File eWay
- A directory on your local drive named **data**.

Input Files and DTDs

The sample system you are creating requires input information. For this exercise, you will create two input files: **Eligible.xml** and **Ineligible.xml**. These files are in an XML format. You do not need to have an XML editor to create these files. Any simple text editor will work.

The system you are creating also needs a structure for receiving information. That structure is described in the Document Type Definition (DTD) files.

Input XML Files

The XML files that you create here, contain the data that the system receives and changes to create your final output.

Create the Input files

1 Copy the following code sections each to separate text files:

- Eligible.xml

```
<?xml version="1.0" encoding="UTF-8"?>
     <Payroll>
          <FirstName>Vanessa</FirstName>
          <LastName>Smith</LastName>
    <Probation>No</Probation>
          <Comments></Comments>
```

```
        <Bonus></Bonus>
    </Payroll>
```

- **Ineligible.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
    <Payroll>
        <FirstName>Tonya</FirstName>
        <LastName>Lee</LastName>
   <Probation>Yes</Probation>
        <Comments></Comments>
        <Bonus></Bonus>
    </Payroll>
```

2 Rename the files to **Eligible.xml** and **Ineligible.xml**.

3 Save the files to **C:\data**.

Input and Output DTD Files

Create the Document Type Definitions

These files are later used to define the way data is mapped in the system.

1 Copy the following code sections each to separate text files:

- **Input.dtd**

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Payroll (FirstName, LastName, Probation, Comments, Bonus)>
<!ELEMENT FirstName (#PCDATA)>
<!ELEMENT LastName (#PCDATA)>
<!ELEMENT Probation (#PCDATA)>
<!ELEMENT Comments (#PCDATA)>
<!ELEMENT Bonus (#PCDATA)>
```

- **Output.dtd**

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Payroll (FName, LName, Message, BonusTotal)>
<!ELEMENT FName (#PCDATA)>
<!ELEMENT LName (#PCDATA)>
<!ELEMENT Message (#PCDATA)>
<!ELEMENT BonusTotal (#PCDATA)>
```

2 Rename the files to **Input.dtd** and **Output.dtd**.

3 Save the files to a local folder.

8.2.2 Create a New Project and Environment

Create a new Project

1 Launch the Enterprise Designer.

2 Right-click your Repository and select **Project.**

A new Project appears in your Project Explorer tree structure.

3 Rename the Project to **Payroll**.

4 Click the **Save All** toolbar button to save your changes.

Add the Input and Output DTD files to your Project

1. Right-click your Payroll Project and select **New: Object Type Definition**.

   The **New Object Type Definition Wizard** appears.
2. Select the **DTD** option from the choices given.
3. Click **Next** to continue.
4. Navigate to the local folder where your **Input.dtd** and **Output.dtd** files are located.
5. Select the **Input** and **Output** DTD files. (You can select multiple files using the CTRL key).
6. Click **Next** to continue.

   The **Select Document Elements** dialog appears.
7. Select both **DTD** files.
8. Click **Next** to continue.

   The **Select OTD** dialog appears.
9. Click **Finish** (do not change any of the default settings).

Environment

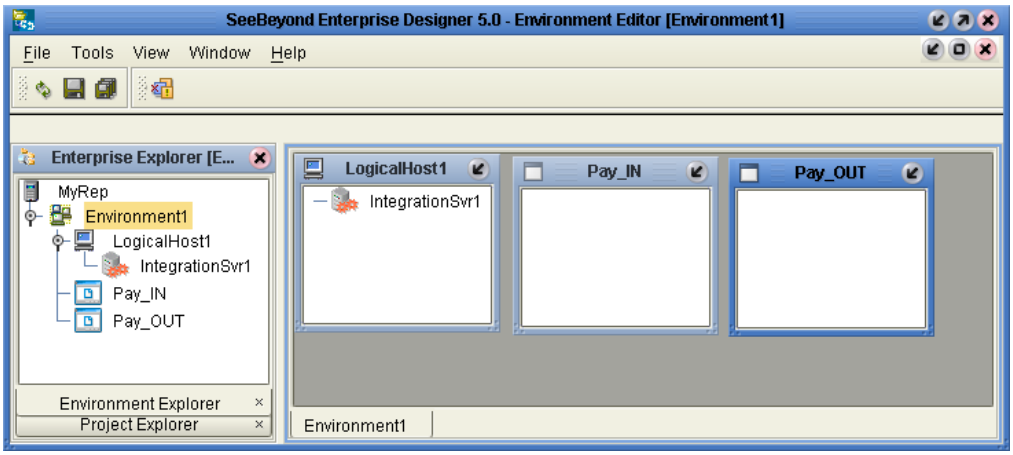
Create a new Environment

1. Select the **Environment Explorer** tab from the Enterprise Designer.
2. Right-click on your **Repository** and select **New Environment**.
3. Right-click on your **Environment** and select **New Logical Host**.
4. Right-click on your **Environment** and select **New File External System**.

   The system prompts you to name the **File External System**.
5. Enter **Pay_IN** as the name of your **File External System**.
6. Select **Inbound File eWay** as the **External System Type**.
7. Repeat step 4 and name the **File External System**: **Pay_OUT**.
8. Select **Outbound File eWay** as the **External System Type**.
9. Right-click the **Logical Host** and select **New SeeBeyond Integration Server**.

   Your new Environment will look like Figure 50.

Figure 50 New Environment



8.2.3 Create the Business Process Model

Create a new Business Process

1. Click the **Project Explorer** tab and right-click on your **Payroll** project.
2. Select **New**: **Business Process**.

   A new Business Process appears in your directory tree under your Payroll project and a blank Business Process appears in the Business Process Designer (right pane).

Add Modelling Elements to the Business Process Model

This section contains detailed instructions to build your model.

Add the File Receive Activity

This activity uses an inbound **File eWay**.

1. Double-click the **SeeBeyond Project** from the Project Explorer tree view.
2. Double-click **eWays** under the SeeBeyond Project.
3. Double-click **File** under eWays.
4. Double-click **FileClient** under File.
5. Select and drag the **Receive** activity from **FileClient** to the Business Process Designer.
6. Place the **Receive** activity to the right of the **Start** activity.

Add the Unmarshal Activity

This activity unmarshalls the input data.

1. Double-click your **Payroll** project from the Project Explorer tree view.
2. Expand the **Input_Payroll DTD**, located under your Project.
3. Select and drag the Input_Payroll **Unmarshal** activity.

4 Place the activity to the right of the **FileClient Receive** activity.

Add the Decision Element

The decision element contains logic that determines what will happen to the incoming data. You will configure the logic in a later step.

1 Select **Branching Activities** from the Business Process Toolbar.

2 Select **Decision** from the **Branching Activities** drop-down list.

3 Drag the **Decision** to the Business Process Designer.

The **Decision** element and **Decision.end** appear on the Business Process Designer canvas.

4 Place the Decision to the right of the **input _Payroll.unmarshal** activity. Leave a space between the Decision and Decision.end.

Add the Business Rule Activities

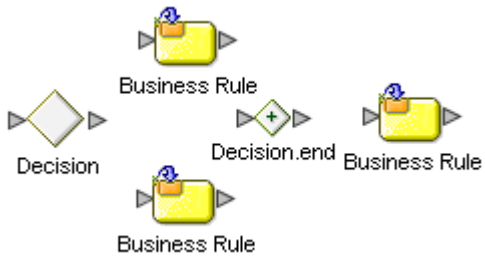
The Business Rule Activities allow you to map and transform data. You will configure the Business Rule Activities later in this exercise.

1 Select the **Business Rule** Activity from the Business Process Toolbar and drag the activity to the Business Process Designer.

2 Repeat step 1 until you have three **Business Rule Activities** on your canvas.

3 Group the **Business Rule Activities** as shown in Figure 51.

Figure 51 Building the Model



Add the Marshal Activity

This activity marshals the data and prepares it for output.

1 Double-click your **Payroll** project from the Project Explorer tree view.

2 Click the icon next to **output_Payroll DTD** to expand.

3 Select and drag the output_Payroll DTD **Marshal** operation to the Business Process Designer.

4 Place the activity to the right of the last **Business Rule** Activity.

Add the File Write Activity

This activity is an outbound **File eWay**.

1 Double-click the **SeeBeyond Project** from the Project Explorer tree view.

2 Double-click **eWays** under the SeeBeyond Project.

3 Double-click **File** under **eWays**.

4 Double-click **FileClient** under **File**.

5 Select and drag the **Write** activity from **FileClient** to the Business Process Designer.

6 Place the **Write** activity to the left of the **End** activity.

8.2.4 Configure Modeling Elements

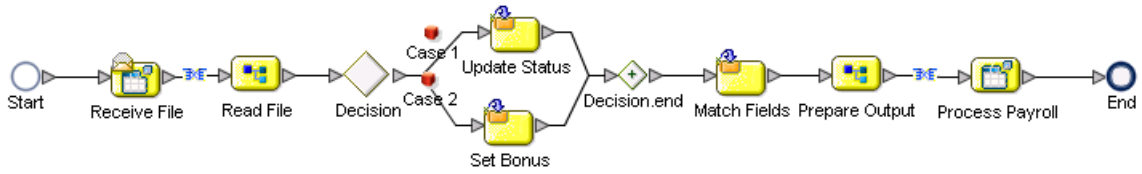
Draw links to connect the model

1 Move your mouse over the **Start** activity until a hand appears.

2 Click and hold your mouse to drag a **Link** between the elements.

3 Repeat steps 1 and 2 to connect the entire model as shown in Figure 52

Figure 52 Linked Model



Rename Modeling Elements

You should rename the elements to represent the activity's role in the business process. This makes it easier to understand the model.

1 From the Business Process Designer, click an element label (the name or title under the activity).

A sunken box appears around the label.

2 Type to rename the element.

3 See Table 9 and rename each of the elements as described in steps 1 and 2.

Table 9 Rename Elements

Current Name	Rename to
FileClient.Receive	Receive File
input_Payroll.unmarshal	Read File
Decision	Check Eligibility
Case 1	Eligible
Case 2	Ineligible

Table 9 Rename Elements

Current Name	Rename to
Business Rule (upper)	Update Status
Business Rule (lower)	Set Bonus
Business Rule	Match Fields
output_Payroll.marshal	Prepare Output
FileClient.write	Process Payroll

Configure Business Rules

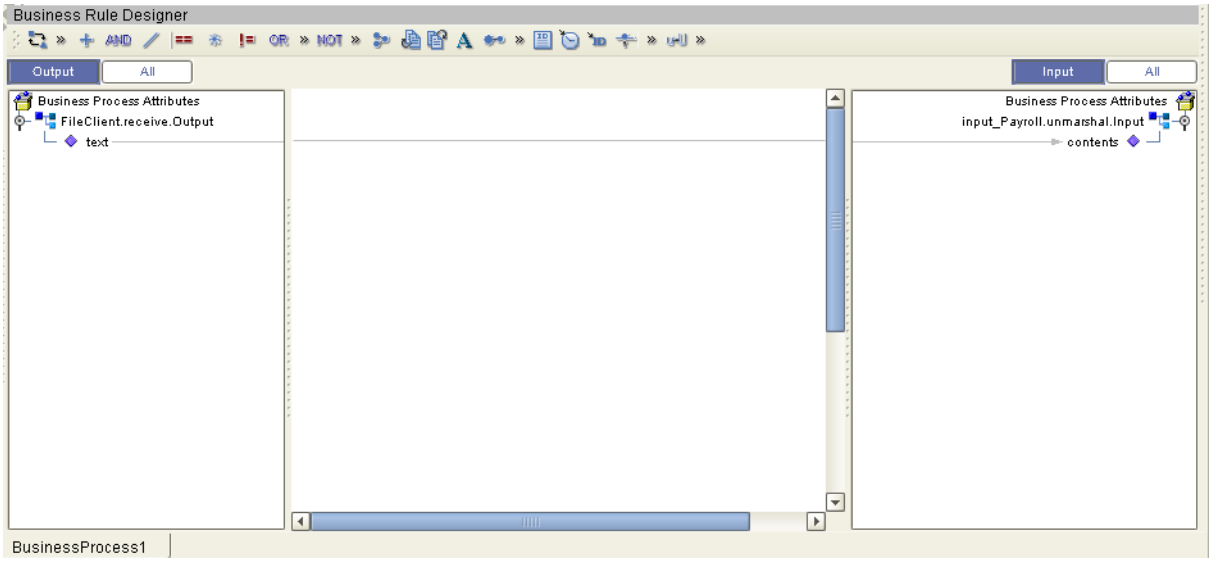
Add Business Rules to Links

There are two links in this exercise that use Business Rules applied to links, to move data through the business process model.

1. From the Business Process Designer, select the link between **Receive File** and **Read File**.
2. Right-click on the link and select **Add Business Rule**.
3. Double-click the link with the new Business Rule icon or select the toolbar icon called **Map Business Process Attributes**.

   The Business Rules Designer appears in the lower pane of the Business Process Designer.
4. Link the **text** node to the **contents** node, as shown in Figure 53 on page 83.

Figure 53 Add Business Rules to Links



5. Select the link between **Prepare Output** and **Process Payroll**.
6. Right-click on the link and select **Add Business Rule**.

7 Double-click the link with the new Business Rule icon or select the toolbar icon called **Map Business Process Attributes**.

The Business Rules Designer appears in the lower pane of the Business Process Designer.

8 Link the **contents** node to the **text** node.

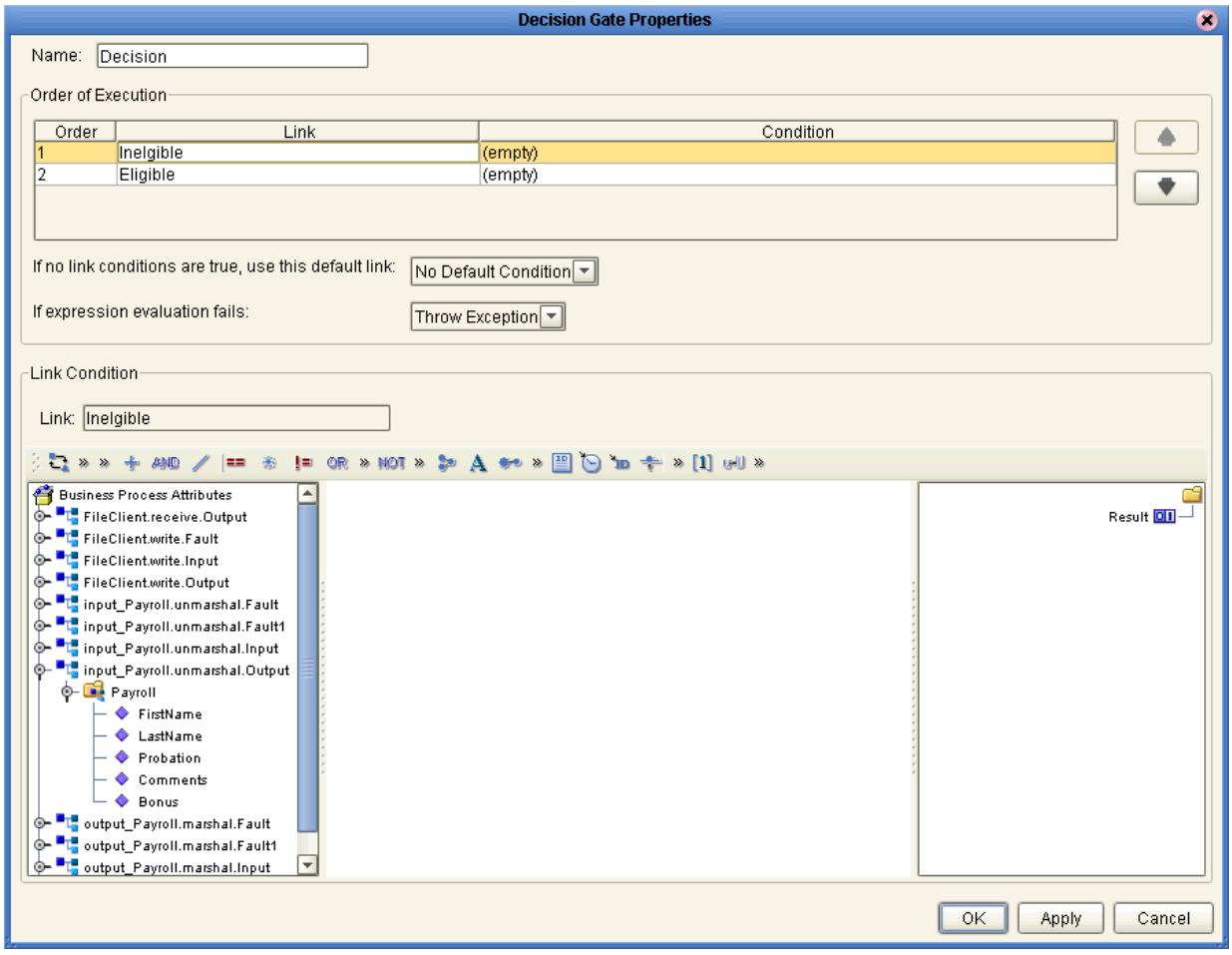
Decision Element

Configure the Decision Logic

1 Double-click the **Decision** element.

The **Decision Gate Properties** dialog appears, shown in Figure 54.

Figure 54 Decision Properties



Configure Case 1

2 Select the case: **Ineligible**.

3 Select the **String Literal** icon from the **Method Pallet** and drag it to the Business Rules Designer.

The **Input** dialog appears.

4 Type **Yes** and click **OK**.

5 Select the **equal** method from the Method Pallet and drag it to the Business Rules Designer.

6 In the **Link Condition** section, find the **input_Payroll.unmarshal.Output** node and expand it.

7 Expand the **Payroll** node and select **Probation**.

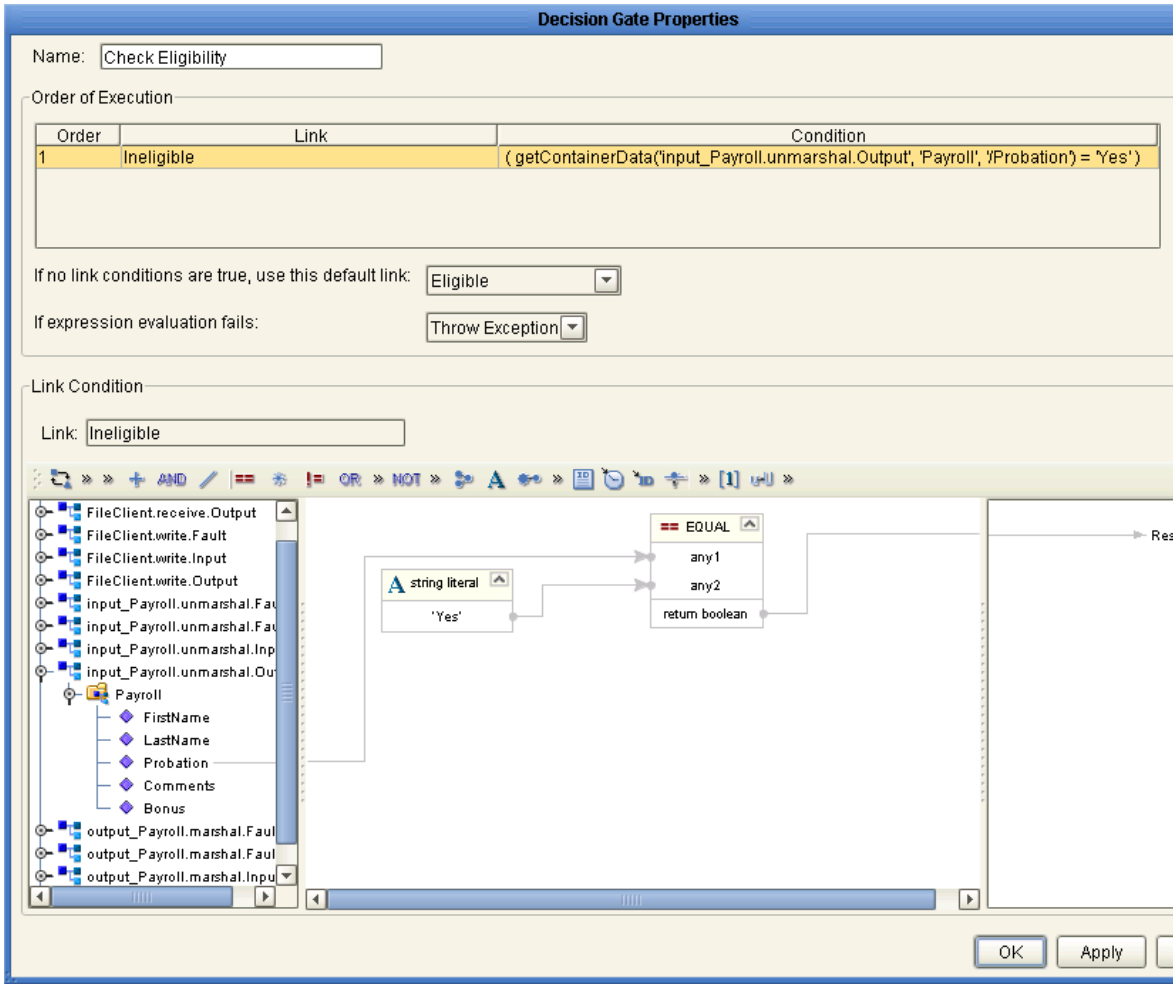
8 Drag a link from the **Probation** node and connect it the **equal** method box, where you see **Any 1**.

9 Drag a link from the **equal** method box where you see **Any 2** to the **String Literal** method box, and connect.

10 Drag a link from the **Return Boolean** section of the **equal** method box, to the **Result (boolean)** pane on the right.

Your **Decision** mapper should look like Figure 55 on page 85.

Figure 55 Completed Decision Gate Properties



Configure Case 2

1. Select the case: **Eligible**.
2. Locate the **Default Condition** drop-down list.
3. Select Eligible from the drop-down list.
4. Click **OK** to exit the **Decision Gate Properties** dialog.

Your business process model should look like **Figure 49 on page 76**.

8.2.5 Configure Business Rule Activities

Update Status Activity

If an employee, in our example, is on Probation, the employee is ineligible for a bonus. In this case, the activity will take the path of the **Update Status** Activity. The **Comments** field is set to let the employee know that they are ineligible for a bonus and the **Bonus** field is set to **$0.00**.

Configure the Update Status Activity

1. Select the **Update Status** Activity.
2. Click the **Map Business Process Attributes** toolbar icon to see the Business Rules Designer.
3. Drag the **Concat** method onto the Business Rules Designer from the Method Palette.
4. Drag the **String Literal** method onto the Business Rules designer from the Method Palette.

   The **Input** dialog appears.
5. Type **Not Eligible for Bonus**.
6. Link the **String Literal** method to **string1** on the **Concat** method
7. On the left pane, expand the **input_Payroll.unmarshal.Output** node and then expand the **Payroll** node.
8. Select **Comments** and drag a link to **string2** on the **Concat** method.
9. On the right pane, expand the **input_Payroll.unmarshal.Output** node and then expand the **Payroll** node.
10. Drag a link from **Return String** on the **Concat** method to **Comments** on the right pane.
11. Drag another **Concat** method onto the Business Rules Designer from the Method Palette.
12. Drag a **String Literal** method onto the Business Rules designer from the Method Palette.

    The **Input** dialog appears.
13. Type **$0.00** to set the Bonus amount.

14 Link the **String Literal** method to **string1** on the **Concat** method

15 On the left pane, expand the **input_Payroll.unmarshal.Output** node and then expand the **Payroll** node.

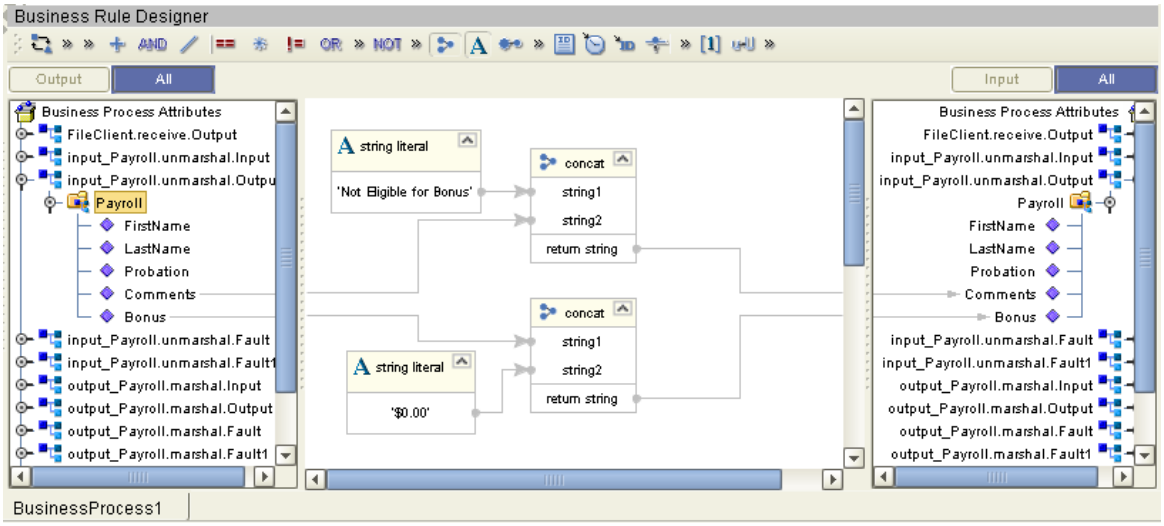
16 Select **Bonus** and drag a link to **string2** on the **Concat** method.

17 On the right pane, expand the **input_Payroll.unmarshal.Output** node and then expand the **Payroll** node.

Drag a link from **Return String** on the **Concat** method to **Bonus** on the right pane.

When you are done, your screen should look like Figure 56.

Figure 56 Update Status Activity



Set Bonus Activity

The **Set Bonus** Activity sets the **Bonus** and **Comments** fields for employees that are eligible for a **Bonus**.

1 Select the **Set Bonus** Activity.

2 Click the **Map Business Process Attributes** toolbar icon to see the Business Rules Designer.

3 Drag the **Concat** method onto the Business Rules Designer from the Method Palette.

4 Drag the **String Literal** method onto the Business Rules Designer from the Method Palette.

The **Input** dialog appears.

5 Type **Eligible for Bonus**.

6 Link the **String Literal** method to **string1** on the Concat method

7 On the left pane, expand the **input_Payroll.unmarshal.Output** node and then expand the **Payroll** node.

8 Select **Comments** and drag a link to **string2** on the **Concat** Method.

9 On the right pane, expand the **input_Payroll.unmarshal.Output** node and then expand the **Payroll** node.

10 Drag a link from **Return String** on the **Concat** method to **Comments** on the right pane.

11 Drag another **Concat** method onto the Business Rules Designer from the Method Palette.

12 Drag a **String Literal** method onto the Business Rules Designer from the Method Palette.

The **Input** dialog appears.

13 Type **$1500** to set the bonus that all eligible employees will receive.

14 Link the **String Literal** method to **string1** on the **Concat** method

15 On the left pane, expand the **input_Payroll.unmarshal.Output** node and then expand the **Payroll** node.

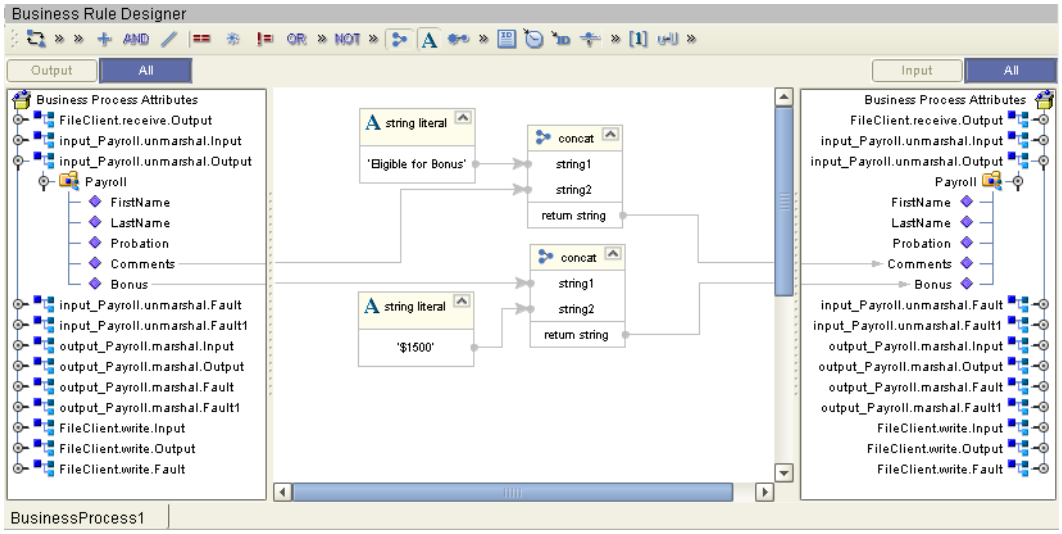
16 Select **Bonus** and drag a link to **string2** on the **Concat** Method.

17 On the right pane, expand the **input_Payroll.unmarshal.Output** node and then expand the **Payroll** node.

18 Drag a link from **Return String** on the **Concat** method to **Bonus** on the right pane.

When you are done, your screen should look like Figure 57.

Figure 57 Set Bonus Activity



Match Fields Activity

The Match Fields Activity maps the input data into the proper format for output.

1 Select the **Match Fields** Business Rule Activity.

2 On the left pane, expand the **input_Payroll.unmarshal.Output** node and then expand the **Payroll** node.

3 On the right pane, expand the **output_Payroll.marshal.Input** node and then expand the **Payroll** node.

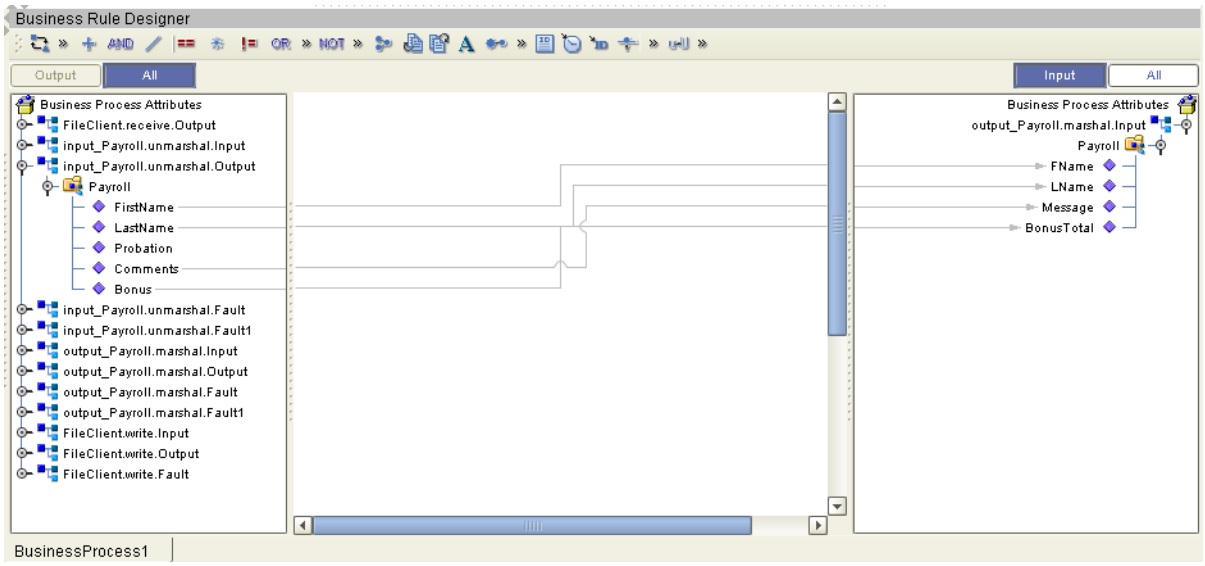
4 Draw a link between the following fields:

Table 10 Match Fields

From	To
FirstName	FName
LastName	LName
Comments	Message
Bonus	BonusTotal

Your screen will look like Figure 58, when you are finished.

Figure 58 Match Fields Activity



8.2.6 Create a Connectivity Map and Deployment Profile

Create the Connectivity Map

1 Right-click your Project and select **New: Connectivity Map**.

A new node will appear under your Project. The default name is **CMap1**.

2 Select the **External Applications** toolbar icon and select **File External Applications**.

3 Drag the **File** icon to the Connectivity Map canvas.

4 Drag a second **File** icon to the canvas.

5 Select **BusinessProcess1** from the Project Explorer and drag it to the canvas.

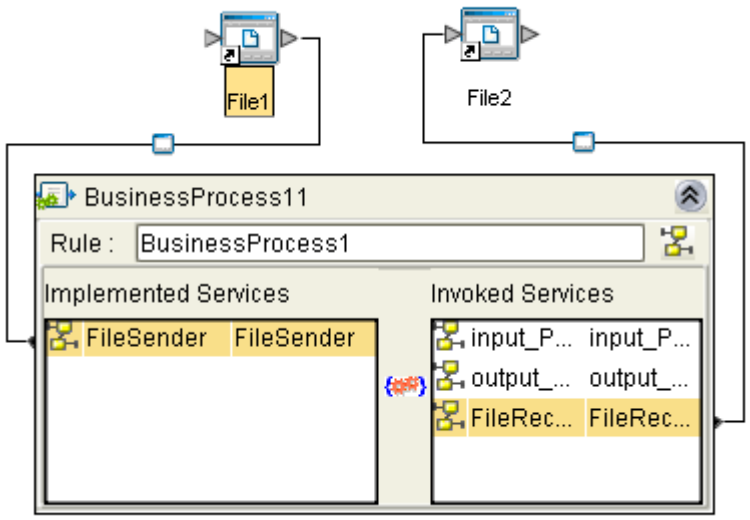
6 Place the **Business Process** between the two **File** icons.

Configure the Business Process

1. Select your **Business Process** from the Project Explorer and drag it to the canvas.
2. Double-click the **Business Process**.

   The **Business Process Service** dialog appears as shown in Figure 59.
3. Drag a link from **File Sender** Service to the **File1** icon.
4. Drag a link from the **File Receiver** Service to **File2** icon.
5. Click the minimize button on the **Business Process** dialog to close the dialog.

Figure 59 Configure Binding



Configure the File Systems

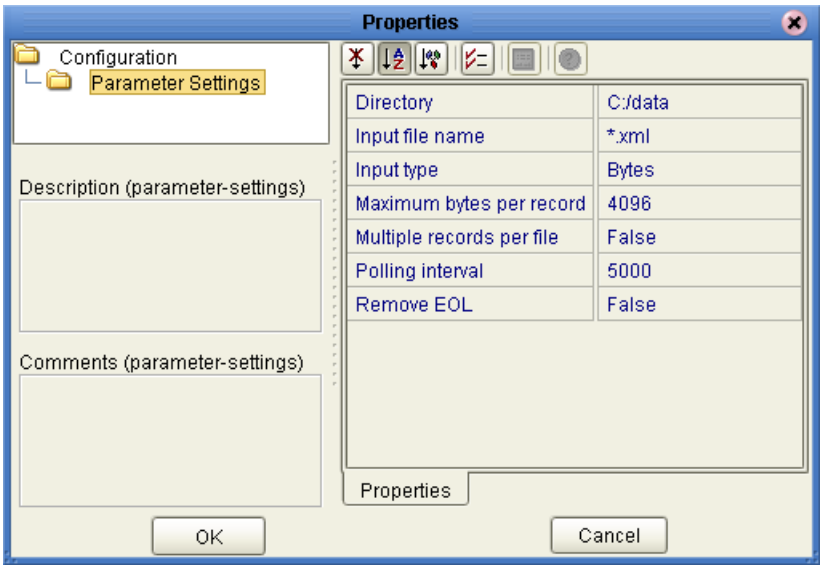
Configure the Inbound File eWay

1. Double-click the link to **File_input** to configure it.

   The **Templates** dialog appears.
2. Select Inbound **File eWay** and select **OK**.

   The **Properties** dialog appears, as shown in **Figure 60 on page 91**.
3. Change the **Directory** to **C:\data**.

Figure 60 Inbound File eWay



4 Change **Input** file name to ***.xml**.

5 Click **OK** to save changes.

Configure the Outbound File eWay

1 Double-click the link **File_output** to configure it.

2 Connect File Receiver to **File_output**.

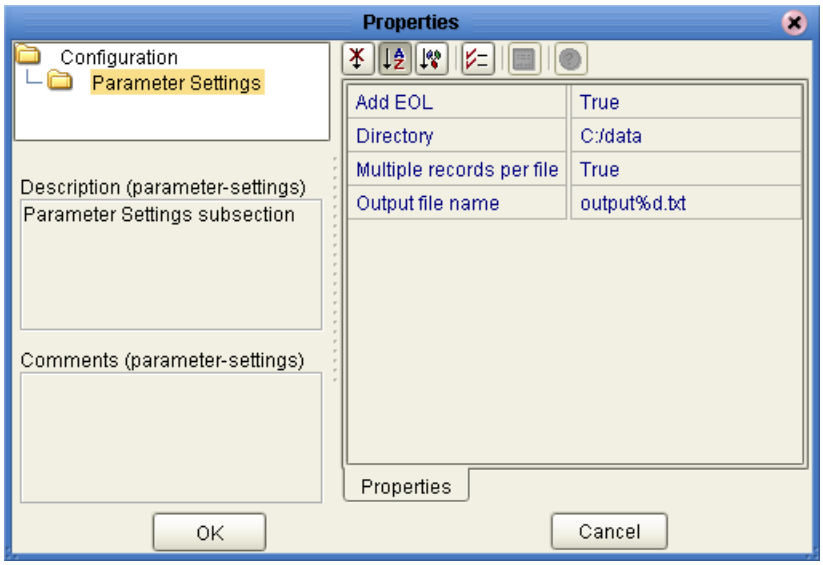
The **Templates** dialog appears.

3 Select **Outbound File eWay** and select **OK**.

The **Properties** dialog appears, as shown in **Figure 61 on page 92**.

4 Change the **Directory** to **C:\data**.

Figure 61 Outbound File eWay



5 Change the **Output** file name to **output%d.txt.**

6 Click **OK** to save changes.

8.3 Deploy and Test the Project

The final steps necessary to run your sample include:

- Creating and Configuring the Deployment Profile.
- Starting the Logical Host.
- Checking your output.

Note: *If you have imported the Payroll Sample Project, start here and continue to the end of this section, to complete your set-up.*

Deployment Profile

Create the Deployment Profile

1 Right-click your **Project** from the **Project Explorer**.

2 Select **New: Deployment Profile**.

3 The **Create Deployment Profile** dialog appears.

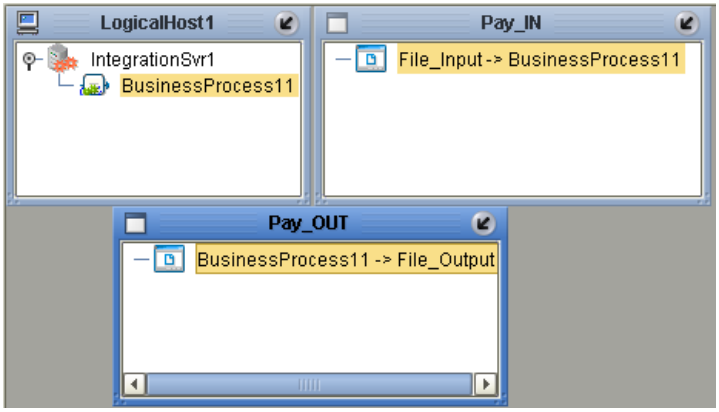
4 The **Deployment Profile** is called **Deployment1** by default. You can accept the default name.

5 Select an **Environment** (**Environment1**).

Configure the Deployment Profile

1. Drag **BusinessProcess11** from the middle pane to the **Integration Server** (**IntegrationSvr1**) located in the **LogicalHost** window.
2. Drag **File1 -> BusinessProcess11** from the middle pane to the **Pay_IN** window.
3. Drag **BusinessProcess11 -> File2** to the **Pay_OUT** window.
4. Click **Activate**.

Figure 62 Deployment Profile



5. Click **No**, so the change will not apply to the **Logical Host** immediately. (This is because your Logical Host is not running yet.)

Start the Logical Host

The Bootstrap process executes your **Payroll Project** and begins the process of polling your input data. The Bootstrap process is performed from a command prompt. Bootstrap will pick up the deployment profile the first time it runs; after that you would select reactivate and click **Yes** to apply the most recent changes to the Logical Host.

Note: *The Bootstrap command is case sensitive on Windows.*

To run the Bootstrap

1. Open a Windows command prompt as shown in the following Figure 63.
2. Navigate to where you installed the Logical Host; for example, **ican50\logicalhost\bootstrap\bin**, then type the following command:

```
CD \ican50\logicalhost\bootstrap\bin
```

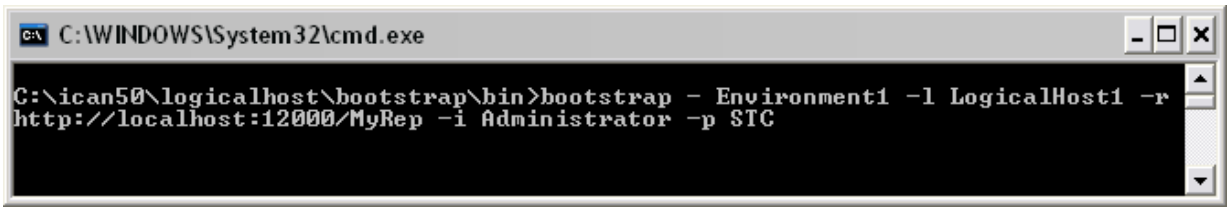
3. To start the Bootstrap process, type the following command:

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

- ***environment_name*** is the name of your environment (for example, *Environment1*),
- ***logicalhost_name*** is the name of your Logical Host (for example, *LogicalHost1*),

- ***repository_URL*** is the full URL of your Repository including the Repository name (for example, *http://localhost:12000/MyRep*),
- ***username*** is your user name
- ***password*** is your password

Figure 63 Bootstrap Command Example



Check output

1. Navigate to **C:\data** and check for an **output.txt** file.
2. Open the **output.txt** file and examine the data. It will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<Payroll>
    <FName>Vanessa</FName>
    <LName>Smith</LName>
    <Message>Eligible for Bonus</Message>
    <BonusTotal>$1500</BonusTotal>
</Payroll>
<?xml version="1.0" encoding="UTF-8"?>
<Payroll>
    <FName>Tonya</FName>
    <LName>Lee</LName>
    <Message>Not Eligible for Bonus</Message>
    <BonusTotal>$0.00</BonusTotal>
</Payroll>
```

8.4 About the Samples

This section will guide you through importing and deploying the sample projects.

8.4.1 Import the End to End Sample

This sample and the accompanying files can be found in the **eInsight_Sample.zip** file. You may download the sample file from the **Documentation** tab of the Enterprise Manager. It is one of the files available when you upload the eInsightDocs.sar file.

Download the eInsight Sample Project

1. Open the Enterprise Manager and click on the Documentation tab.
2. Select eInsight Business Process Manager from the Products list.
3. Select Download Sample and save to your local drive.

4 Unzip the compressed file and extract the files to another folder.

This compressed file contains the **Payroll_Project.zip** file as well as:

- Eligible.xml
- Ineligible.xml
- Readme.txt

Import the Sample Project

To run the project, you must import the **Payroll_Project.zip** file.

1 Right-click your Repository folder in the Enterprise Explorer and select Import.

The Import Manager dialog appears.

2 Click Browse and find **Payroll_Project.zip**.

3 Select the file and click Import.

4 Close the Import Manager dialog.

Once the import is complete, you can go directly to **"Step 5: Deploy and Test the System" on page 76** to run your sample.

8.4.2 Import the Correlation Sample

This sample and the accompanying files can be found in the e**Insight_Correlation_Sample.zip** file. You may download the sample file from the **Documentation** tab of the Enterprise Manager. It is one of the files available when you upload the eInsightDocs.sar file.

Download the eInsight Correlation Sample

1 Open the Enterprise Manager and click on the Documentation tab.

2 Select eInsight Business Process Manager from the Products list.

3 Select Download Sample and save to your local drive.

4 Unzip the compressed file and extract the files to another folder.

This compressed file contains the CorrelationProject.zip file as well as:

- input_corrReq-CPina.txt
- input_corrReq-KComella.txt
- input_corrRes-CPina.txt
- input_corrRes-KComella.txt
- output_corr1.dat
- Readme.txt

Import the Sample Project

To run the project, you must import the CorrelationProject.zip file.

1 Right-click your Repository folder in the Enterprise Explorer and select Import.

The Import Manager dialog appears.

2 Click Browse and find CorrelationProject.zip.

3 Select the file and click Import.

4 Close the Import Manager dialog.

Once the import is complete, continue to **"Deploy and Test the Project" on page 96** to run your sample.

8.4.3 Deploy and Test the Project

The final steps necessary to run your sample include:

- Creating and Configuring the Deployment Profile.
- Starting the Logical Host.
- Checking your output.

Note: *Check-out all components that are currently checked-in, so that you can make changes. Imported projects have several components checked-in by default.*

Deployment Profile

Create the Deployment Profile

1 Right-click your **Project** from the **Project Explorer**.

2 Select **New: Deployment Profile**.

3 The **Create Deployment Profile** dialog appears.

4 The **Deployment Profile** is called **Deployment1** by default. For this example, the default is used.

5 Select the **Environment** (**CorrEnv**).

Configure the Deployment Profile

1 Drag **bpMainProcess1** from the middle pane to the **Integration Server** (**IntegrationSvr1**) located in the **LogicalHost** window.

2 Drag **bpSubProcess1** from the middle pane to the **Integration Server** (**IntegrationSvr1**) located in the **LogicalHost** window.

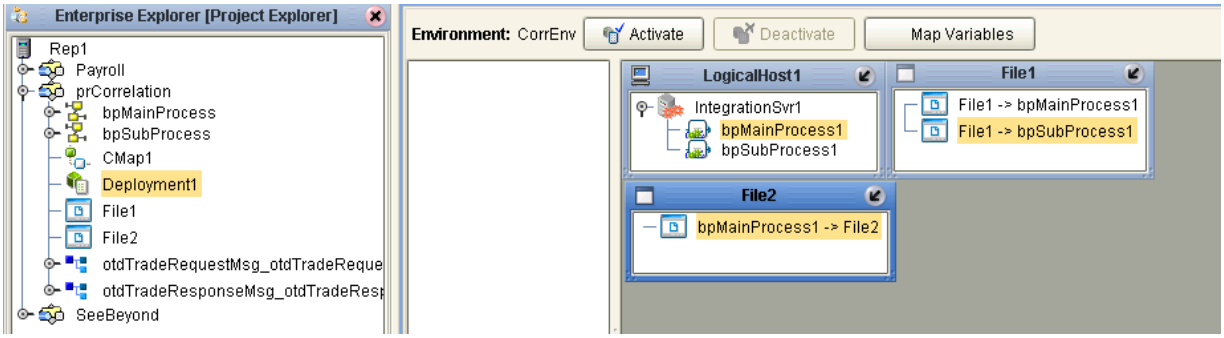
3 Drag **File1 -> bpMainProcess1** from the middle pane to the **File1** window.

4 Drag **File1 -> bpSubProcess1** from the middle pane to the **File1** window.

5 Drag **bpMainProcess1 -> File2** to the **File2** window.

6 Click **Activate**.

Figure 64 Deployment Profile



7 Click **No**, so the change will not apply to the **Logical Host** immediately. (This is because your Logical Host is not running yet.)

Start the Logical Host

The Bootstrap process executes your **project** and begins the process of polling your input data. The Bootstrap process is performed from a command prompt. Bootstrap will pick up the deployment profile the first time it runs; after that you would select reactivate and click **Yes** to apply the most recent changes to the Logical Host.

Note: *The Bootstrap command is case sensitive on Windows.*

To run the Bootstrap

1 Open a Windows command prompt as shown in the following Figure 63.

2 Navigate to where you installed the Logical Host; for example, **ican50\logicalhost\bootstrap\bin**, then type the following command:

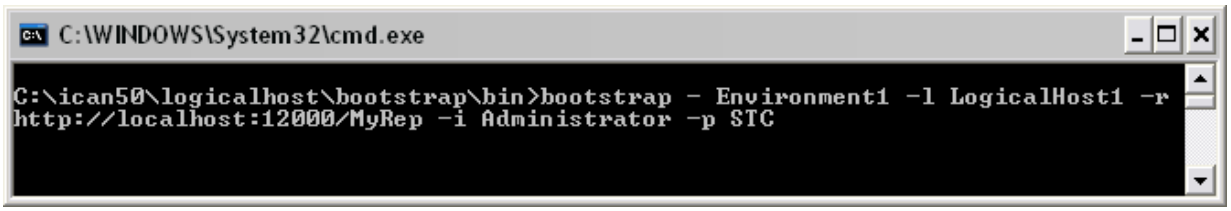
```
CD \ican50\logicalhost\bootstrap\bin
```

3 To start the Bootstrap process, type the following command:

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

- ***environment_name*** is the name of your environment (for this example, *CorrEnv*),
- ***logicalhost_name*** is the name of your Logical Host (for this example, *LogicalHost1*),
- ***repository_URL*** is the full URL of your Repository including the Repository name (for example, *http://localhost:12000/MyRep*),
- ***username*** is your user name
- ***password*** is your password

Figure 65 Bootstrap Command Example



Check output

1. Navigate to **C:\data** and check for an **output.dat** file.
2. Open the **output.dat** file and examine the data. It will look like this:

```
Waiting on response for<otdTradeRequestMsg>
    <Trader>CPina</Trader>
</otdTradeRequestMsg>

Waiting on response for<otdTradeRequestMsg>
    <Trader>KComella</Trader>
</otdTradeRequestMsg>

Correlated request for trader KComella with response: <?xml
version="1.0" encoding="UTF-8"?>
<otdTradeResponseMsg>
    <Trader>KComella</Trader>
    <Action>Sell</Action>
    <Quantity>10</Quantity>
    <StockSymbol>MSFT</StockSymbol>
</otdTradeResponseMsg>
Correlated request for trader CPina with response: <?xml
version="1.0" encoding="UTF-8"?>
<otdTradeResponseMsg>
    <Trader>CPina</Trader>
    <Action>Buy</Action>
    <Quantity>2000</Quantity>
    <StockSymbol>SBYN</StockSymbol>
</otdTradeResponseMsg>
```

Chapter 9

Workflow Solutions With the User Activity

This chapter covers the User Activity and its role in the business process workflow. The case studies in this chapter are designed to illustrate the functionality of the User Activity, in addition to showing an example of the Worklist Manager and Worklist Viewer.

9.1 Overview

This chapter provides three ways to learn about implementing a project. Depending on your needs, you can:

- Create the Worklist Manager sample from scratch: **"Case Study Overview" on page 109**
- Import the Worklist Manager sample and run it: **"Import the Worklist Manager Sample" on page 129**
- Import and run the User Activity sample that demonstrates the correlation feature: **"Import the User Activity Sample" on page 129**.

9.2 Overview

The User Activity allows users to participate in business processes. eInsight ESB supports the definition of organization hierarchies and user roles for task assignment. Tasks can be escalated and delegated by users from custom Worklists and activity processing windows. Working with SeeBeyond eVision Studio, eInsight ESB can create tailored workflow tasks with processing pages that are tailored specifically for each task. See **Case Study: Audit Processing** on page 110 for a complete Task Assignment example.

1. Create the Business Process and add a User Activity.
2. Set up a repository of users, organizational structures and roles in LDAP. (See **Organizational Roles** on page 104.)
3. Set up user assignments. See **Configure Task Assignment** on page 106.
4. Use the Worklist Manager to **Manage Tasks** on page 108.

See **"Case Study: Audit Processing" on page 110** for an end to end User Activity exercise.

9.2.1 User Activity Configuration

This section provides an overview of how to configure the User Activity. This option uses OpenLDAP to determine who belongs to the organizational structure.

Configure a User Activity

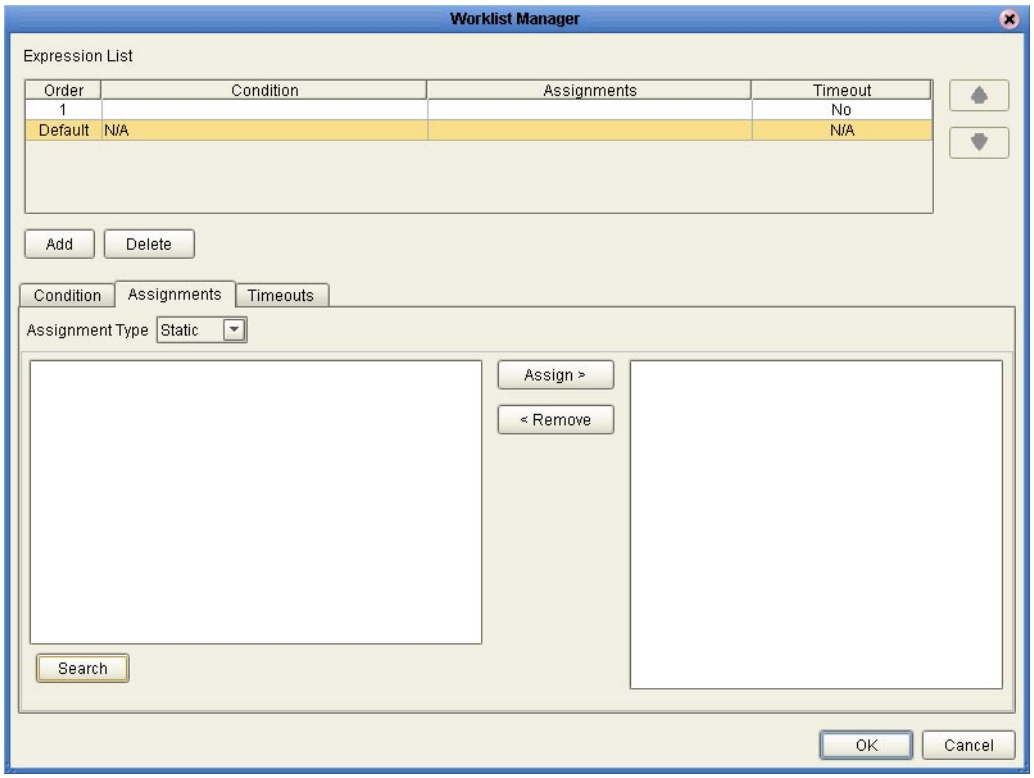
The basic steps to configure the User Activity are as follows:

1. Create a business process model, (see **"Building a Business Process Model" on page 23**).
2. Configure your database for persistence and run the Database Scripts, (see **"Persistence and Monitoring" on page 62**).
3. Add a User Activity as part of your business model. See **"Configure a User Activity" on page 100**.
4. Create an **eVision Studio PageFlow** (see the *eVision Studio User's Guide* for details).
5. Drag the **eVision PageFlow** to the User Activity, see **"Create the eVision Pages" on page 115**.

Figure 66 Configured User Activity

6. Add a Worklist Viewer to your Environment, see **"Create a New Project and Environment" on page 113**.
7. Add Flex Attributes (optional). See **Flex Attributes** on page 102 for details.

Figure 67 Worklist Manager Settings



8 Run Task Assignment, see **"Task Assignment" on page 106**.

Configure a User Activity inside a While Loop

When creating a business process that includes a User Activity inside of a While Loop (see **"While" on page 30**), you must take a few extra steps in the configuration, as outlined below.

1 Configure the User Activity as described above (**"Configure a User Activity" on page 100**).

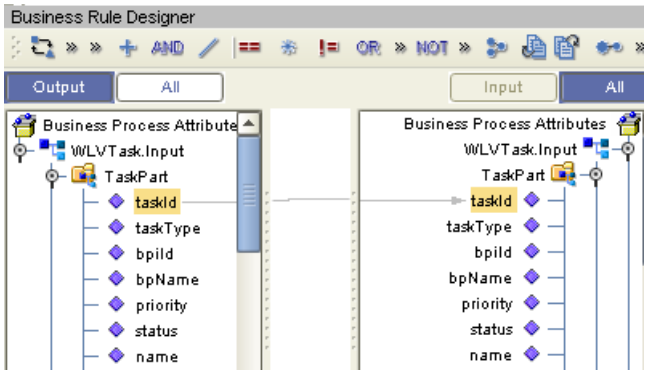
2 Add a business rule to the link leaving the User Activity.

3 Open the Business Rule Designer and expand the **WLVTask.Input** node.

4 Copy a business process attribute from the left pane to the right pane, as shown in Figure 68.

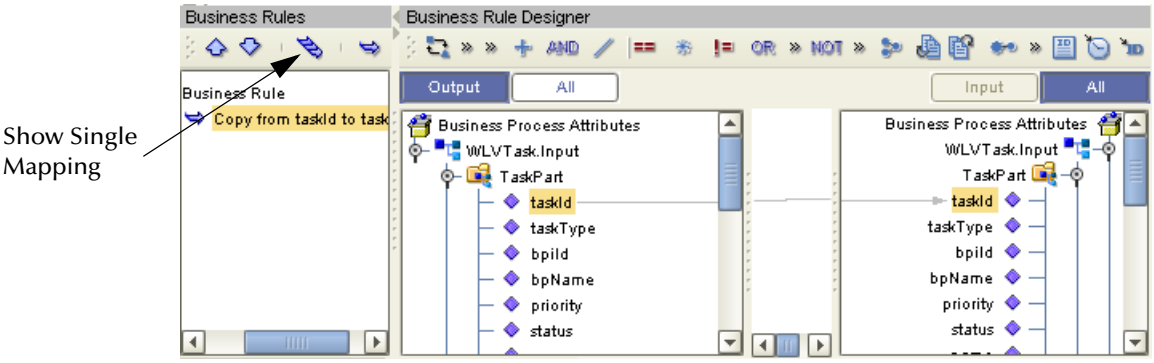
The purpose of this step is to create an output container.

Figure 68 Copy Business Process Attribute



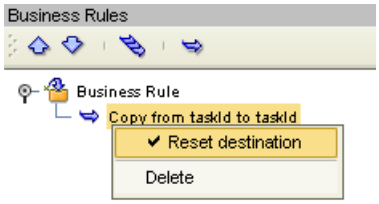
5 Open the Business Rules pane and click **Show Single Mapping**, as shown in Figure 69.

Figure 69 Business Rules



6 Right-click the copy rule and select **Reset Destination**, as shown in Figure 70.

Figure 70 Reset Destination



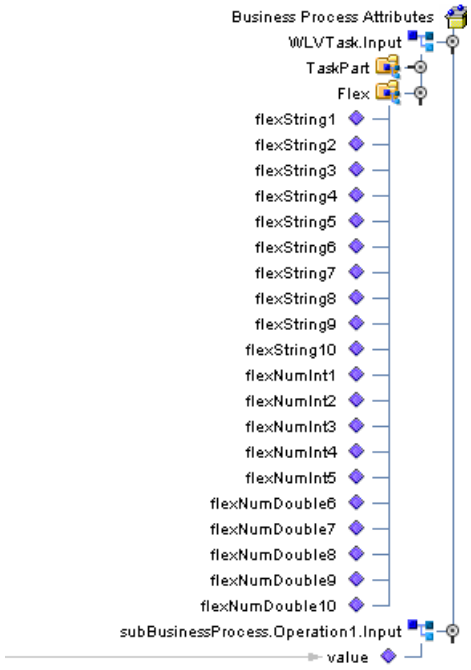
7 Save the business process.

Flex Attributes

Flex attributes are customizable attributes for use with Task Assignment. The attributes appear in the Business Rule Designer (as shown in **Figure 71 on page 103**)and also, as columns in the Worklist Manager.

You can map values to these attributes in the Business Rule Designer, so that the values appear in the Worklist Manager. You can also label the attributes so they are easily identifiable in the Worklist Manager.

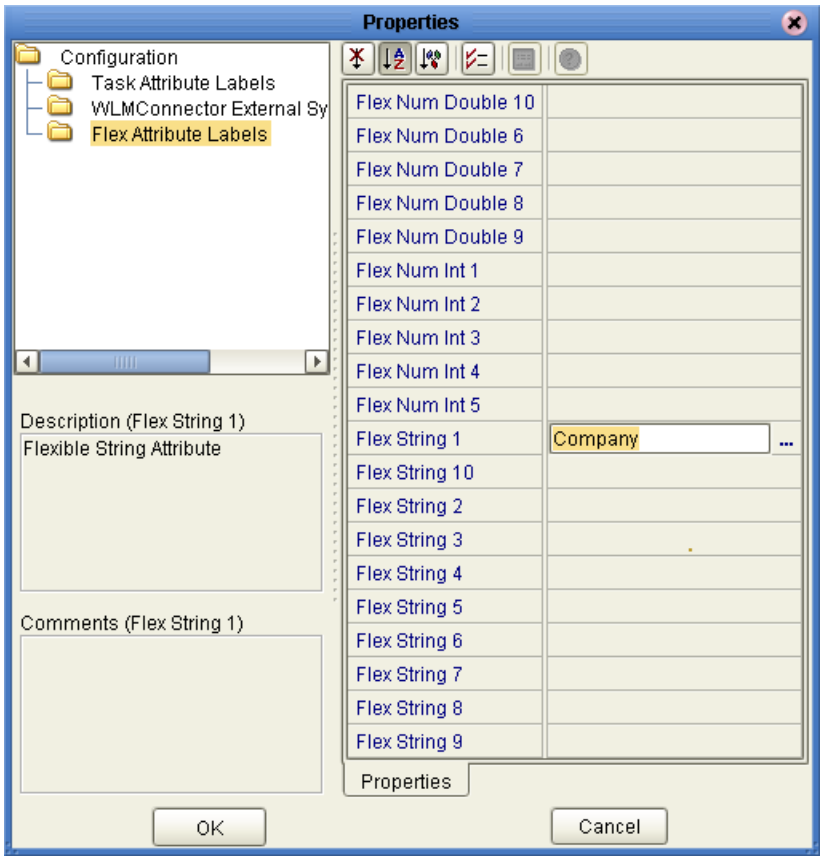
Figure 71 Flex Attributes in the Business Rule Designer



Customize Flex Attribute Labels

1. From the Environment Explorer tab, right-click the **WLV** component and select **Properties**.
2. From the **WLV Configuration** options, select **Flex Attribute Labels**, as shown in **Figure 72 on page 104**.

Figure 72 Flex Attributes



3 Create labels for as many attributes as necessary.

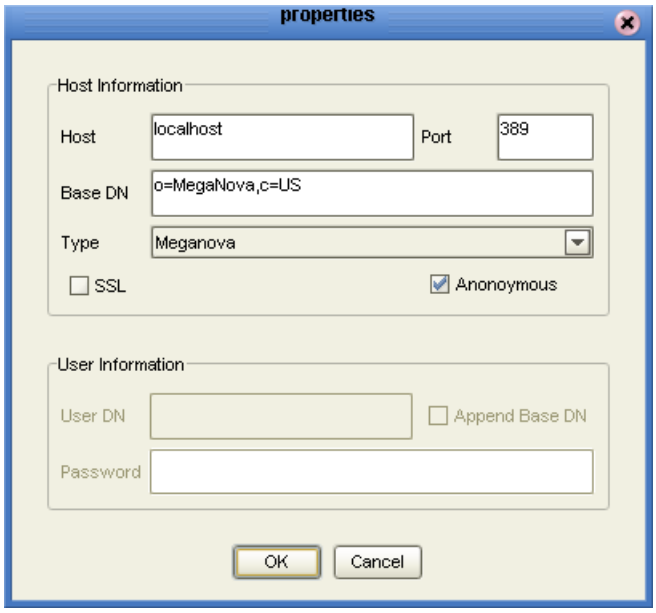
4 Click **OK** to save changes.

Organizational Roles

Organizational roles help define processes based on a person's position or title. By entering information about the structure of your organization, you can make processes easier to manage.

Support for LDAP means that you can use repositories of users, hierarchical organizational structures, and roles. An LDAP based applications can be used to populate the Worklist Manager with members of your organization and their organizational role. You assign rights based on organizational role assignments.

Figure 73 LDAP Connection Settings



For example, you may allow managers to view their subordinate's activity list (also called a **Worklist**) and the ability to reassign tasks in that list. As a manager, when you login to your Worklist, you see your tasks and the tasks of your direct reports.

Note: *See your third party software vendor's manual for information about setting up your organization's information in LDAP.*

User Activity in a UNIX Environment

If your ICAN environment includes a Logical Host on UNIX, you must configure your LDAP Provider URL to connect to your LDAP server.

Example of ICAN environment:

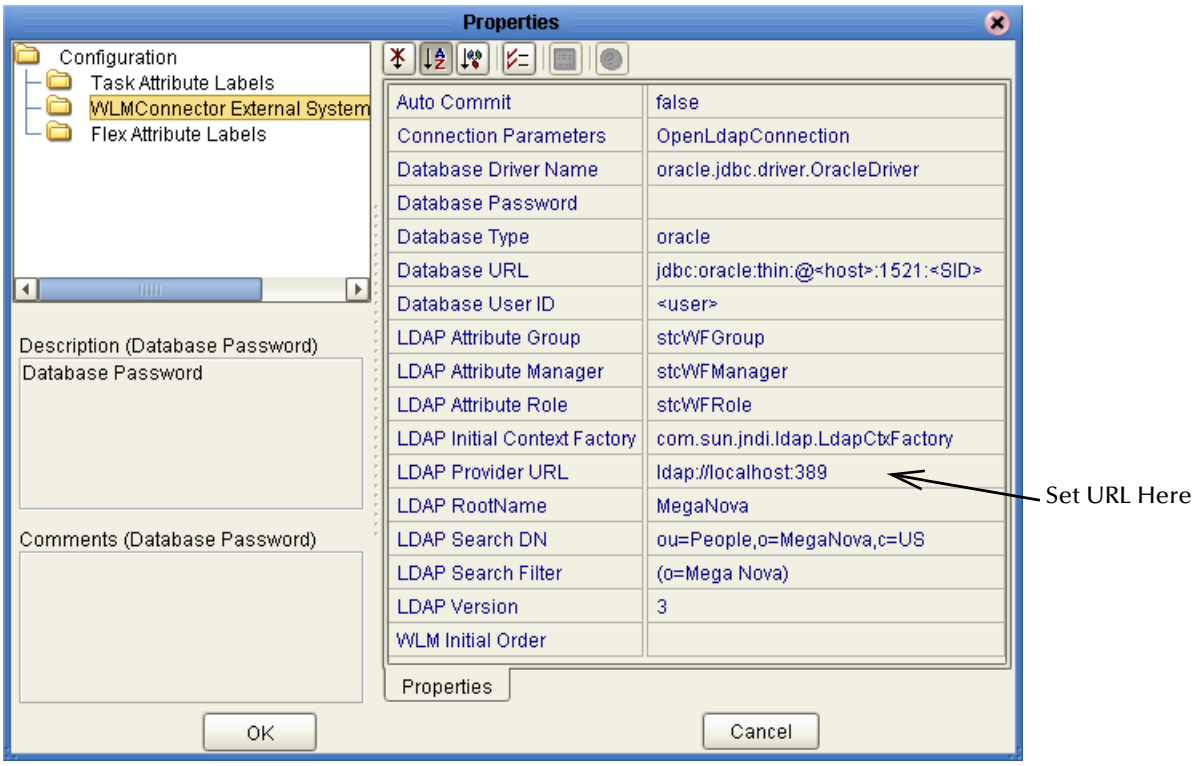
- LogicalHost - UNIX
- SeeBeyond Repository - Windows
- LDAP - Windows

In this environment, the LDAP Provider URL, in the WLM property sheet needs to be set to an exact URL.

Set LDAP Provider URL

1. From the Environment Explorer tab, right-click the **WLV** component and select **Properties**.
2. From the **WLV Configuration** options, select **WLM Connector External System**, as shown in **Figure 74 on page 106**.

Figure 74 WLM Connector Settings



3 Enter the exact URL to your LDAP server in the LDAP Provider URL field.

4 Click **OK** to save changes.

9.2.2 Task Assignment

Task assignment allows you to set up and view tasks, depending on your organizational role. With certain management level rights, you may assign a person who will receive a task, if the activity fails. See **Case Study: Audit Processing** on page 110 for a complete Task Assignment example.

Configure Task Assignment

1 Double-click the **User Activity**.

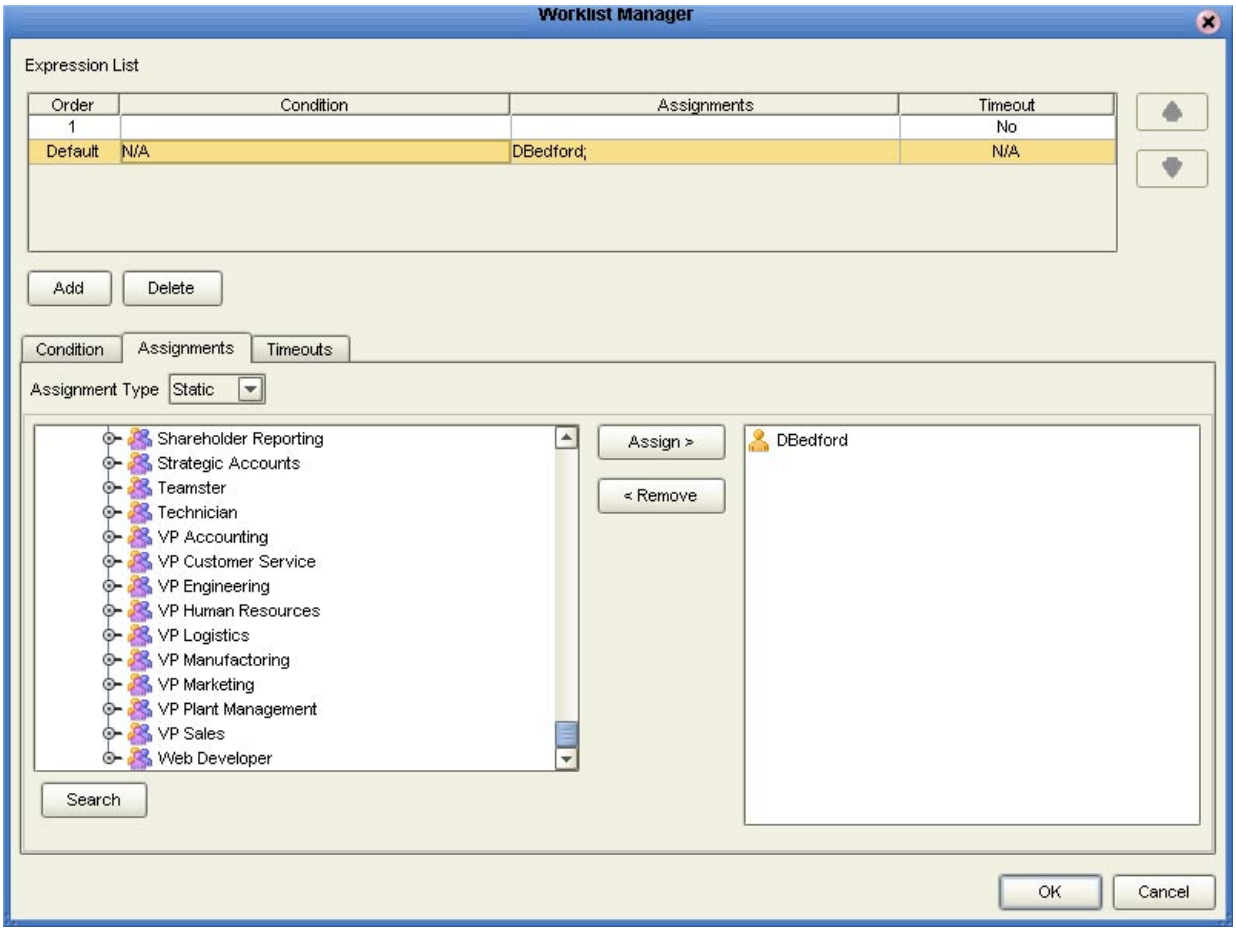
The Worklist Manager dialog displays (see Figure 75).

2 Click **Search** to find an LDAP source, as shown in Figure 73.

The user list displays (See Figure 75).

3 Click **OK** to return to the Worklist Manager dialog.

Figure 75 Worklist Manager: Assignments Tab



4 Navigate the Roles or Groups list to find your assignees**.**

5 Select individuals from the list and click **Assign**.

6 Select the **Condition** tab to enter an expression.

You can create expressions in the Business Rule environment to evaluate an activity for user completion.

7 You may also:

- Add another rule by choosing **Add**.
- Delete an assignment from this page by selecting an expression and choosing **Delete**.

8 Click **OK** when you have completed the configuration.

Using the Worklist Manager

Login to the Worklist Manager to view your list of tasks. You will see your own tasks as well as any subordinates assigned to you (if applicable). You can manage your tasks and/or the tasks of your subordinates from this view. See Figure 77 for an example of a

manager's Worklist view. See **Workflow Solutions With the User Activity** on page 99 for a complete Task Assignment example.

Manage Tasks

From the Worklist Manager, you can manage your tasks and/or the tasks of your subordinates. You can access the Worklist Manager with a web browser by entering:

```
http://<localhost>:<port>/wlm
```

- <localhost>: The system where your Repository is running.
- <port>: The port number to access your Repository.

Figure 76 Worklist Manager Login

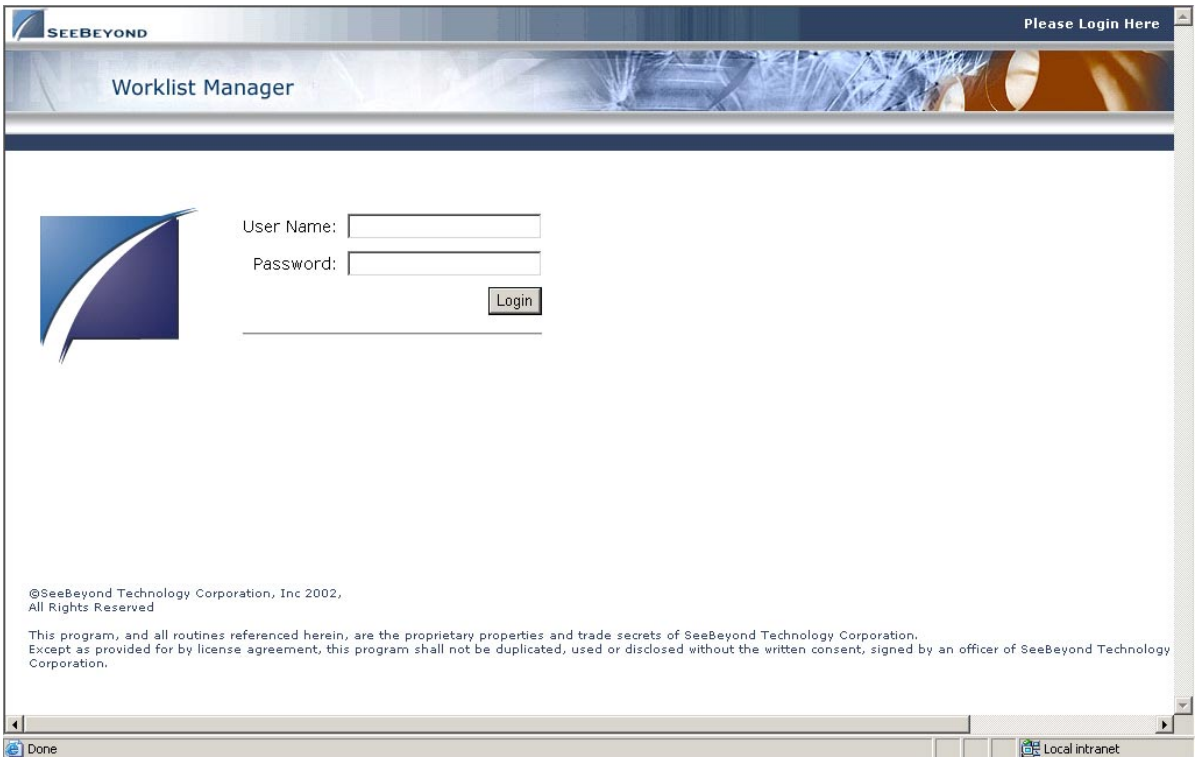
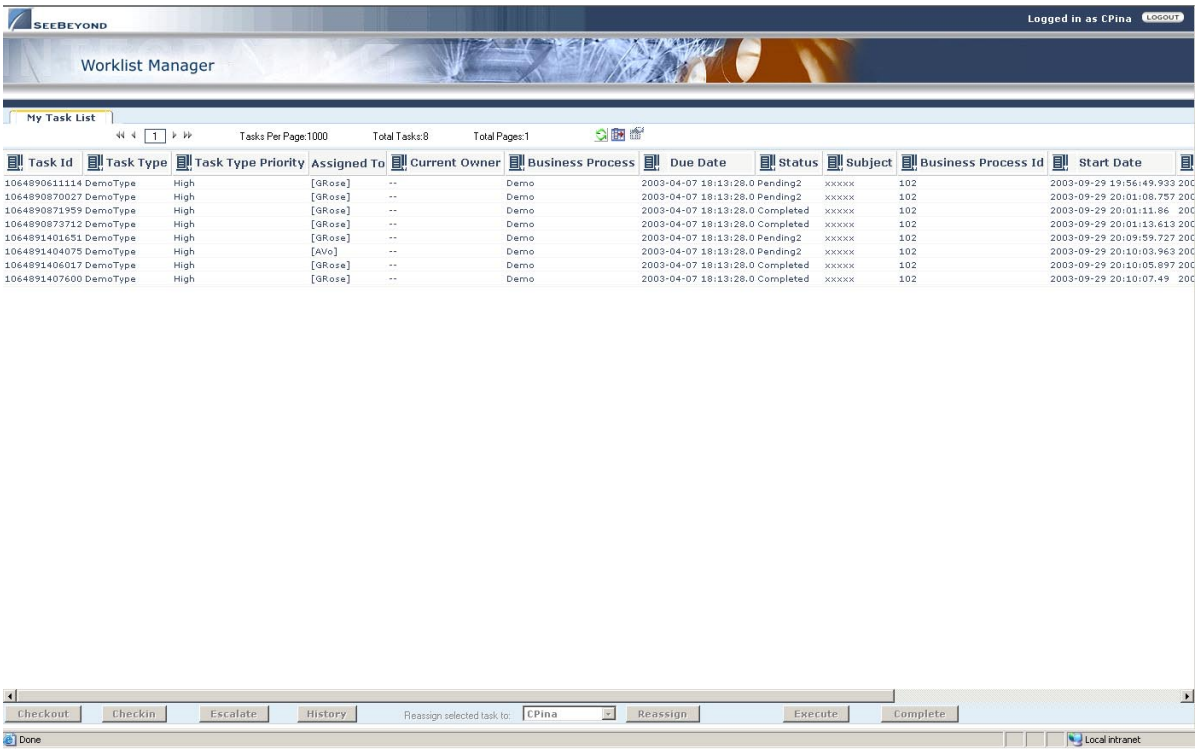


Figure 77 Worklist Manager: Task View



Here is an explanation of the options you will see:

- **Checkout/Checkin** is necessary to ensure that more than one person is not working on the same task. This is the first thing you must do before you can make any changes to a task.
- **Escalate** sends the task to the user's manager.
- **History** provides a record of all past information about the task.
- **Reassign** allows you to pass a task to another team member. Click **Reassign** and select a name from the drop-down list. This option is available for manager's only.
- **Execute** opens the task so that you can perform the task.
- **Complete** commits the changes. You must select **Complete** before leaving the Worklist Manager or your changes will be lost.

Note: *See* **Case Study: Audit Processing** *on page 110 for a complete Task Assignment example.*

9.3 Case Study Overview

Implementing a User Activity is the process of translating the vision of the business user into a functioning task assignment system. The User Activity must be configured but some may require modification.

Create the Input File

The first step in this implementation requires that you create the input file for the system. In an actual implementation, your input may come from an external system.

Create a New Project and Environment

In the second step of this implementation, you will create a new Project where your business process and eVision Pages will reside and a new Environment for your Project.

Create the eVision Pages and Business Process Models

In this step, you will create a new business process and a subprocess with a user activity that links to the task management system. You will also create the eVision Pages that act as the user interface to the system.

Create a Connectivity Map and Deployment Profile

When you create the Connectivity Map, you are making the connections between the system components and the external systems. You will also create a Deployment Profile that you will activate when you put the system into production.

Deploy and Test the System

To run the system, you must invoke the Bootstrap. This action will pick up your Deployment Profile and execute your task assignment system. Once the system processes your input files and data entered, an output file is created. To verify that this implementation has completed properly, check the output file.

9.4 Case Study: Audit Processing

The case study discussed in this chapter illustrates a simplified implementation of an auditing system. In this case, eInsight ESB receives data as a text file as well as user input from an eVision page.

Once the system receives the data, a task appears in the Worklist Manager for the assigned user(s). The user has the option to complete the task or escalate the task to a manager. If the user is a manager, he/she also has the option to reassign the task to a subordinate.

9.4.1 Before You Begin

To complete this exercise, you need to have the following:

- ICAN 5.0 products installed:
  - eGate Integrator
  - eInsight Enterprise Service Bus

  - eVision Studio
  - File eWay
- An Oracle database.
- An LDAP directory server. (For this example, **OpenLDAP** is used with the **MegaNova** database).
- A directory on your local drive named **data**.

LDAP Configuration

This sample uses the OpenLDAP application as the LDAP directory, with custom data (MegaNova). You will need to follow the instructions for your LDAP application to add data to the LDAP database. You must create your own directory data and data file. You can use the names suggested in our example, or substitute your own. If your directory structure is not the same, the sample may not run as written.

1 Install an LDAP application.

For this example, OpenLDAP is used. At publication, this application is available from:

```
http://www.openldap.org
```

2 Install the binary file or compile your own. Install or uncompress it to:

```
c:\ldap
```

3 After installation, create a configuration file for your LDAP installation. See your LDAP documentation for details.

This configuration file is used in our sample:

Figure 78 Example: slapd.conf

```
include  c:/ldap/schema/core.schema
include  c:/ldap/schema/cosine.schema
include  c:/ldap/schema/inetorgperson.schema
include  c:/ldap/schema/stc.schema

pidfile  c:/ldap/slapd.pid
argsfile c:/ldap/slapd.args

databaseldbm
suffix   "o=MegaNova,c=US"
rootdn   "cn=Manager,o=MegaNova,c=US"
rootpw   secret
directoryc:/ldap/MegaNovaDB
indexobjectClasseq
```

4 Add the schema definition to your schema folder:

Example: c:/ldap/schema/stc.schema

```
# schema/stc.schema
#
# Auxiliary Object Class to be used with Person (or subtypes of it)
#
#  Roles allows a user to be cross functional
```

```
attributetype (1.3.6.1.4.1.1351.666.1.1
    NAME 'stcWFRole'
    DESC 'STC WorkFlow User Roles' SUP organizationalUnitName )

#  Group provide department level groupings
attributetype (1.3.6.1.4.1.1351.666.1.2
    NAME 'stcWFGroup'
    DESC 'STC WorkFlow User Group' SUP organizationalUnitName )

#  Manager shows hierarchy in the organization
attributetype (1.3.6.1.4.1.1351.666.1.3
    NAME 'stcWFManager'
    DESC 'STC WorkFlow User Manager'
    EQUALITY distinguishedNameMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE )

# stcWorkFlowPerson
# The stcWorkFlowPerson represents people who are associated with an
# organization in both a department and a role.  It is an auxiliary
class
objectclass( 1.3.6.1.4.1.1351.666.1
    NAME 'stcWorkFlowPerson'
    DESC 'STC Work Flow Assignment Person'
    AUXILIARY
    MAY (
        stcWFRole $ stcWFGroup $ stcWFManager )
    )
```

5 Start the LDAP server from the command line.

Example: `c:\ldap> slapd.exe`

6 Create or add the sample data (.ldif file) to the LDAP server.

Example: `c:\ldap>  ldapmodify -a -v -D cn=Manager,o=MegaNova,c=US -h localhost -P 2 -x -w secret -f MegaNova.ldif`

You can manually modify the .ldif file or use an LDAP software utility, such as the OpenLDAP browser Softerra LDAP Browser from:

`http://www.softerra.com/products/products.php`

Input File

The sample system you are creating requires input information. For this exercise, you will create an input file: **input1.txt**. The file that you create here, contains the data that the system receives and changes to create your final output.

Create the Input File

1 Create a text file with your name, for example:

- **input1.txt:**

```
Mary Smith
```

2 Save the file to **c:\data**.

9.4.2 ICAN Configuration

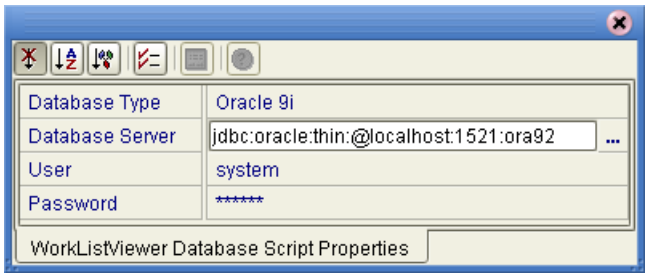
Run Worklist Manager Database Scripts

1 From the Enterprise Explorer, expand the SeeBeyond folder and the eInsight ESB folder.

2 Right-click the **Worklist Viewer** and select **Checkout**.

3 Right-click the **Database Scripts** folder and select **Properties**, as shown in Figure 79.

Figure 79 Worklist Viewer Database Properties



4 Configure the database properties to connect to your database.

Note: *The Oracle user must have DBA privileges to create the new wlm user*

5 Right-click on the **Oracle Install Scripts** and select **Run**.

A View your database to verify that the tablespace **wlm_data** exists.

B View your database to verify that the user **wlm** is defined.

9.4.3 Create a New Project and Environment

Create a new Project

1 Launch the Enterprise Designer.

2 Right-click your Repository and select **New Project.**

A new Project appears in your Project Explorer tree structure.

3 Rename the Project to **wlmProject**.

4 Click the **Save All** toolbar button to save your changes.

Create a new Environment

1 Select the **Environment Explorer** tab from the Enterprise Designer.

2 Right-click on your **Repository** and select **New Environment**.

3 Right-click on your **Environment** and select **New Logical Host**.

4 Right-click on your **Environment** and select **New File External System**.

The system prompts you to name the **File External System**.

5 Enter **File_IN** as the name of your **File External System**.

6 Select **Inbound File eWay** as the **External System Type**.

7 Repeat step 4 and name the **File External System**: **File_OUT**.

8 Select **Outbound File eWay** as the **External System Type**.

9 Right-click **Logical Host** and select **New SeeBeyond Integration Server**.

Create a Worklist Viewer and eVision External System

1 From the Environment Explorer create a **New Worklist Viewer** and name it **myWLV**.

2 Right-click on the **myWLV** and select **Properties**.

3 Select the **WLM Connector External System Configuration** and configure the following options:

A Database Password: **wlm**

B Database URL: enter your Oracle settings

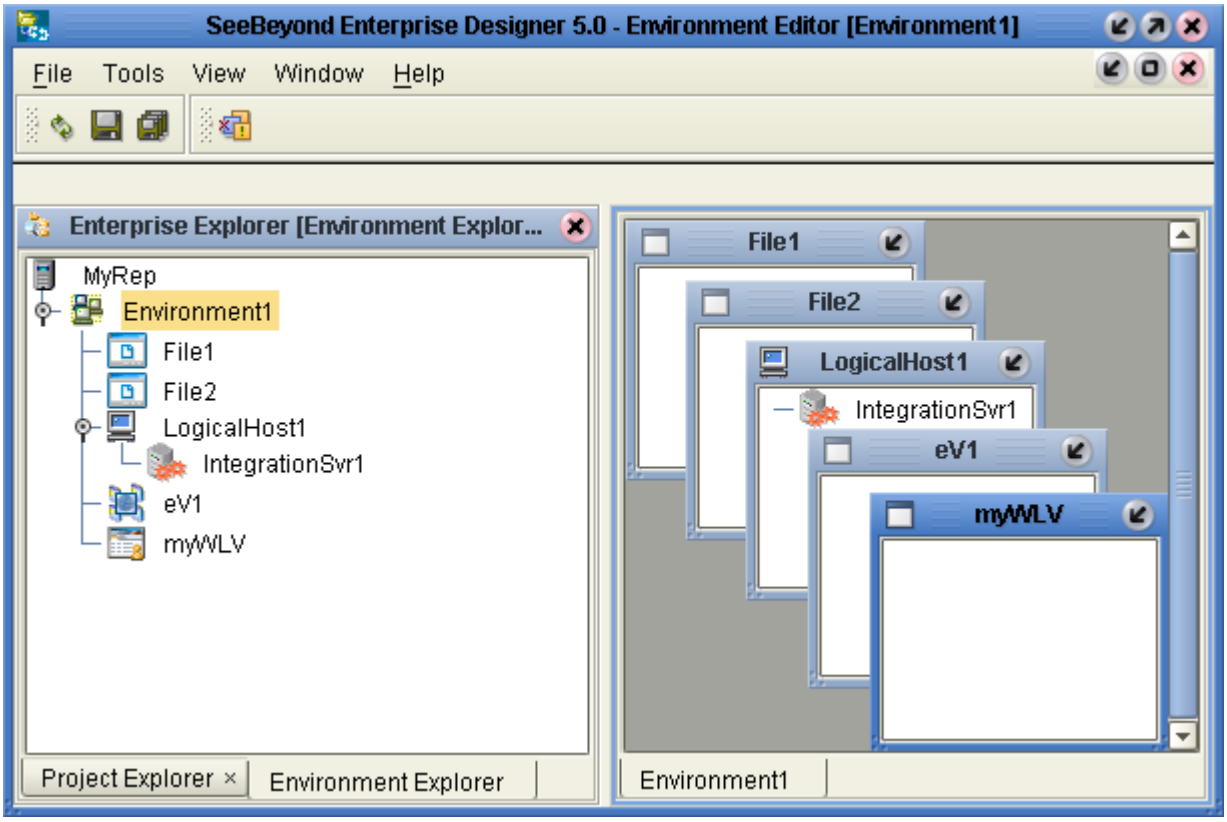
C Database User ID: **wlm**

4 Add an **eVision External System** and name it **eV1**.

5 Select **OK**.

Your new Environment will look like Figure 80.

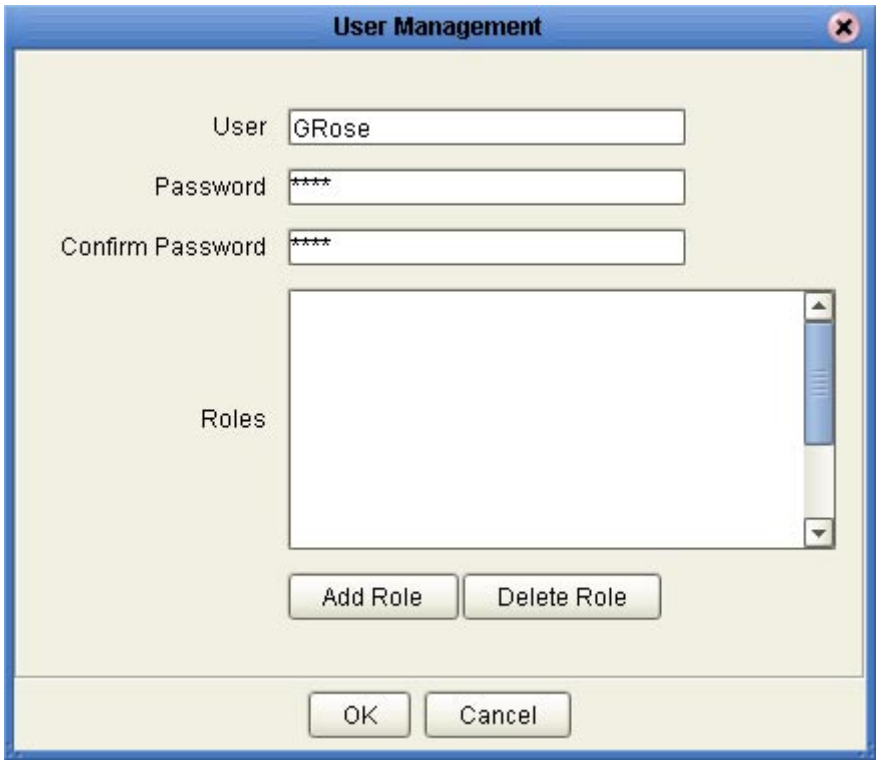
Figure 80 New Environment



Add the LDAP users to your environment

1. Right-click on your Environment and select User Management.
2. Add the following users to the Worklist Manager:
   - GRose
   - CPina
   - KComella
3. Fill in the Password information and Add a Role for each user, as follows:
   - Password: **pass**
   - Role: **all**

Figure 81 User Management



In the LDAP hierarchy for this example, the order from manager to subordinate is:

- KComella - Senior Manager
  - CPina - Manager
    - GRose - User

9.4.4 Create the eVision Pages

1. Right-click on your Project (**wlmProject**) and select New Page Layout.
2. Create a new Blank Page and name it **auditPage**.

A Select the **html text** element from the Page tools and place it on the top center of the eVision page.

B Replace the default text with **Audit Info**.

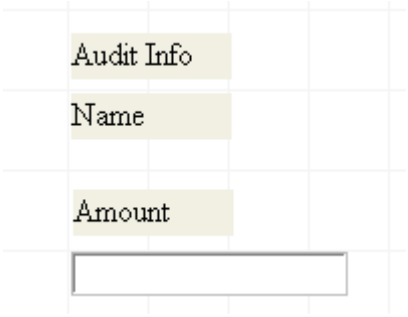
C Select another **html text** element and place it on the page, under and to the left of Audit Info.

D Enter **Name** for the default text.

E Select a **textbox** element and place it on the eVision page, to the right of the Name label.

F Place a **Submit Button** under the **textbox**, as shown in Figure 82.

Figure 82 auditPage Layout



3 Create a new Blank Page and name it **exitPage**.

A Select an **html link** element and place it near the top and center on the eVision page.

B Enter **Close Window** as the default text.

C Enter **javascript: void window.close()** in the property for the **href field**.

4 Click **OK** to close the properties page.

9.4.5 Create the Business Process Models

This section contains detailed instructions to build your model.

Create the Sub-Process

1 Click the **Project Explorer** tab and right-click on your **wlmProject**.

2 Select **New**: **Business Process**.

A new Business Process appears in your directory tree under your Project and a blank Business Process appears in the Business Process Designer (right pane).

3 Rename the business process to **subBusinessProcess**.

4 Right-click on the business process and select the **Open Property Sheet**.

5 Click on the **Business Process Attribute** tab.

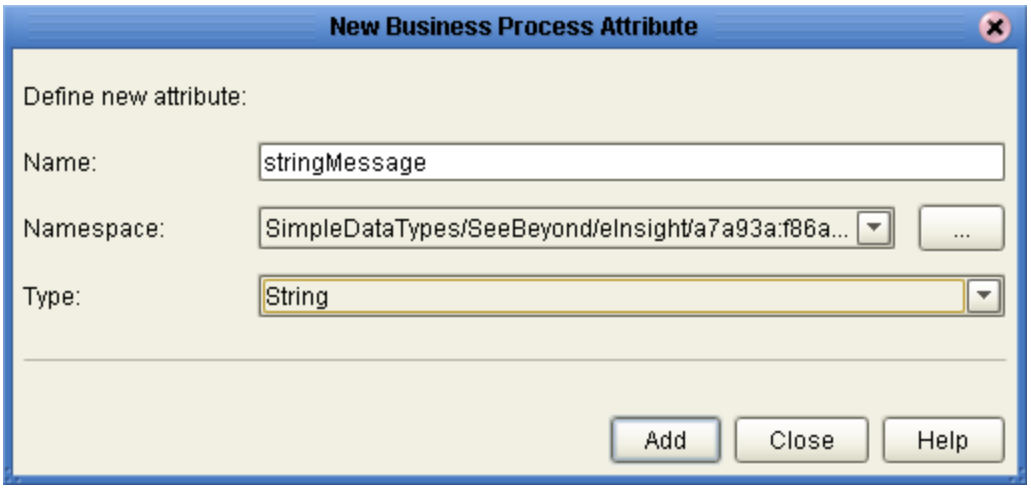
6 Click **Create** to add a new Business Process Attribute.

A Name the new Business Process Attribute: **stringMessage**.

B Select **SimpleDataTypes** for the namespace

C Select **String** for the type, as shown in Figure 83.

Figure 83 New Business Process Attribute



Create the WSDL file

1 Click on the WSDL tab and click **Create**.

2 Click **porttype** and **operation.**

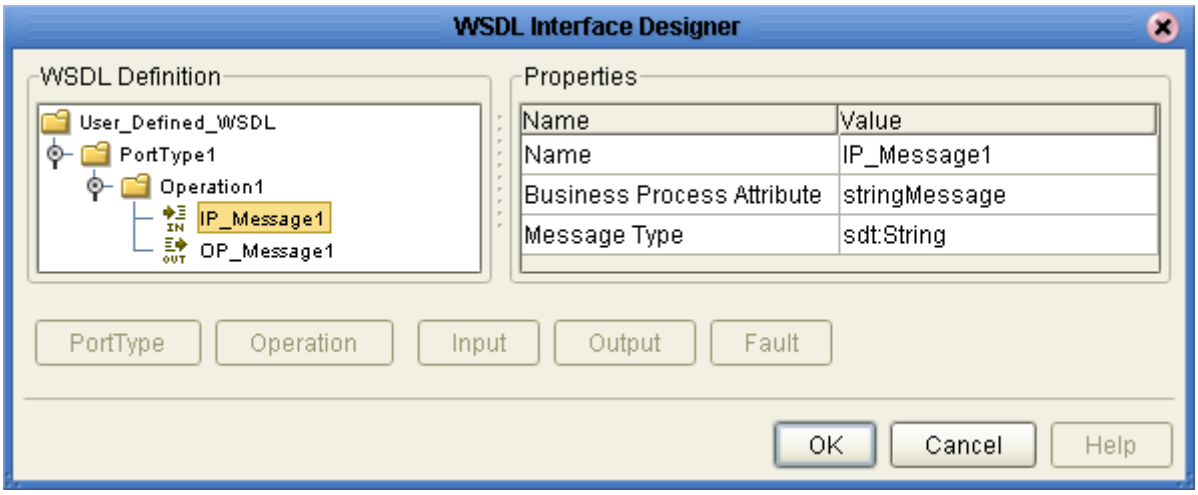
3 Click **input operation**.

4 Go up one level to create an **output operation.**

5 For **IP_Message1**, select **stringMessage** for the Business Process Attribute.

6 For **OP_Message1**, select **stringMessage** for the Business Process Attribute.

Figure 84 New WSDL File



7 Select **OK**.

Create a Partner

1 Click on the **Partners** tab.

2 Create a new partner called **wlmpartner**.

Create the SubProcess Model

1 Select a Receive Activity from the eInsight ESB toolbar and place it on the Business Process Designer (to the right of the Start Activity).

2 Select a Reply Activity from the eInsight ESB toolbar and place it on the Business Process Designer (to the left of the End Activity).

3 Select the Show operation from the **auditPage** and place it on the Business Process Designer, between the Receive and Reply Activities.

4 Select the Show operation from the **exitPage** and place it on the Business Process Designer, to the right of the auditPage.

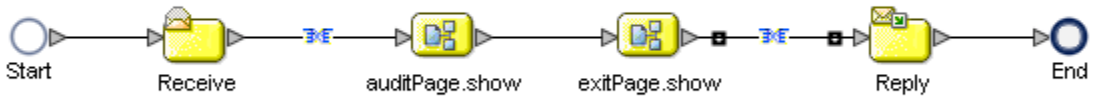
5 Link the activities:

A Move your mouse over the **Start** activity until a hand appears.

B Click and hold your mouse to drag a **Link** between the elements.

6 Repeat steps 5A and 5B to connect the entire model.

Figure 85 subBusiness Process

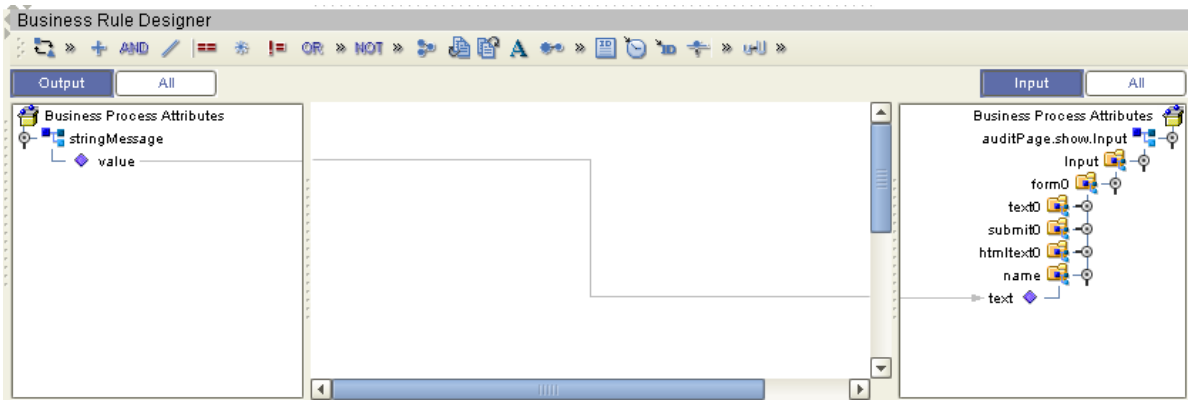


Add Business Rules to the Links

1 Right-click on the link between the **Receive Activity** and the **auditPage** and select **Add a Business Rule**.

2 Map the **stringMessage** value node to the text of the **Name** element in the input of the **auditPage**, as shown in Figure 86.

Figure 86 Add Business Rule to Link



3 Right-click on the link between the **exitPage** and the **Reply Activity** and select **Add a Business Rule**.

A Place a **Concat** Method on the **Business Rule Designer** from the Method Palette.

B Map a link from the left pane's **Receive Activity:stringMessage** to **String1** of the **Concat** Method Box.

C Place a **String Literal** Method on the **Business Rule Designer** and enter "**:**".

D Map a link from the **String Literal** to **String2** of the **Concat** Method Box.

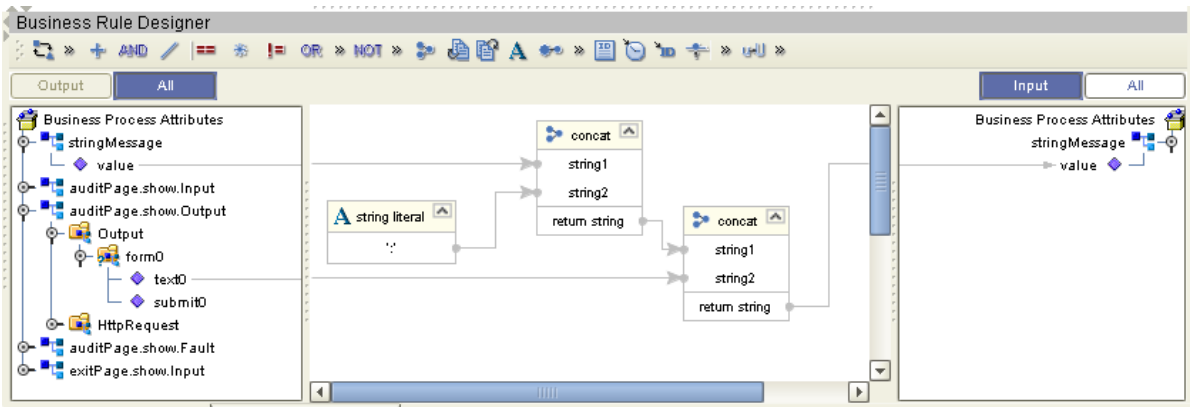
E Place another **Concat** Method on the **Business Rule Designer**.

F Map a link from the first **Concat** Method Box's Return String to String1 of the second **Concat** Method Box.

G Map a link from the **text0** node under **auditPage.show.Output** to **String2** of the second **Concat** Method Box.

H Map a link from the **Return String** of the second **Concat** Method Box to the value node under **stringMessage**.

Figure 87 Add Business Rule to Second Link



Define the WSDL for the Receive and Reply Activity

1 Select the **Receive** Activity and then click **Property Sheet** from the eInsight ESB toolbar.

A Select **wlmpartner** from the **Partner** field.

B Select **sbcUserDef:PortType1** for the **Port Type**.

C Select **Operation1** for the **Operation**.

2 Repeat for the **Reply** Activity.

Create the Parent Business Process

1 Right-click the **wlmProject** and select **New Business Process**.

2 Add a **File Receive** Activity:

A Double-click the **SeeBeyond Project** from the Project Explorer tree view.

B Double-click **eWays** under the SeeBeyond Project.

C Double-click **File** under eWays.

D Double-click **FileClient** under File.

E Select and drag the **Receive** activity from **FileClient** to the Business Process Designer.

F Place the **Receive** activity to the right of the **Start** activity.

3 Add the File Write Activity:

A Double-click the **SeeBeyond Project** from the Project Explorer tree view.

B Double-click **eWays** under the SeeBeyond Project.

C Double-click **File** under eWays.

D Double-click **FileClient** under File.

E Select and drag the **Write** activity from **FileClient** to the Business Process Designer.

F Place the **Write** activity to the left of the **End** activity.

4 Add a User Activity:

A Select the User Activity icon from the eInsight ESB toolbar.

B Place the User Activity on the Business Process Designer, between the File Receive and File Write Activities.

5 Drag and drop the SubBusiness Process Operation Node (from the Project Explorer pane) onto the User Activity.

Figure 88 BusinessProcess1

9.4.6 Configure the Modeling Elements

1 Right-click on the link between the **File Receive Activity** and the **User Activity** and select **Add a Business Rule.**

2 Map a link from the **File Receive text** node to the input of the **subBusiness Process value** node.

3 Right-click on the link between the **User Activity** and the **File Write Activity** and select **Add a Business Rule**.

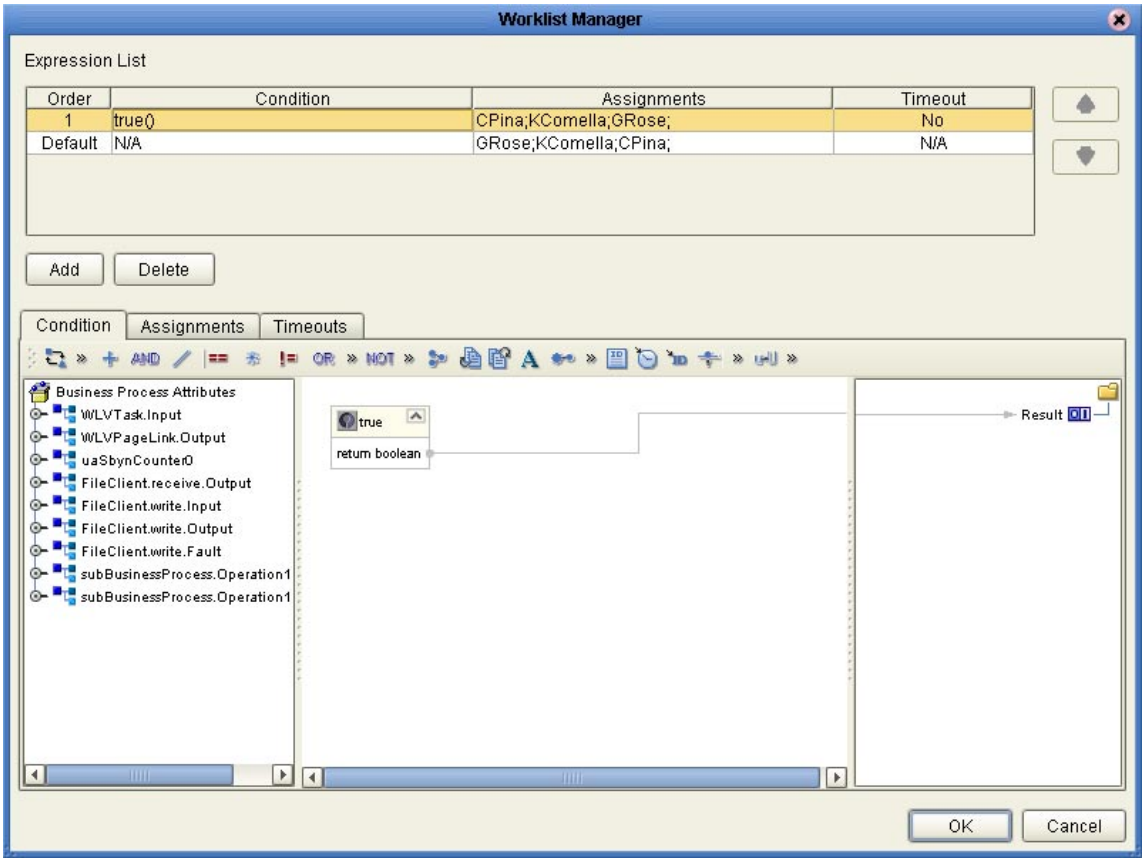
4 Map a link from the output **text** node of the **User Activity** to the **File Write text** node.

Configure the User Activity

1 Double-click the **User Activity**.

The User Activity properties appear, as shown in Figure 89.

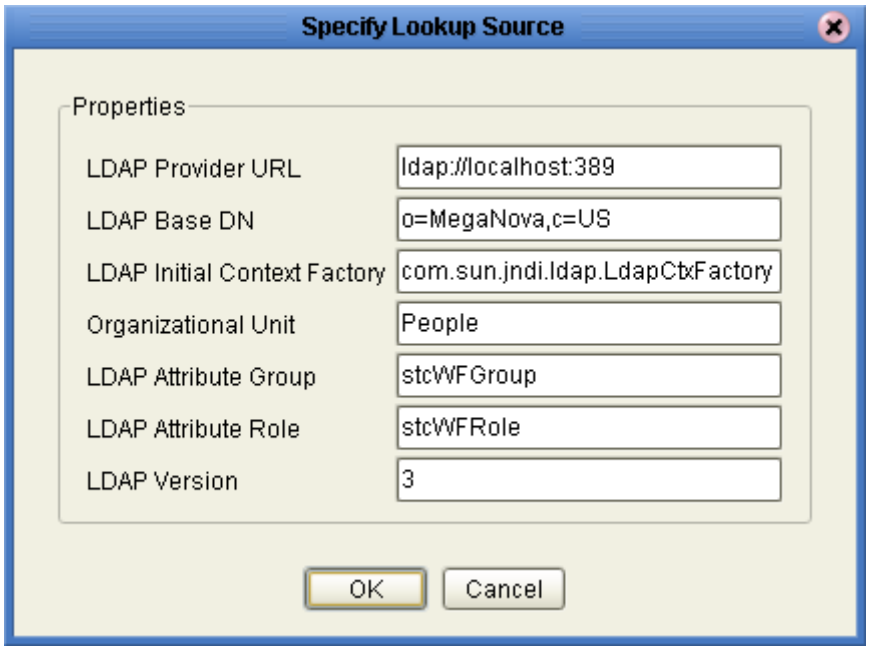
Figure 89 User Activity Properties



2 Click **Add** to create a new Case.

3 Click **Connect** to access the **LDAP** user directory.

Figure 90 Connect to LDAP



For this example, accept the default configuration.

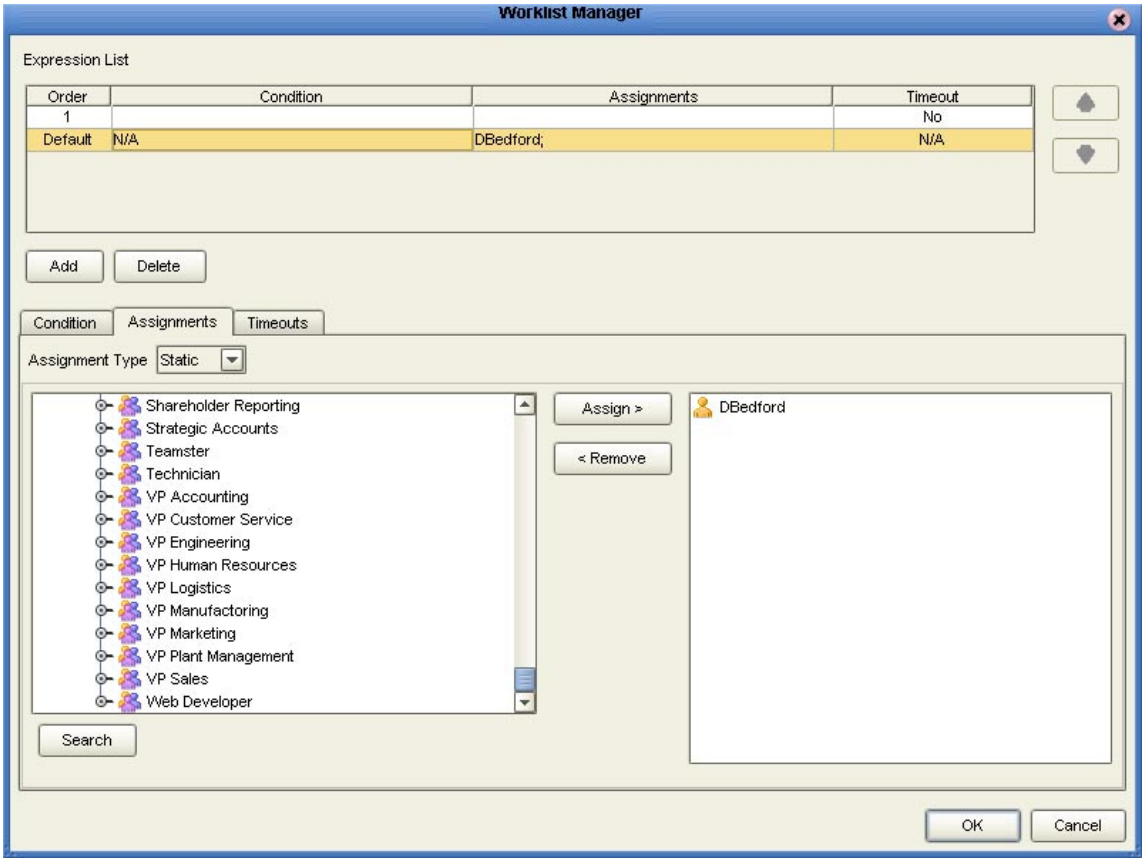
4 Select the following users, for the first case and the default, from the LDAP directory:

A GRose (located under Roles:Marketing)

B CPina (located under Roles:Executives)

C KComella (located under Roles:Executives)

Figure 91 Assign Users



5 Create a **True** condition for the first case.

Note: *Note this is a static configuration. Dynamic allows you to assign users during runtime.*

9.4.7 Create and Configure the Connectivity Map

Create the Connectivity Map

1 Right-click your Project and select **New: Connectivity Map**.

A new node will appear under your Project. The default name is **CMap1**.

2 Select the **External Applications** toolbar icon and select **File External Applications**.

3 Drag the **File** icon to the Connectivity Map canvas.

4 Drag a second **File** icon to the canvas.

5 Select **BusinessProcess1** from the Project Explorer and drag it to the canvas.

6 Select **subBusinessProcess** from the Project Explorer and drag it to the canvas.

7 Add a **WebConnector** and a **WLM Connector** to the Connectivity Map.

Configure the Connectivity Map Elements

1 Double-click the **BusinessProcess11** and **SubBusinessProcess1**.

The **Business Process Service** dialogs appear.

A Drag a link from **File Sender** Service to the **File1** icon.

B Drag a link from the **File Receiver** Service to **File_output** icon.

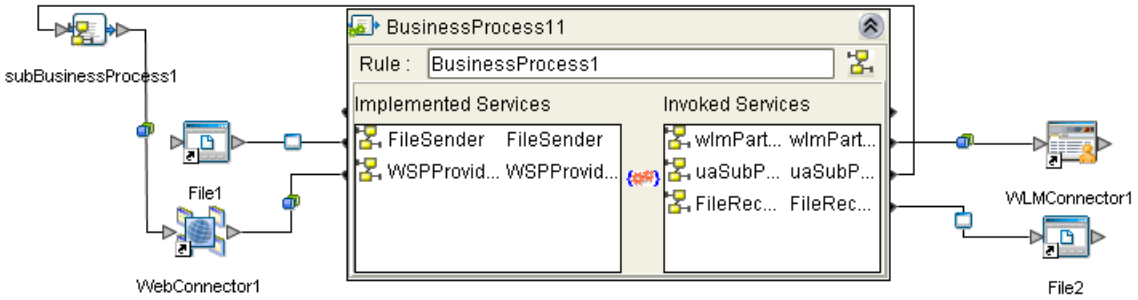
C Drag a link from the **WSPProvider** to the **WebConnector**.

D Drag the **WLMPartner** to the **WLM Connector**.

E Drag the **UASubProcPartner** to the **WLMPartner** on the SubProcess.

2 Click the minimize buttons on the **Business Process** dialogs to close.

Figure 92 Completed Connectivity Map



Configure the File Systems

Configure the Inbound File eWay

1 Double-click the link to **File1** to configure it.

The **Templates** dialog appears.

2 Select Inbound **File eWay** and select **OK**.

The **Properties** dialog appears.

3 Change the **Directory** to **C:\data**.

4 Change **Input** file name to **input*.txt**.

5 Click **OK** to save changes.

Configure the Outbound File eWay

1 Double-click the link **File2** to configure it.

The **Templates** dialog appears.

2 Select **Outbound File eWay** and select **OK**.

The **Properties** dialog appears.

3 Change the **Directory** to **C:\data**.

4 Change the **Output** file name to **output%d.txt.**

5 Click **OK** to save changes.

9.4.8 Deploy and Test the System

Deployment Profile

Create the Deployment Profile

1 Right-click your **Project** from the **Project Explorer**.

2 Select **New: Deployment Profile**.

3 The **Create Deployment Profile** dialog appears.

4 The **Deployment Profile** is called **Deployment1** by default. You can accept the default name.

5 Select the **Environment** (**Environment1**) that you created previously.

Configure the Deployment Profile

1 Drag **BusinessProcess11** and **SubBusinessProcess1** from the middle pane to the **Integration Server** (**IntegrationSvr1**) located in the **LogicalHost** window.

2 Drag **File1 -> BusinessProcess11** from the middle pane to the **File_IN** window.

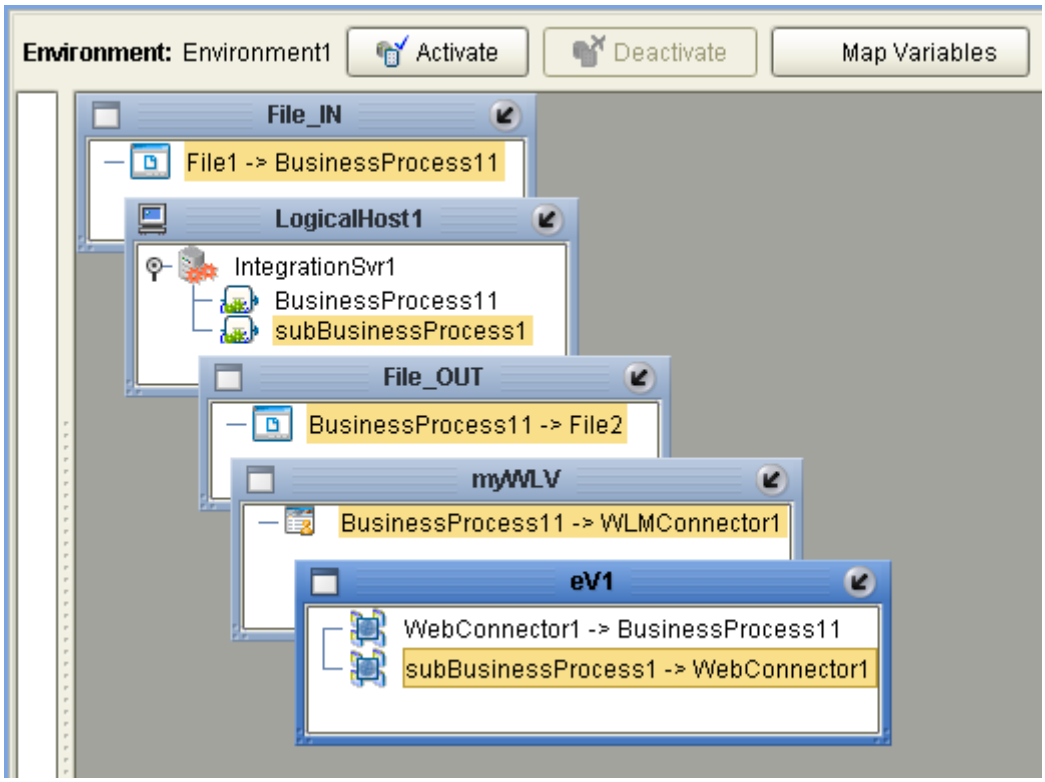
3 Drag **BusinessProcess11 -> File2** to the **File_OUT** window.

4 Drag **WebConnector1->BusinessProcess11** to **eV1**.

5 Drag **SubBusinessProcess1->WebConnector1** to **eV1**.

6 Drag **BusinessProcess11->WLMConnector** to **myWLV**.

Figure 93 Worklist Manager Deployment Profile



7 Click **Activate**.

8 Click **No**, so the change will not apply to the **Logical Host** immediately. (This is because your Logical Host is not running yet.)

Start the Logical Host

The Bootstrap process executes your **Project** and begins the process of polling your input data. The Bootstrap process is performed from a command prompt. Bootstrap will pick up the deployment profile the first time it runs; after that you would select reactivate and click **Yes** to apply the most recent changes to the Logical Host.

Note: *The Bootstrap command is case sensitive on Windows.*

To run the Bootstrap

1 Open a Windows command prompt.

2 Navigate to where you installed the Logical Host; for example, **ican50\logicalhost\bootstrap\bin**, then type the following command:

```
CD \ican50\logicalhost\bootstrap\bin
```

3 To start the Bootstrap process, type the following command:

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

- ***environment_name*** is the name of your environment (for example, *Environment1*),
- ***logicalhost_name*** is the name of your Logical Host (for example, *LogicalHost1*),
- ***repository_URL*** is the full URL of your Repository including the Repository name (for example, *http://localhost:12000/MyRep*),
- ***username*** is your user name, and
- ***password*** is your password.

Test the Task Assignment System

1. Connect to **http://localhost:18003/wlm** (This is the default for all WLMs).
2. Login as **GRose**.

Note: *Username: GRose, Password: pass.*

The task list is empty.

3. Copy your **input1.txt** file to **c:/data**.
4. Click refresh in the Worklist Manager.

   A new task appears.
5. Select the new task and click the **Checkout** button.
6. Click **Execute** to launch the task.

   The eVision **auditPage** appears, with the name portion completed with the name in your file.
7. Enter an amount in the amount text field.

   The exit page appears.
8. Click Close Window.

   You are back to the task view.
9. Click Complete from the task view.

   The task disappears from your task list.

Other tests

1. Login as **CPina**.
2. Escalate the task.
3. Logout and login as **KComella**.
4. Allow the **KComella** to complete the task.

Check output

1. Navigate to **C:\data** and check for an **output.txt** file.
2. If the file exists, open it and examine the data. It should look like this:

```
Mary Smith:10,000
```

Note: *Your input depends on the contents of your file and the entry you make in the Amount field.*

9.5 About the Samples

This section will guide you through importing and deploying the sample projects.

9.5.1 Import the Worklist Manager Sample

This sample and the accompanying files can be found in the eInsight_WLM_Sample.zip file. You may download the sample file from the **Documentation** tab of the Enterprise Manager. It is one of the files available when you upload the eInsightDocs.sar file.

Download the eInsight Sample Project

1. Open the Enterprise Manager and click on the Documentation tab.
2. Select eInsight Business Process Manager from the Products list.
3. Select Download Sample and save to your local drive.
4. Unzip the compressed file and extract the files to another folder.

   This compressed file contains the **wlmProject.zip** file and Readme.txt.

Import the Sample Project

To run the project, you must import the **wlmProject.zip** file.

1. Right-click your Repository folder in the Enterprise Explorer and select Import.

   The Import Manager dialog appears.
2. Click Browse and find **wlmProject.zip**.
3. Select the file and click Import.
4. Close the Import Manager dialog.

Once the import is complete, you can go directly to **"Deploy and Test the System" on page 126** to run your sample.

9.5.2 Import the User Activity Sample

This sample and the accompanying files can be found in the **eInsight_User_Activity_Sample.zip** file. You may download the sample file from the **Documentation** tab of the Enterprise Manager. It is one of the files available when you upload the eInsightDocs.sar file.

Download the eInsight Correlation Sample

1. Open the Enterprise Manager and click on the Documentation tab.
2. Select eInsight Business Process Manager from the Products list.
3. Select Download Sample and save to your local drive.

4 Unzip the compressed file and extract the files to another folder.

This compressed file contains the UserActivityProject.zip file as well as:

- input_ua1-KComella.txt
- input_ua2-CPina.txt
- output_ua1.dat
- Readme.txt

Import the Sample Project

To run the project, you must import the UserActivityProject.zip file.

1 Right-click your Repository folder in the Enterprise Explorer and select Import.

The Import Manager dialog appears.

2 Click Browse and find UserActivityProject.zip.

3 Select the file and click Import.

4 Close the Import Manager dialog.

Once the import is complete, continue to **"Deploy and Test the Project"** to run your sample.

9.5.3 Deploy and Test the Project

The final steps necessary to run your sample include:

- Creating and Configuring the Deployment Profile.
- Starting the Logical Host.
- Checking your output.

Note: *Check-out all components that are currently checked-in, so that you can make changes. Imported projects have several components checked-in by default.*

Deployment Profile

Create the Deployment Profile

1 Right-click your **Project** from the **Project Explorer**.

2 Select **New: Deployment Profile**.

3 The **Create Deployment Profile** dialog appears.

4 The **Deployment Profile** is called **Deployment1** by default. For this example, the default is used.

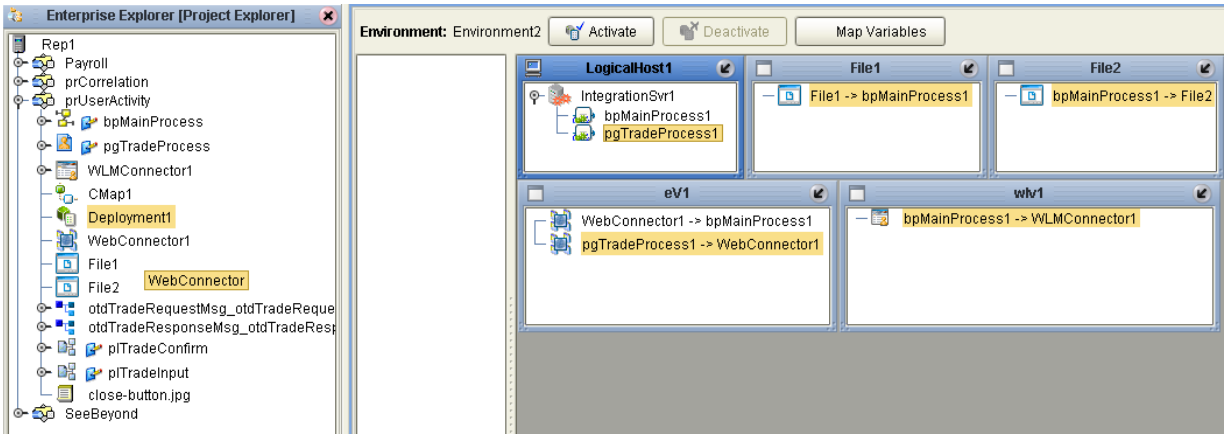
5 Select the **Environment**.

Configure the Deployment Profile

Configure the Deployment Profile

1. Drag **bpMainProcess1** and **pgTradeProcess1** from the middle pane to the **Integration Server** (**IntegrationSvr1**) located in the **LogicalHost** window.
2. Drag **File1 -> bpMainProcess1** from the middle pane to the **File1** window.
3. Drag **bpMainProcess1 -> File2** to the **File2** window.
4. Drag **WebConnector1->bpMainProcess1** to **eV1**.
5. Drag **pgTradeProcess1 ->WebConnector1** to **eV1**.
6. Drag **bpMainProcess1 ->WLMConnector** to **myWLV**.
7. Click **Activate**.

Figure 94 User Activity Deployment Profile



8. Click **No**, so the change will not apply to the **Logical Host** immediately. (This is because your Logical Host is not running yet.)

Start the Logical Host

The Bootstrap process executes your **project** and begins the process of polling your input data. The Bootstrap process is performed from a command prompt. Bootstrap will pick up the deployment profile the first time it runs; after that you would select reactivate and click **Yes** to apply the most recent changes to the Logical Host.

Note: *The Bootstrap command is case sensitive on Windows.*

To run the Bootstrap

1. Open a Windows command prompt as shown in the following Figure 95.
2. Navigate to where you installed the Logical Host; for example, **ican50\logicalhost\bootstrap\bin**, then type the following command:

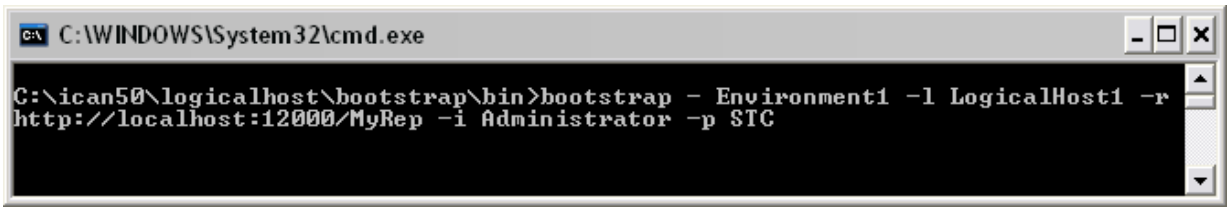
```
CD \ican50\logicalhost\bootstrap\bin
```

3. To start the Bootstrap process, type the following command:

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

- ***environment_name*** is the name of your environment (for this example, *CorrEnv*),
- ***logicalhost_name*** is the name of your Logical Host (for this example, *LogicalHost1*),
- ***repository_URL*** is the full URL of your Repository including the Repository name (for example, *http://localhost:12000/MyRep*),
- ***username*** is your user name
- ***password*** is your password

Figure 95 Bootstrap Command Example



Check output

1. Navigate to **C:\data** and check for the **output_ua1.dat** file.
2. Open the **output_ua1.dat** file and examine the data. It will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<otdTradeResponseMsg>
    <Trader>CPina</Trader>
    <Action>Sell</Action>
    <Quantity>2000</Quantity>
    <StockSymbol>SBYN</StockSymbol>
</otdTradeResponseMsg>
<?xml version="1.0" encoding="UTF-8"?>
<otdTradeResponseMsg>
    <Trader>KComella</Trader>
    <Action>Buy</Action>
    <Quantity>10</Quantity>
    <StockSymbol>EBAY</StockSymbol>
</otdTradeResponseMsg>
```

Appendix A

Method Palette

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

10.1 Operators

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

Figure 96 Method Palette: Operator Tab

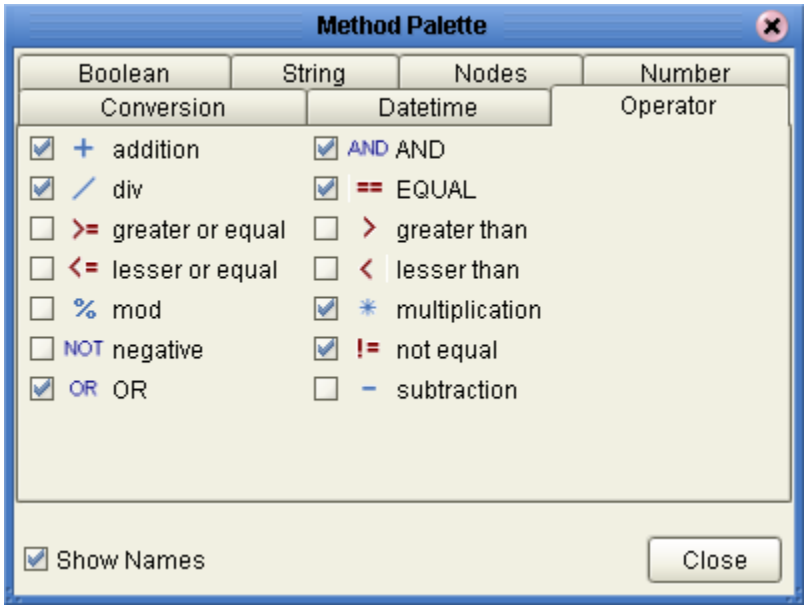


Table 11 Operator Methods

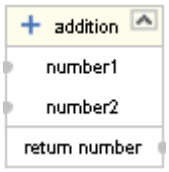
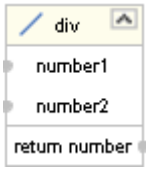
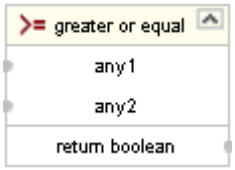
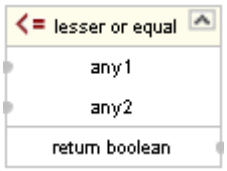
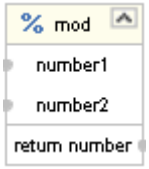
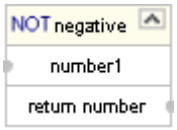
Symbol	Name	Function
	addition	Adds the value of *number1* to the value of *number2*, returns the sum.
	div	Divides the value of *number1* by the value of *number2*, returns the quotient.
	greater or equal	Returns Boolean true if *number1* is greater than or equal to *number2*; otherwise, returns Boolean false.
	lesser or equal	Returns Boolean true if *number1* is less than or equal to *number2*; otherwise, returns Boolean false.
	mod	Used to divide two numbers and return only the remainder.
	negative	Converts the input number to negative. Result is a negative number having the same absolute value as the input number.

Table 11 Operator Methods (Continued)

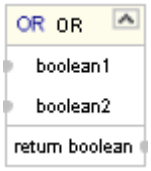
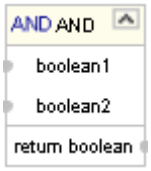
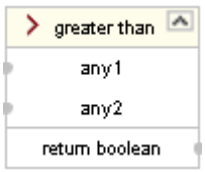
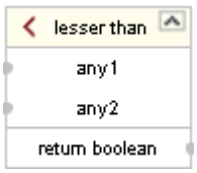
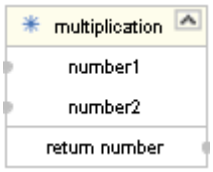
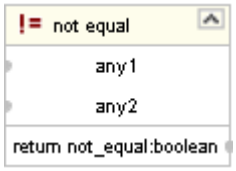
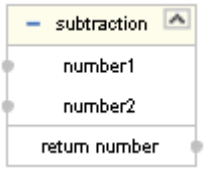
Symbol	Name	Function
	OR	Returns Boolean false if both *boolean1* and *boolean2* are false; otherwise, returns Boolean true.
	AND	Returns Boolean true if both *boolean1* and *boolean2* are true; otherwise, returns Boolean false.
	EQUAL	Returns Boolean true if *number1* is equal to *number2*; otherwise, returns Boolean false.
	greater than	Returns Boolean true if *number1* is greater than *number2*; otherwise, returns Boolean false.
	lesser than	Returns Boolean true if *number1* is less than *number2*; otherwise, returns Boolean false.
	multiplication	Multiplies the value of *number1* by the value of *number2*, returns the product.

Table 11 Operator Methods (Continued)

Symbol	Name	Function
	not equal	Returns Boolean true if *number1* is not equal to *number2*; otherwise, returns Boolean false.
	subtraction	Subtracts the numerical value of *number2* from the numerical value of *number1*, returns the difference.

10.2 String

The String methods allow you to manipulate string data.

Figure 97 Method Palette: String Tab

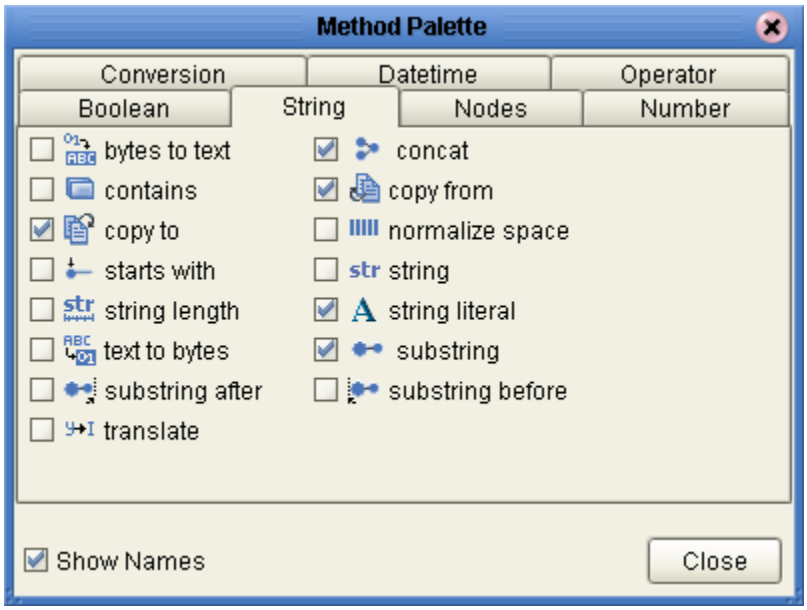


Table 12 String Methods

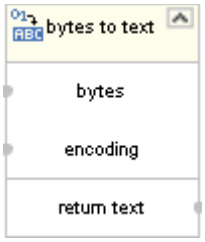
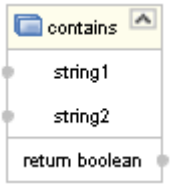
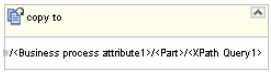
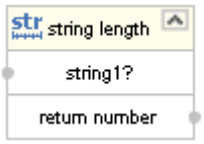
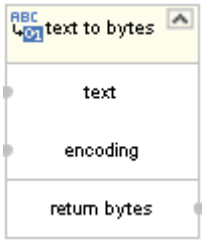
Symbol	Name	Function
	bytes to text	Decodes bytes into text using the specified encoding. If no encoding is specified, the platform's default encoding is used.
	contains	Returns true if the second string is contained within the first string, otherwise it returns false
	copy to	Allows you to type in the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

Table 12 String Methods (Continued)

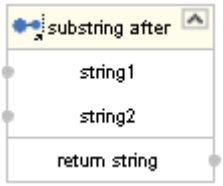
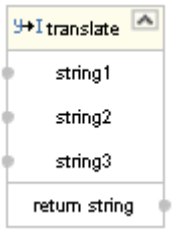
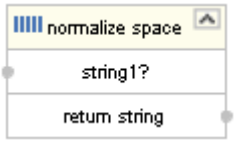
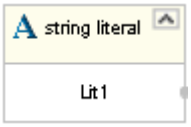
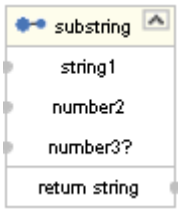
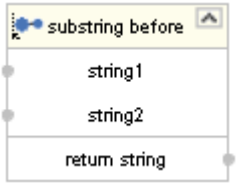
Symbol	Name	Function
	substring after	Returns the part of the string in the string argument that occurs after the substring in the substring argument
	translate	Performs a character by character replacement. It looks in the value argument for characters contained in string1, and replaces each character for the one in the same position in the string2
	concat	Returns the concatenation of all its arguments
	copy from	Allows you to type in xpath expression for the source of a copy operation. This is useful for entering xpath predicates. Note: This is for advanced users who are familiar with xpath and BPEL syntax
	normalize space	Removes leading and trailing spaces from a string
	string	Converts the value argument to a string

Table 12 String Methods (Continued)

Symbol	Name	Function
	string literal	A sequence of characters of fixed length and content
	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.

10.3 Number

The Number methods allow you to work with number data.

Figure 98 Method Palette: Number Tab

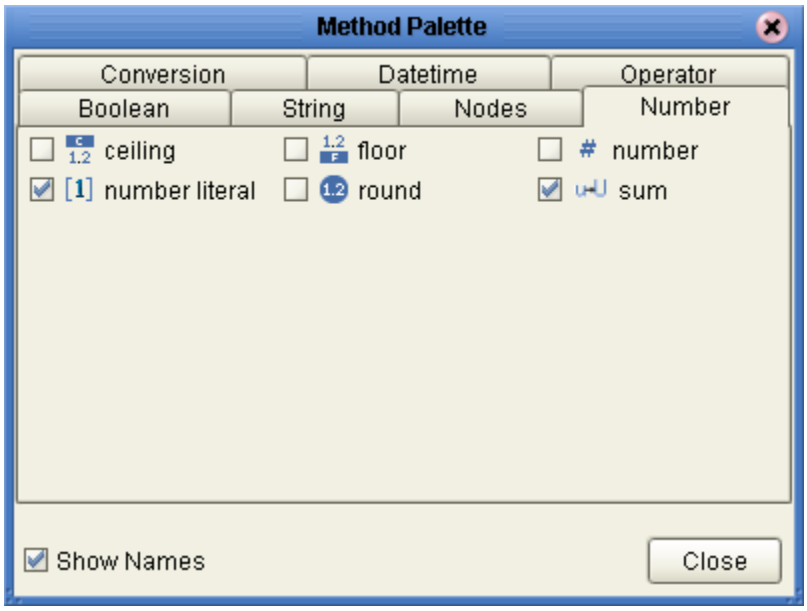


Table 13 Number Methods

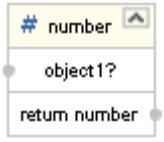
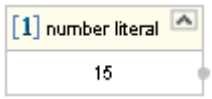
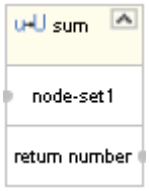
Symbol	Name	Function
	ceiling	Returns the smallest integer that is not less than the number argument
	floor	Returns the largest integer that is not greater than the number argument
	number	Converts the value argument to a number
	number literal	A literal number string of fixed length and content

Table 13 Number Methods (Continued)

Symbol	Name	Function
	round	Rounds the number argument to the nearest integer
	sum	Returns the total value of a set of numeric values in a node-set

10.4 Boolean

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

Figure 99 Method Palette: Boolean Tab

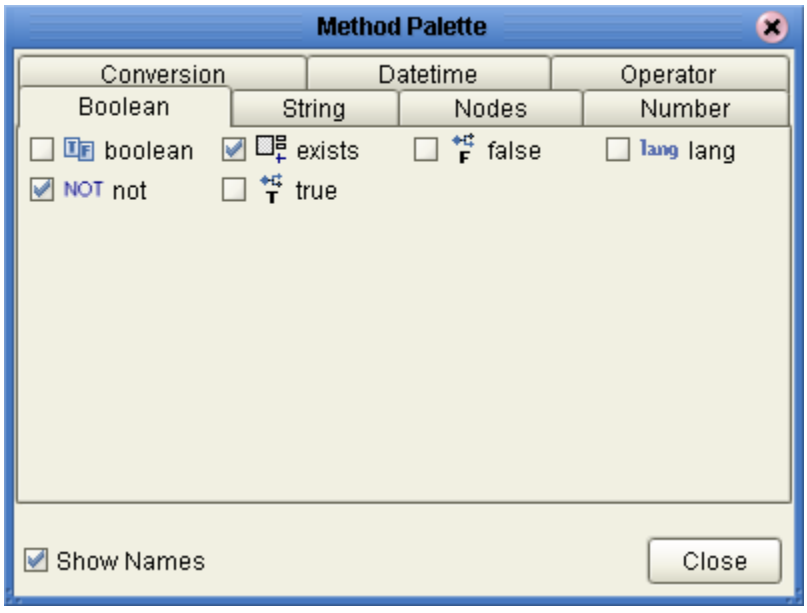
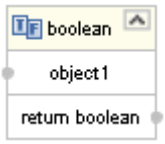
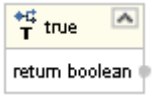
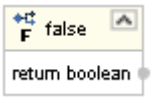
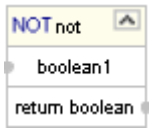
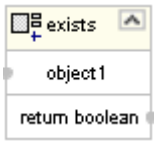


Table 14 Boolean Methods

Symbol	Name	Function
	boolean	Converts the value argument to Boolean and returns true or false.
	true	Returns true
	false	Returns false
	lang	Returns true if the language argument matches the language of the xsl:lang element, otherwise it returns false.
	not	Returns true if the condition argument is false, and false is the condition argument is true.
	exists	Checks to see if a value is present and returns a Boolean result.

10.5 Nodes

Node methods allow you to manipulate your data.

Figure 100 Method Palette: Nodes Tab

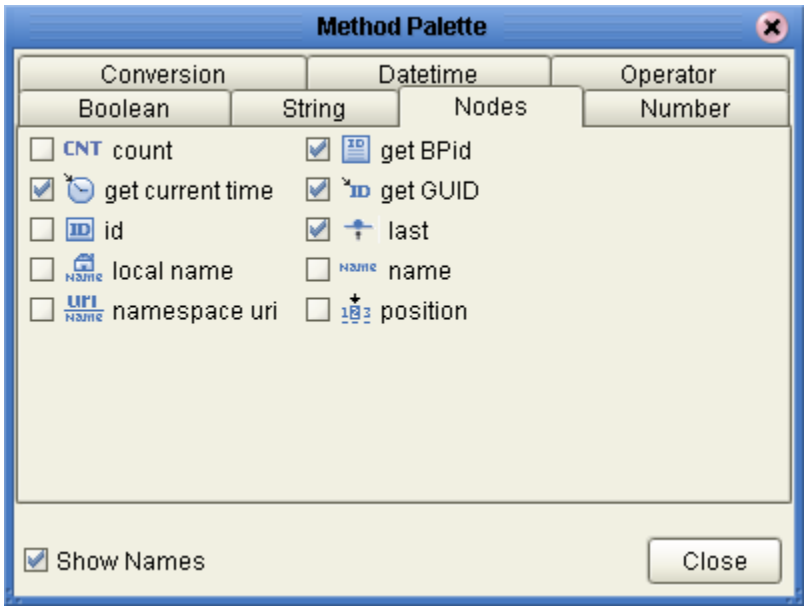


Table 15 Nodes Methods

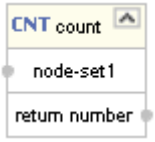
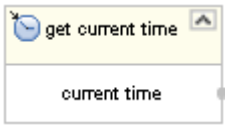
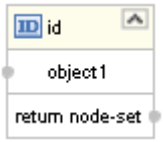
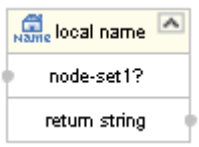
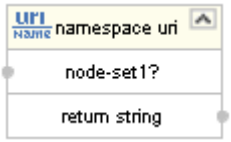
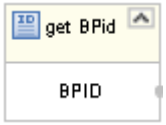
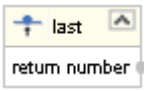
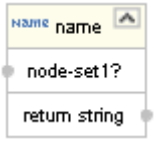
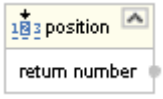
Symbol	Name	Function
	count	Returns the number of nodes in a node-set
	get current time	Gets the current time in ISO 8601 format (e.g. 2003-08-15T02:03:49.92Z).
	id	Selects elements by their unique ID
	local name	Returns the local part of a node. A node usually consists of a prefix, a colon, followed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

Table 15 Nodes Methods (Continued)

Symbol	Name	Function
	namespace uri	Returns the namespace URI of a specified node
	get BPid	Gets the business process instance ID.
	get GUID	Gets a randomly generated globally unique ID.
	last	Returns the position number of the last node in the processed node list
	name	Returns the name of a node
	position	Returns the position in the node list of the node that is currently being processed

10.6 Datetime

Datetime methods allow you to manipulate date, time, and duration of data.

Figure 101 Method Palette: Datetime Tab

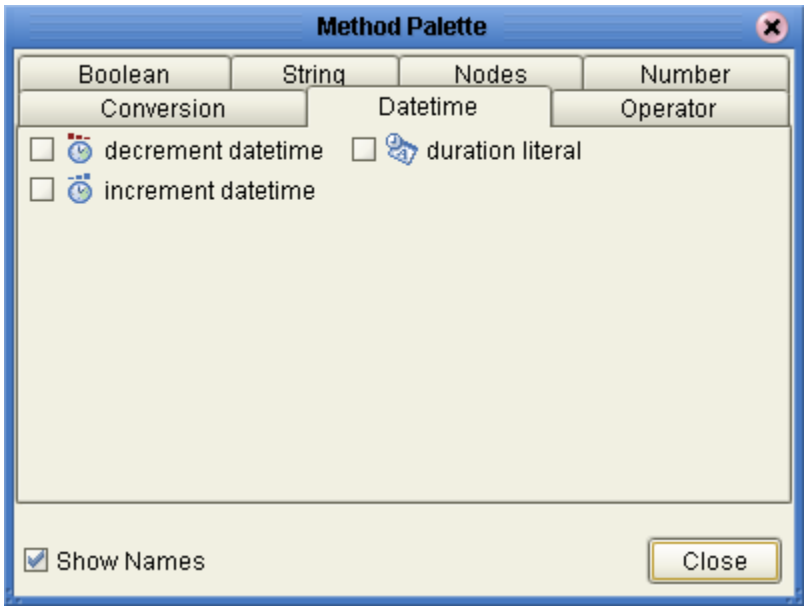
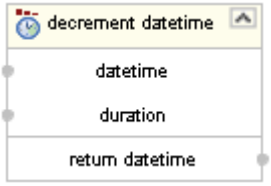
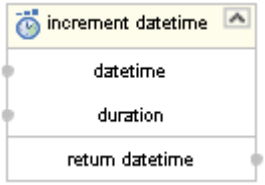
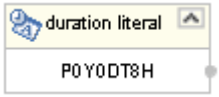


Table 16 Datetime Methods

Symbol	Name	Function
	decrement datetime	Dynamically decreases the date or time by a certain duration, such as days or hours.
	increment datetime	Dynamically increases the date or time by a certain duration, such as days or hours.
	duration literal	Allows you to set an actual date or time.

10.7 Conversion

The Convert method allows you to make conversions from various data types.

Figure 102 Method Palette: Conversion Tab

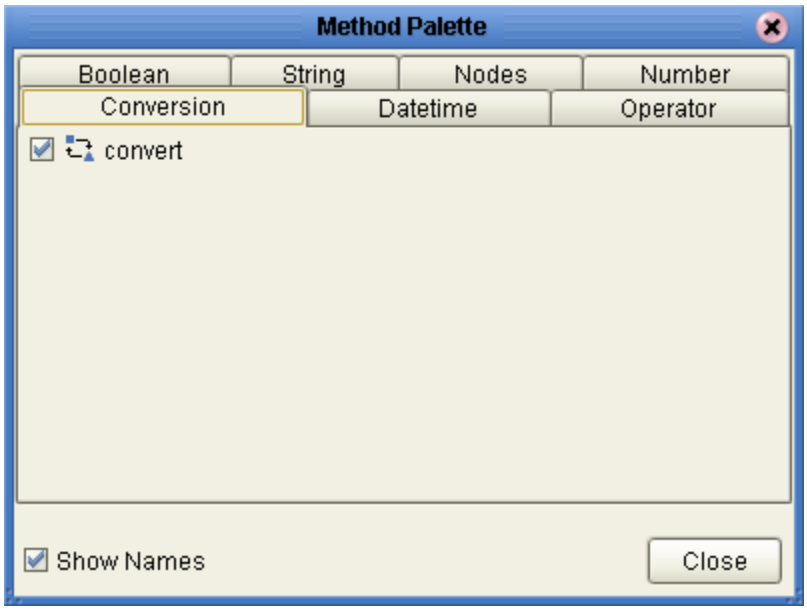
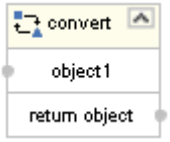


Table 17 Conversion Methods

Symbol	Name	Function
	convert	The convert function that takes in one input link and one output link. The data type conversions are described in **"Data Type Conversions" on page 146**.

10.7.1 Data Type Conversions

The Business Rule Designer supports a Convert function that takes in one input link and one output link. The Convert function is implemented from tree to tree mapping only. The Convert function is valid for conversions between leaf nodes. The Conversion function checks if the mapping is valid. The valid conversions are based off the following conversions.

String

Table 18 String

To	From
Boolean	custom
Float	parse
Double	parse
Decimal	parse
Byte	parse
Short	parse
Int	parse
Long	parse
Duration	parse
dateTime	parse
time	parse
date	parse
gYearMonth	parse
gYear	parse
gMonthDay	parse
gDay	parse
gMonth	parse
hexBinary	textToByte
base64Binary	textToByte
anyURI	parse
QName	parse
NOTATION	parse

Boolean

Table 19 Boolean

To	From
String	toString

Float

Table 20 Float

To	From
String	toString
Boolean	boolean
Double	floatToDouble
Decimal	floatToDecimal
Byte	floatToByte
Short	floatToShort
Int	floatToInt
Long	floatToLong

Double

Table 21 Double

To	From
String	toString
Boolean	boolean
Float	doubleToFloat
Decimal	doubleToDecimal
Byte	doubleToByte
Short	doubleToShort
Int	doubleToInt
Long	doubleToLong

Decimal

Table 22 Decimal

To	From
String	toString
Boolean	boolean
Float	decimalToFloat
Double	decimalToDouble
Byte	decimalToByte
Short	decimalToShort

Table 22 Decimal (Continued)

To	From
Int	decimalToInt
Long	decimalToLong

Byte

Table 23 Byte

To	From
String	toString
Boolean	boolean()
Float	byteToFloat
Double	byteToDouble
Decimal	byteToDecimal
Short	byteToShort
Int	byteToInt
Long	byteToLong

Short

Table 24 Short

To	From
String	toString
Boolean	boolean()
Float	shortToFloat
Double	shortToDouble
Decimal	shortToDecimal
Byte	shortToByte
Int	shortToInt
Long	shortToLong

Int

Table 25 Int

To	From
String	toString
Boolean	boolean()
Float	intToFloat
Double	intToDouble
Decimal	intToDecimal
Byte	intToByte
Short	intToShort
Long	intToLong

Long

Table 26 Long

To	From
String	toString
Boolean	boolean()
Float	longToFloat
Double	longToDouble
Decimal	longToDecimal
Byte	longToByte
Short	longToShort
Int	longToInt

Duration

Table 27 Duration

To	From
String	toString
Boolean	boolean

dateTime

Table 28 dateTime

To	From
String	toString
Boolean	boolean
time	dateTimeToTime
date	dateTimeToDate
gYearMonth	dateTimeToGYearMonth
gYear	dateTimeToGYear
gMonthDay	dateTimeToGMonthDay
gDay	dateTimeToGDay
gMonth	dateTimeToGMonth

time

Table 29 time

To	From
String	toString
Boolean	boolean

date

Table 30 date

To	From
String	toString
Boolean	boolean
gYearMonth	dateToGYearMonth
gYear	dateToGYear
gMonthDay	dateToGMonthDay
gDay	dateToGDay
gMonth	dateToGMonth

gYearMonth

Table 31 gYearMonth

To	From
String	toString
Boolean	boolean
gYear	gYearMonthToGYear
gMonth	gYearMonthToGMonth

gYear

Table 32 gYear

To	From
String	toString
Boolean	boolean

gMonthDay

Table 33 gMonthDay

To	From
String	toString
Boolean	boolean
gDay	gMonthDayToGDay
gMonth	gMonthDayToGMonth

gDay

Table 34 gDay

To	From
String	toString
Boolean	boolean

gMonth

Table 35 gMonth

To	From
String	toString
Boolean	boolean

hexBinary

Table 36 hexBinary

To	From
String	byteToText
Boolean	boolean
base64Binary	hexBinaryToBase64Binary

base64Binary

Table 37 base64Binary

To	From
String	byteToText
Boolean	boolean
hexBinary	base64BinaryToHexBinary

anyURI

Table 38 anyURI

To	From
String	toString
Boolean	boolean

QName

Table 39 QName

To	From
String	toString
Boolean	boolean

NOTATION

Table 40 NOTATION

To	From
String	toString
Boolean	boolean

Glossary

activity
An organizational unit for performing a specific function. An activity defines a step of a particular business process.

activity states
The stages that activities within the business process instance go through as the business process version is being run.

Business Process Attribute
Attributes pass user-defined control information (programming arguments) to and from the eInsight Enterprise Service Bus and its activities.

business process
A business process is a collection of actions and messages, revolving around a specific business practice, that flow in a specific pattern to produce an end result.

business process attributes
Business process attributes pass user-defined control information (programming arguments) to and from the e*Insight Business Process Manager, external sources, and internal components.

business process instance (BPI)
A unique instantiation of a business process.

business process model
The graphical representation of a business process.

business process version
A form or variant of the original business process model.

Collaboration
A component of an eWay that receives and processes Events and forwards the output to other eGate components.

eInsight Enterprise Service Bus (eInsight ESB)
The component within the ICAN Suite that facilitates the automation of the business process flow of business activities.

Decision
Controls the logical flow of data-based decisions in the business process model. A Decision outputs specific information when specified input conditions are met.

GUI
Graphical User Interface. A type of computer interface that enables the user to perform actions via the use of symbols, visual metaphors and pointing devices.

Business Process Designer
The Business Process Designer is the portion of the eInsight ESB where you create the business process model, in the form of a flow chart.

security
Security is the ability to limit user access to specific items based on a pre-determined profile of the user.

state
See *activity states*

string
A sequence of text characters.

Sub-process
A sub-process is a business process which is called, or used by, another business process, as a sub-component.

tree view
The tree view displays a hierarchical representation of all the components, and their activities.

User activity
Allows external applications to access attributes in the business process.

Index