SUN SEEBEYOND

eWAY™ ADAPTER FOR LOTUS NOTES/DOMINO USER'S GUIDE

Release 5.1.3

Sun microsystems

Part Number: 820-0988

Version 20070417082508

Contents

Chapter 1

Introduction

Welcome to the Sun SeeBeyond eWay™ Adapter for Lotus Notes/Domino User's Guide. This document describes how to install, configure, and implement the Sun SeeBeyond eWay™ Adapter for Lotus Notes/Domino in a typical Sun Java™ Composite Application Platform Suite environment.

What's in This Chapter

1.1 Lotus Notes/Domino eWay Overview

IBM's Lotus Notes/Domino integrated messaging and Collaboration platform is designed to provide email messaging, calendar (scheduling), and Enterprise integration, using a company's existing technology resources.

The Sun Java Composite Application Platform Suite, using the Lotus Notes/Domino eWay, enhances Lotus Notes/Domino by providing specialized application connectivity, robust data transformation, business logic execution, and management of intelligent message routing. The Lotus Notes/Domino eWay (referred to as the Notes/Domino eWay throughout this document) enables the Suite to access data from a Lotus Notes database.

The eWay acts as an outbound adapter to populate messages to a Lotus Notes/Domino database.

The Lotus Notes/Domino eWay supports

- Access of Remote or local databases.
- Sending email messages.
- Direct access by View or UNID.
- Sequential access to documents according to View.
- Ability to create, update, or delete documents.

- Double-byte Character Sets (DBCS) are supported to enable Japanese character support.

The Lotus Notes/Domino eWay supports Java Collaboration Definitions to provide Business Logic. The Lotus Notes eWay does not support eInsight Business Processes.

1.2 What's New in This Release

The Sun SeeBeyond eWay™ Adapter for Lotus Notes/Domino includes the following changes and new features:

New for Version 5.1.3

- This is a maintenance release. No new features.

New for Version 5.1.2

- This is a maintenance release. No new features.

New for Version 5.1.1

- Version Control: An enhanced version control system allows you to effectively manage changes to the eWay components.
- Multiple Drag-and-Drop Component Mapping from the Deployment Editor: The Deployment Editor now allows you to select multiple components from the Editor's component pane, and drop them into your Environment component.
- Support for Runtime LDAP Configuration: eWay configuration properties now support LDAP key values.
- Connectivity Map Generator: Generates and links your Project's Connectivity Map components using a Collaboration or Business Process.
- Added support for Lotus Notes/Domino version 7.0.

Many of these features are documented further in the *Sun SeeBeyond eGate Integrator User's Guide* or the *Sun SeeBeyond eGate Integrator System Administrator Guide*.

1.3 What's in This Document

This document provides information about installing, configuring, and using the Sun SeeBeyond eWay™ Adapter for Lotus Notes/Domino and includes the following chapters:

- **Chapter 1 "Introduction"** provides an overview of the Lotus Notes/Domino eWay.
- **Chapter 2 "Installing the Lotus Notes/Domino eWay"** describes how to install the Lotus Notes/Domino eWay and documentation.
- **Chapter 3 "Lotus Notes/Domino eWay Properties"**describes how to configure the Lotus Notes/Domino eWay to run in your environment.

- **Chapter 4 "Implementing a Lotus Notes/Domino eWay Project"** provides an introduction to the Lotus Notes/Domino eWay components and information on how these components are created and implemented in an eGate Project.

Lotus Notes/Domino Javadoc

The Lotus Notes/Domino eWay Javadoc documents the available Java methods provided with the Lotus Notes/Domino eWay. The Javadoc is uploaded with the eWay's documentation file, **LotusNoteseWayDocs.sar**, and downloaded from the Documentation tab of the Sun Java Composite Application Platform Suite Installer. To access the full Javadoc, extract the Javadoc to an easily accessible folder, and double-click the **index.html** file.

1.3.1 Scope of the Document

This user's guide provides a description of the Sun SeeBeyond eWay™ Adapter for Lotus Notes/Domino. It includes directions for installing the eWay, configuring the eWay properties, and implementing the eWay's sample Projects. This document is also intended as a reference guide, listing available properties, functions, and considerations. For a reference of available Lotus Notes/Domino eWay Java methods, see the associated Javadoc.

1.3.2 Intended Audience

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

1.3.3 Text Conventions

The following conventions are observed throughout this document.

Table 1 Text Conventions

Text Convention	Used For	Examples
Bold	Names of buttons, files, icons, parameters, variables, methods, menus, and objects	▪ Click **OK**. ▪ On the **File** menu, click **Exit**. ▪ Select the **eGate.sar** file.
`Monospaced`	Command line arguments, code samples; variables are shown in ***`bold italic`***	`java -jar ***filename***.jar`
Blue bold	Hypertext links within document	See **Text Conventions** on page 8
Blue underlined	Hypertext links for Web addresses (URLs) or email addresses	http://www.sun.com

1.4 Sun Microsystems, Inc. Web Site

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

http://www.sun.com

1.5 Documentation Feedback

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

CAPS_docsfeedback@sun.com

Chapter 2

Installing the Lotus Notes/Domino eWay

This chapter explains the procedures for installing the Lotus Notes/Domino eWay.

What's in This Chapter

2.1 Lotus Notes/Domino eWay System Requirements

The Lotus Notes/Domino eWay Readme contains the latest information on:

- Supported Operating Systems
- System Requirements
- External System Requirements
- The Lotus Notes/Domino eWay Readme is uploaded with the eWay's documentation file (NotesDominoeWayDocs.sar) and can be accessed from the Documentation tab of the Sun Java Integrator Suite Installer. Refer to the Lotus Notes/Domino eWay Readme for the latest requirements before installing the Lotus Notes/Domino eWay.

2.2 Installing the Lotus Notes/Domino eWay

The Sun Java™ Composite Application Platform Suite Installer, a web-based application, is used to select and upload eWays and add-on files during the installation process. The following section describes how to install the components required for this eWay.

Note: *When the Repository is running on a UNIX operating system, the eWays are loaded from the Sun Java Composite Application Platform Suite Installer running on a Windows platform connected to the Repository server using Internet Explorer.*

2.2.1 Installing the eWay on a JavaCAPS Supported System

Follow the directions for installing the Sun Java Composite Application Platform Suite in the *Sun Java Composite Application Platform Suite Installation Guide. After you have installed eGate, do the following:*

1. From the Sun Java Composite Application Platform Suite Installer's **Select Sun Java Composite Application Platform Suite Products to Install** table (Administration tab), expand the **eWay** option.
2. Select the products for your Sun Java Composite Application Platform Suite and include the following:
   - **FileeWay** (the File eWay is used by most sample Projects)
   - NotesDominoeWay.sar (to install the Lotus Notes/Domino eWay)

   To upload the Lotus Notes/Domino eWay User's Guide, Help file, Javadoc, Readme, and sample Projects, select the following:
   - **NotesDominoeWayDocs.sar**
3. Once you have selected all of your products, click **Next** in the top-right or bottom-right corner of the **Select Sun Java Composite Application Platform Suite Products to Install** box.
4. From the **Selecting Files to Install** box, locate and select your first product's SAR file. Once you have selected the SAR file, click **Next**. Your next selected product appears. Follow this procedure for each of your selected products. The **Installation Status** window appears and installation begins after the last SAR file has been selected.
5. Once your eWay installation is finished, continue installing the Sun Java Composite Application Platform Suite as instructed in the *Sun Java Composite Application Platform Suite Installation Guide.*

Adding the eWay to an Existing Suite Installation

If you are adding the eWay to an existing Sun Java Composite Application Platform Suite installation, do the following:

1. Complete steps 1 through 4 above.
2. Once your product's installation is finished, open the Enterprise Designer and select **Update Center** from the Tools menu. The **Update Center Wizard** appears.
3. For Step 1 of the wizard, simply click **Next**.
4. For Step 2 of the wizard, click the **Add All** button to move all installable files to the **Include in Install** field, then click **Next**.
5. For Step 3 of the wizard, wait for the modules to download, then click **Next**.
6. The wizard's Step 4 window displays the installed modules. Review the installed modules and click **Finish.**
7. When prompted, restart the IDE (Integrated Development Environment) to complete the installation.

After Installation

Once you install the eWay, it must then be incorporated into a Project before it can perform its intended functions. See the *Sun SeeBeyond eGate™ Integrator User's Guide* for more information on incorporating the eWay into an eGate Project.

2.2.2 Installing eWay Enterprise Manager plug-ins

The **Sun SeeBeyond Enterprise Manager** is a Web-based interface that allows you to monitor and manage your Java Composite Application Platform Suite applications. The Enterprise Manager requires an eWay specific "plug-in" for each of your installed eWays. These plug-ins enable the Enterprise Manager to target specific alert codes for each eWay type, as well as to start and stop the inbound eWays.

The *Sun Java Composite Application Platform Suite Installation Guide* describes how to install the Sun SeeBeyond Enterprise Manager. The *Sun SeeBeyond eGate™ Integrator System Administration Guide* describes how to monitor servers, Services, logs, and alerts using the Sun SeeBeyond Enterprise Manager and the command-line client.

The **eWay Enterprise Manager plug-ins** are available from the **List of Components to Download** under the Sun Java Composite Application Platform Suite Installer's **Downloads** tab.

There are two ways to add the eWay Enterprise Manager plug-ins:

1. From the Enterprise Manager:
   - A. From the **Enterprise Manager**'s Explorer toolbar, click the **Configuration** icon.
   - B. Click the **Web Applications Manager** tab, go to the **Auto-Install from Repository** tab, and connect to your Repository.
   - C. Select the application plug-ins you require, and click **Install**. The application plug-ins are installed and deployed.
2. From the **Sun Java Composite Application Platform Suite Installer:**
   - A. From the **Sun Java Composite Application Platform Suite Installer's Download tab**, select the Plug-Ins you require and save them to a temporary directory.
   - B. Log onto the **Sun SeeBeyond Enterprise Manager**. From the **Enterprise Manager**'s Explorer toolbar, click the **Configuration** icon.
   - C. Click the **Web Applications Manager** tab and go to the **Manage Applications** tab.
   - D. Browse for and select the WAR file for the application plug-in that you downloaded, and click **Deploy**. The plug-in is installed and deployed.

Lotus Notes/Domino eWay Alert Codes

You can view and delete alerts using the Enterprise Manager. An alert is triggered when a specified condition occurs in a Project component. The purpose of the alert is to warn the administrator or user that a condition has occurred.

To View the eWay Alert Codes

1. Add the eWay Enterprise Manager plug-in for this eWay.
2. From the Enterprise Manager's **Explorer** toolbar, click the **Configuration** icon.
3. Click the **Web Applications Manager** tab and go to the **Manage Alert Codes** tab. Your installed alert codes are displayed under the **Results** section. If your eWay alert codes are not available displayed under **Results**, do the following
   A. From the **Install New Alert Codes** section, browse to and select the eWay alert properties file for the application plug-in that you added. The alert properties files are located in the **alertcodes** folder of your Sun Java Composite Application Platform Suite installation directory.
   B. Click **Deploy**. The available alert codes for your application are displayed under **Results**. A listing of available this eWay's alert codes is displayed in Table 2.

Table 2 Lotus Notes/Domino eWay Alert Codes

Alert Code	Description	User Action
NOTESDOMINOEWAY-GETDATABASE-FAILED000001	Failed to open notes/domino database {0} on server {1}.	External configuration information is invalid. Verify that the following are correct: ▪ Database Type ▪ Notes/Domino Database ▪ Notes/Domino Server (if connected remotely)
NOTESDOMINOEWAY-GETDOCUMENT-FAILED000003	Failed to get Document;	*Reserved for future development*
NOTESDOMINOEWAY-SENDEMAIL-FAILED000002	Failed to send document as email.	*Reserved for future development*
NOTESDOMINOEWAY-SESSION-FAILED000001	Failed to establish session to notes/domino database {0} on server {1} as user {2}.	▪ The Domino server is down; start your server ▪ External configuration information is invalid. Verify that the following are correct: ♦ Database Type ♦ Notes/Domino Database ♦ Notes/Domino Server ♦ Notes/Domino User ♦ Password

Alert Code	Description	User Action
NOTESDOMINOEWAY-SESSION-FAILED000002	Failed to establish session to notes/domino, Reason {0}.	▪ Check the Reason description for the cause of not able to establish a session ▪ External configuration information is invalid. Verify that the following are correct: ♦ Database Type ♦ Notes/Domino Database ♦ Notes/Domino Server ♦ Notes/Domino User ♦ Password

An alert code is a warning that an error has occurred. It is not a diagnostic. The user actions noted above are just some possible corrective measures you may take. Refer to the log files for more information. For information on Managing and Monitoring alert codes and logs, see the *Sun SeeBeyond eGate Integrator System Administration Guide.*

2.3 Adding Lotus Notes/Domino JAR files

One of the following Lotus Notes/Domino JAR files must be added to the Enterprise Designer and the Logical Host of your Sun Java Composite Application Platform Suite. These Jar files are available from your Lotus Notes/Domino installation. Choose the JAR file that is appropriate for your system.

- **Notes.jar** is required for local Domino object access. The file is located in the Notes directory.
- **NCSO.jar** is required for remote Domino object access. The file is located in the Notes/Data/domino/java directory.

Add the JAR file to your Enterprise Designer

Prior to building your Lotus Notes/Domino eWay Project, add your selected JAR file to the Enterprise Designer in the following location:

```
<JavaCAPS51>\edesigner\usrdir\modules\ext\notesdominoadapter
```

where `<JavaCAPS51>` is your Suite installation directory.

Add the JAR file to your Logical Host

Prior to deploying your Lotus Notes/Domino eWay Project, add your selected JAR file to the Logical Host in the following location:

```
<JavaCAPS51>\logicalhost\is\lib
```

where <*JavaCAPS51*> is your Suite installation directory.

Note: *If the Logical Host is running, it must be recycled to pickup the new Jar files.*

2.3.1 Enabling Java access to a Remote Domino Server

For remote access (when using **NCSO.jar**) the following steps are required:

- To enable Java access to a remote Domino server, a DIIOP and HTTP server must be configured and run. Running the DIIOP and HTTP server creates the file **diiop_ior** in the domino/html directory. The diiop_ior.txt file is queried by the remote application for a connection. If the DIIOP server has not been run previously, there is no diiop_ior.txt file in the directory.
- Anonymous logging for HTTP must be turned on when using HTTP. If DIIOP is used, configure the port number in the Environment properties.
- In addition, to access a remote Lotus Notes/Domino server using anything other than the configured default port, you must configure the Notes/Domino Server property as **server:port** *(server name:port number)*.

See **Java Access to the Domino Objects** at **http://lotus.com/** for more information.

2.4 ICAN 5.0 Project Migration Procedures

This section describes how to transfer your current ICAN 5.0 Projects to Sun Java Composite Application Platform Suite, version 5.1.3. Only Projects developed on ICAN version 5.0.2 and above can be migrated successfully to the Sun Java Composite Application Platform Suite. To migrate your ICAN 5.0 Projects, do the following:

Export the Project

1. Before you export your Projects, save your current ICAN 5.0 Projects to your Repository.
2. From the Project Explorer, right-click your Project and select **Export** from the shortcut menu. The Export Manager appears.
3. Select the Project that you want to export in the left pane of the Export Manager and move it to the Selected Projects field by clicking the **Add to Select Items** (arrow) button, or click **All** to include all of your Projects.
4. In the same manner, select the Environment that you want to export in the left pane of the Export Manager and move it to the Selected Environments field by clicking the **Add to Select Items** (arrow) button, or click **All** to include all of your Environments.
5. Browse to select a destination for your Project ZIP file and enter a name for your Project in the **ZIP file** field.
6. Click **Export** to create the Project ZIP file in the selected destination.

Install Sun Java Composite Application Platform Suite

7. Install the Sun Java Composite Application Platform Suite, including all eWays, libraries, and other components used by your ICAN 5.0 Projects.
8. Start the Sun SeeBeyond Enterprise Designer.

Import the Project

9 From the Enterprise Designer's Project Explorer tree, right-click the Repository and select **Import Project** from the shortcut menu. The Import Manager appears.

10 Browse to and select your exported Project file.

11 Click **Import**. A warning message, **"Missing APIs from Target Repository**," may appear at this time. This occurs because various product APIs were installed on the ICAN 5.0 Repository when the Project was created, that are not installed on the Sun Java Composite Application Platform Suite Repository. These APIs may or may not apply to your Projects. You can ignore this message if you have already installed all of the components that correspond to your Projects. Click **Continue** to resume the Project import.

12 Close the Import Manager after the Project is successfully imported.

Deploy the Project

13 A new Deployment Profile must be created for each of your imported Projects. When a Project is exported, the Project's components are automatically *"checked in"* to Version Control to write-protected each component. These protected components appear in the Explorer tree with a red padlock in the bottom-left corner of each icon. Before you can deploy the imported Project, the Project's components must first be *"checked out"* of Version Control from both the Project Explorer and the Environment Explorer. To *"check out"* all of the Project's components, do the following:

- A From the Project Explorer, right-click the Project and select **Version Control > Check Out** from the shortcut menu. The Version Control - Check Out dialog box appears.
- B Select **Recurse Project** to specify all components, and click **OK**.
- C Select the Environment Explorer tab, and from the Environment Explorer, right-click the Project's Environment and select **Version Control > Check Out** from the shortcut menu.
- D Select **Recurse Environment** to specify all components, and click **OK**.

14 If your imported Project includes File eWays, these must be reconfigured in your Environment prior to deploying the Project. To reconfigure your File eWays, do the following:

- A The Environment File External System properties can now accommodate both inbound and outbound eWays. If your previous Environment includes both inbound and outbound File External Systems, delete one of these (for example, the outbound File External System).
- B From the Environment Explorer tree, right-click your remaining File External System, and select **Properties** from the shortcut menu. The Properties Editor appears.
- C The Directory property has been relocated from the Connectivity Map Properties to the Environment Properties. Set the inbound and outbound Directory values, and click **OK**.

15 Deploy your Projects.

Note: *Only Projects developed on ICAN 5.0.2 and above can be imported and migrated successfully into the Java Composite Application Platform Suite.*

Chapter 3

Lotus Notes/Domino eWay Properties

This chapter describes how to create and configure the Lotus Notes/Domino eWay.

What's in This Chapter

3.1 Configuring the Lotus Notes/Domino eWay Properties

All eWays contain a set of configuration properties with parameters that are unique to that eWay type. After an eWay is created and an External System for that eWay type is created in the Project's Environment, the eWay's properties can be modified for your specific system. The Lotus Notes/Domino eWay properties are modified from the **Environment Explorer tree**. These parameters are commonly global, applying to all eWays (of the same type) in the Project.

Unlike many other eWays, none of the Lotus Notes/Domino eWay properties are accessed from the Connectivity Map.

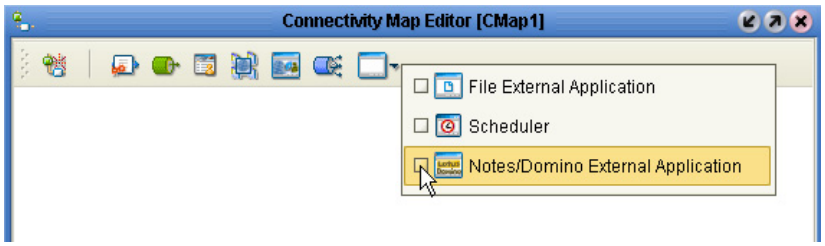
3.1.1 Selecting Lotus Notes/Domino as the External Application

To create a Lotus Notes/Domino eWay you must first create a Lotus Notes/Domino External Application in your Connectivity Map. Lotus Notes/Domino eWays are located between a Lotus Notes/Domino External Application and a Service. Services are containers for Java Collaborations, Business Processes, eTL processes, and so forth.

To create the Lotus Notes/Domino External Application

1. From the Connectivity Map toolbar, click the External Applications icon.
2. Select the **Lotus Notes/Domino External Application** from the menu (see **Figure 1 on page 19**). The selected Lotus Notes/Domino External Application icon appears on the Connectivity Map toolbar.

Figure 1 External Applications Selection Menu



3 Drag the new **Lotus Notes/Domino External Application** from the toolbar onto the Connectivity Map canvas. This represents an external Lotus Notes/Domino system.

From the Connectivity Map, you associate (bind) the External Application with the Service to establish an eWay. When **Notes/Domino** is selected as the External Application, it automatically applies the default Lotus Notes/Domino eWay properties, provided by the OTD, to the eWay that connects it to the Service. Most component eWays (other eWays, such as the File eWay) display an eWay properties node in the link between the External Application and the service. The Lotus Notes/Domino eWay does not possess Connectivity Map properties, and therefore, no eWay node is provided in this link.

3.1.2 Modifying the Lotus Notes/Domino eWay Properties

A Project's eWay properties can be modified after the eWays have been established in the Connectivity Map and the Environment has been created.

1 From the Environment Explorer tree, right-click the Lotus Notes/Domino External System. Select **Properties** from the shortcut menu (or double-click the External System). The **Properties Editor** opens with the Lotus Notes/Domino eWay Environment properties.

2 Make any necessary modifications to the Environment parameters of the Lotus Notes/Domino eWays, and click **OK** to save the settings.

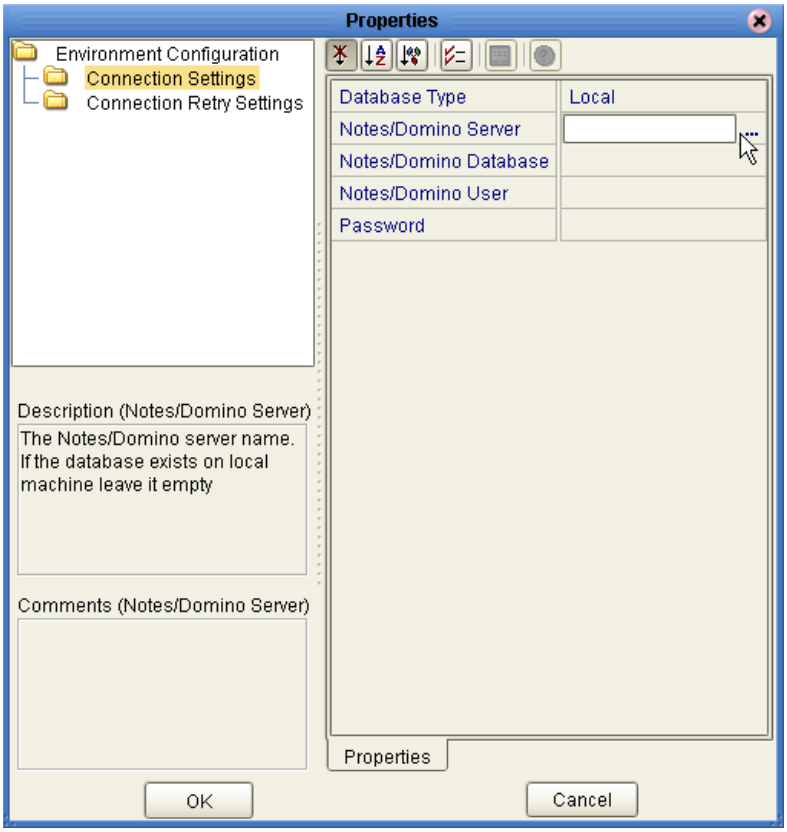
3.1.3 Using the Properties Editor

Modifications to the eWay properties are made using the Lotus Notes/Domino eWay Properties Editor.

Modifying the Default eWay Properties

1 From the upper-right pane of the Properties Editor, select the Connection Settings subdirectory of the Environment Configuration directory. The editable parameters contained in that subdirectory are now displayed in the right pane, as shown in **Figure 2 on page 20**.

Figure 2 Properties Editor - Lotus Notes/Domino eWay Environment Settings



2 Click on any property field to make it editable. For example, click on the **Notes/Domino Server** property to edit the server name. If a parameter's value is true/false or multiple choice, the field reveals a submenu of property options.

When selected, many fields offer an ellipsis button. Click on the ellipsis (. . .) in the properties field to open a separate configuration dialog box. This is helpful for large values that cannot be fully displayed in the parameter's property field. Enter the property value in the dialog box and click **OK**. The value is now displayed in the parameter's property field.

3 A description of each parameter is displayed in the **Description** pane when that parameter is selected, providing an explanation of any required settings or options.

4 The **Comments** pane provides an area for recording notes and information regarding the currently selected parameter. This is saved for future referral.

5 After modifying the configuration properties, click **OK** to close the Properties Editor and save the changes.

3.2 Lotus Notes/Domino eWay Environment Properties

The Lotus Notes/Domino eWay configuration parameters, accessed by right-clicking the Lotus Notes/Domino eWay External System in the Environment Explorer tree, are organized into the following sections:

3.2.1 Connection Settings

The **Connection Settings** section of the Lotus Notes/Domino eWay Environment properties contains the top-level parameters displayed in Table 3.

Table 3 Environment - Connector Properties

Name	Description	Required Value
Database Type	Specifies whether the Notes/Domino database is on a local or remote server.	Select **Local** or **Remote**. ***Note:*** *See* **"Add the Required JAR File to the Logical Host." on page 54** *for additional requirements.*
Notes/Domino Server	Specifies the name of the Notes/Domino server.	The name and port number of the Notes/Domino server in the following format: *servername:port*. If the database is located on a local computer leave this field blank.
Notes/Domino Database	Specifies the name of the Notes/Domino database. The value cannot be **empty String**, or **null**.	The Notes/Domino database name. The value cannot be **empty String**, or **null**.
Notes/Domino User	Specifies a user name with authorization to access the Notes/Domino database.	The valid user login name used to access the database.
Password	Specifies the password for the Notes/Domino user.	The login password for the specified Notes/Domino user.

3.2.2 Connection Retry Settings

The **Connection Retry Settings** section of the Lotus Notes/Domino eWay Environment properties contains the top-level parameters displayed in Table 3.

Table 4 Environment - Connector Properties

Name	Description	Required Value
ConnectionRetries	Specifies the number of attempts eGate allows to establish a connection with the Notes/Domino server.	An integer indicating the number of attempts allowed to establish a connection. The configured default is **20**.
ConnectionRetryInterval	Specifies the configured length of the pause before each reattempt to access the destination file. This property is used in conjunction with the property **"ConnectionRetries" on page 22**.	An integer indicating the configured length of the time in milliseconds before each reattempt to access the destination file. The configured default is **10000 (10 seconds).**

Chapter 4

Implementing a Lotus Notes/Domino eWay Project

This chapter provides an introduction to the Lotus Notes/Domino eWay components and information on how these components are created and implemented in an eGate Project. It is assumed that the reader understands the basics of creating a Project using the SeeBeyond Enterprise Designer. For more information about creating eGate Projects, see the *Sun SeeBeyond eGate Integrator Tutorial* and the *Sun SeeBeyond eGate Integrator User's Guide*.

Chapter Topics

4.1 Lotus Notes/Domino eWay Components

This chapter presents two sample Lotus Notes/Domino eWay Projects. The eWay components that are unique to the Lotus Notes/Domino eWay include the following:

NotesDomino OTD

The NotesDomino OTD contains methods and attributes used to create the Business Rules that invoke the Notes/Domino program.

Lotus Notes/Domino eWay Properties File

The properties file for the Lotus Notes/Domino eWay contains parameters used to connecting with specific external systems. These parameters are configured using the Properties Editor. For more information about the Lotus Notes/Domino eWay properties file and the Properties Editor see **Configuring the Lotus Notes/Domino eWay Properties** on page 18.

4.2 Lotus Notes/Domino eWay Considerations

Note the following considerations before proceeding with the sample:

Adding Lotus Notes/Domino JAR files

One of the following Lotus Notes/Domino JAR files must be added to the Enterprise Designer and the Logical Host of your Sun Java Composite Application Platform Suite. These files are available from your Lotus Notes/Domino installation. Choose the JAR file that is appropriate for your system.

- **Notes.jar** is required for local Domino object access. The file is located in the Notes directory.
- **NSCO.jar** is required for remote Domino object access. The file is located in the Notes/Data/domino/java directory.

This JAR file must be added to your Enterprise Designer prior to building your Lotus Notes/Domino eWay Project, and to your Logical Host prior to deploying your eWay Project.

For more information see **"Adding Lotus Notes/Domino JAR files" on page 14**.

4.3 Importing a Sample Project

The Sample eWay Projects are included as part of the installation package. To import a sample eWay Project to the Enterprise Designer do the following:

1. The sample files are uploaded with the eWay's documentation SAR file and downloaded from the Enterprise Manager's Documentation tab. The **NotesDomino_Sample.zip** file contains the various sample Project zip files. Extract the samples from the Enterprise Manager to a local file.
2. Save all unsaved work before importing a Project.
3. From the Enterprise Designer's Project Explorer pane, right-click the Repository and select **Import** from the shortcut menu. The **Import Manager** appears.
4. Browse to the directory that contains the sample Project zip file. Select the sample file (for this sample, **prjNotesDomino_Sample_JCD.zip**) and click **Import**. After the sample Project is successfully imported, click **Close**.
5. Before an imported sample Project can be run you must do the following:
   - Create an **Environment** (see **"Creating an Environment" on page 54**)
   - Configure the eWays for your specific system (see **"Configuring the eWays" on page 55**)
   - Create a **Deployment Profile** (see **"Creating the Deployment Profile" on page 57**)

- Create and start a domain (see **"Creating and Starting the Domain" on page 58**)
- Build and deploy the Project (see **"Building and Deploying the Project" on page 58**)
- Copy the sample database file, **SAMPLE.nsf**, that was downloaded with the sample zip file, to your Lotus/Notes/Data file.

4.4 The Lotus Notes/Domino eWay Sample Projects Overview

The Lotus Notes Domino eWay includes two sample Projects:

- **prjNotesDomino_Sample_JCD**
- **prjNotesDomino_EMailSample_JCD**

4.4.1 The prjNotesDomino_Sample_JCD Project Overview

The **prjNotesDomino_Sample_JCD** Project contains four sample processes: RetrieveDoc, CreateDoc, UpdateDoc, and DeleteDoc, based on the Collaboration of the same name. Each of these processes are run by the sample Project, depending on the input data file name.

- **RetrieveDoc:** Retrieves all document and items in the Lotus Notes/Domino database. These are written to the output file.
- **CreateDoc:** Creates a single document with one or more items, in the Lotus Notes/Domino database. The output file presents the message "Document Added" if the process succeeds or "Document Not Created" if the process fails.
- **UpdateDoc:** Updates a specific document and/or the items, and saves it to the original document. The output file presents the message "Item Updated" if the process succeeds or "Item Failed to Update" if the process fails.
- **DeleteDoc:** Deletes the specified document in the Lotus Notes/Domino database. The output file presents the message "Document Deleted" if the process succeeds or "Document Failed to Delete" if the process fails.

4.4.2 The prjNotesDomino_EMailSample_JCD Project Overview

The **prjNotesDomino_EMailSample_JCD** Project contains one sample process (Collaboration): jcdMail. jcdMail sends a document to the recipient(s) through the Lotus Notes/Domino server. An email document can contain up to three different recipient fields: **SendTo**, **CopyTo**, and **BlindCopyTo**. Other than the recipient items, an email document can contain additional items, such as **Subject**, **Body**, and **Attachment**. This sample Project demonstrates how to add these items to a Notes document.

4.5 Creating the prjNotesDomino_Sample_JCD Project

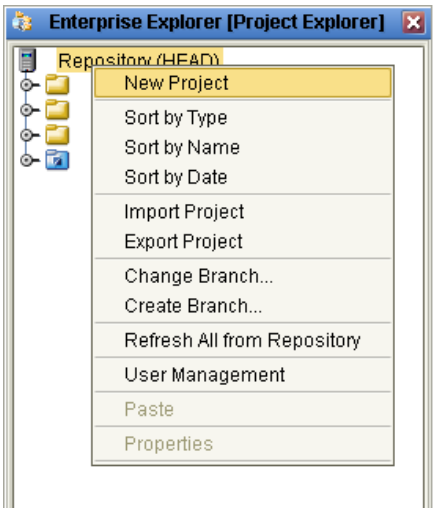
The following pages provide step by step directions that demonstrate how the sample Project and its components are created manually.

4.5.1 Creating a Project

The first step is to create a new Project in the SeeBeyond Enterprise Designer.

1. Start the Enterprise Designer.
2. From the Project Explorer tree, right-click the Repository and select **New Project** (see **Figure 3 on page 26**). A new Project (**Project1**) appears on the Project Explorer tree.

Figure 3 Enterprise Explorer - New Project



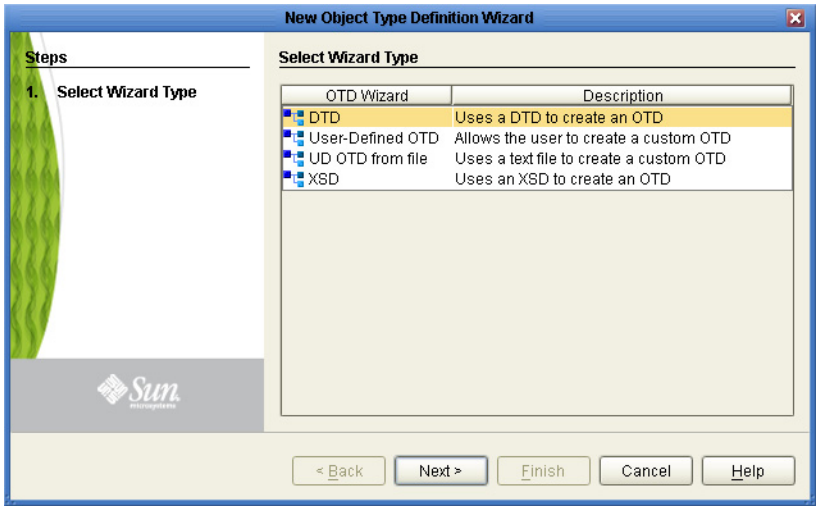
3. Rename the Project (for this sample, **prjNotesDomino_Sample_JCD**).

4.5.2 Creating OTDs from the DTD Files

The next step in the sample is to create two OTDs from the **NOTES.dtd** (Document Type Definition) and the **NotesUpdate.dtd** files provided with the sample. This is done using the **New Object Type Definition Wizard**.

1. From the Project Explorer tree, right-click your sample Project and select **New** > **Object Type Definition** from the shortcut menu. The **New Object Type Definition Wizard** appears.
2. From the **Select Wizard Type** box select **DTD** as the wizard type and click **Next**.

Figure 4 OTD Wizard



3 From the **Browse DTD Files** pane browse to and select the **Notes.dtd** file that was downloaded with the sample Project. Click **Select**. The **Notes.dtd** file appears in the **Selected DTD Files** box. Repeat this process to select the **NotesUpdate.dtd** file also. Click **Next**.

4 For step three of the wizard, Select Document Elements, select both **Notes_NOTES_DOCUMENT** and **NotesUpdate_UPDATE_ITEM**, and click **Next**.

5 For step four of the wizard, **Select OTD Options**, leave the default settings (do not select any options) and click **Finish**.

6 Both new OTDs now appear under your Project in the Project Explorer tree, and can now be used in your Collaborations.

4.5.3 Creating the Collaboration Definitions

The next step in the prjNotesDomino_Sample_JCD Project is to create four Java Collaborations using the **Collaboration Definition Wizard (Java)**. Once the Collaboration Definitions have been created, the Business Rules of the Collaboration are written using the **Collaboration Editor (Java)**.

Creating the CreateDoc Collaboration Definition

1 From the Project Explorer, right-click the sample Project and select **New** > **Collaboration Editor (Java)** from the shortcut menu. The **Collaboration Definition Wizard (Java)** appears.

2 Enter a Collaboration Definition name (for this sample, **CreateDoc**) and click **Next**.

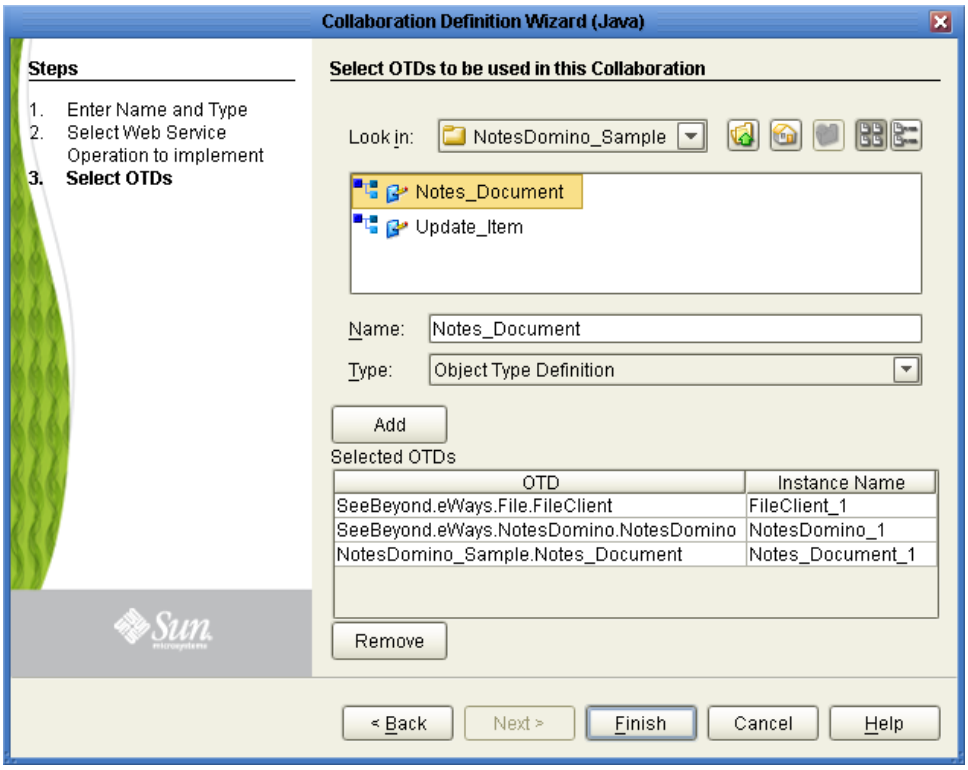
3 For Step 2 of the wizard, from the Web Services Interfaces selection window, double-click **Sun SeeBeyond** > **eWays** > **File** > **FileClient** > **receive**. The File Name field now displays **receive.** Click **Next**.

4 For Step 3 of the wizard, from the Select OTDs selection window, double-click **Sun SeeBeyond** > **eWays** > **File** > **FileClient**. The **FileClient_1** OTD is added to the Selected OTDs field.

5 Click the **Up One Level** button to return to the Repository. Double-click **Sun SeeBeyond** > **eWays** > **NotesDomino** > **NotesDomino**. The Selected OTDs field now lists the **NotesDomino_1** OTD (see Figure 5).

6 Click the **Up One Level** button to return to the Repository. Double-click **prjNotesDomino_Sample_JCD** > **Notes_Document**. The Selected OTDs field now lists the **NOTES_DOCUMENT_1** OTD (see **Figure 5 on page 28**).

Figure 5 Collaboration Definition Wizard (Java) - Select OTDs



7 Click **Finish**. The Collaboration Editor (Java) with the new Collaboration, **CreateDoc** appears, and the Collaboration is added to the Project Explorer tree.

Creating the DeleteDoc, RetrieveDoc, and UpdateDoc Collaboration Definition

Create three additional Collaborations for your Project following the same steps as those used to create the **CreateDoc** Collaboration. Table 5 contains the options required to create each Collaboration.

Table 5 prjNotesDomino_Sample_JCD Collaboration Files

Collaboration Name and type	Web Service Operation to Implement	Select OTDs
DeleteDoc	Sun SeeBeyond\eWays\File\FileClient**receive**	▪ Sun SeeBeyond\eWays\File/**FileClient** ▪ Sun SeeBeyond\eWays\NotesDomino**NotesDomino** ▪ Sun prjNotesDomino_Sample_JCD/**Update_Item**
RetrieveDoc	Sun SeeBeyond\eWays\File\FileClient**receive**	▪ Sun SeeBeyond\eWays\File/**FileClient** ▪ Sun SeeBeyond\eWays\NotesDomino**NotesDomino** ▪ Sun prjNotesDomino_Sample_JCD**Notes_Document**

Collaboration Name and type	Web Service Operation to Implement	Select OTDs
UpdateDoc	Sun SeeBeyond\eWays\File\FileClient**receive**	▪ Sun SeeBeyond\eWays\File**FileClient** ▪ Sun SeeBeyond\eWays\NotesDomino**NotesDomino** ▪ Sun prjNotesDomino_Sample_JCD**Update_Item**

Upon completion there are four Collaborations listed in the Project Explorer tree, and the Collaboration Editor is displayed with four tabs at the bottom of the window, one for each of the new Collaborations.

4.5.4 Using the Collaboration Editor (Java)

The next step to create the sample Project is to create the Business Rules for each of the four Collaborations using the Collaboration Editor (Java). Step by step directions for creating the Business Rules for each Collaboration are located as follows:

- **Create the CreateDoc Collaboration Business Rules** on page 29
- **Creating the DeleteDoc Collaboration Business Rules** on page 35
- **Creating the RetrieveDoc Collaboration Business Rules** on page 40
- **Creating the UpdateDoc Collaboration Business Rules** on page 44

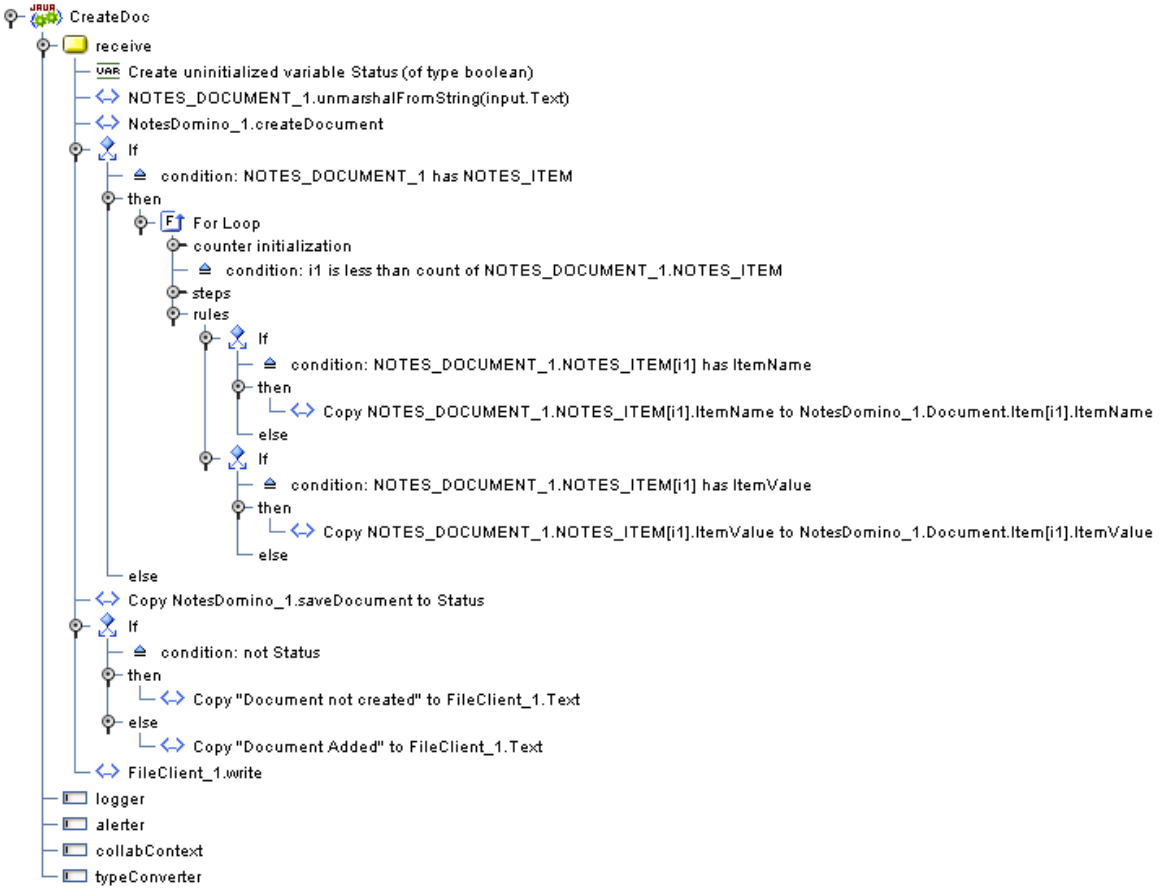
If you happen to make a mistake while creating a Business Rule, simply right-click the rule and select **Delete** from the shortcut menu.

Once you have created all the Business Rules for a Collaboration, click the Validate icon on the Collaboration Editor's toolbar to "precompile" the Java Collaboration Definition. If any errors are found, they are displayed in a validation panel that appears at the bottom of the Collaboration Editor. To locate a noted error in the Collaboration, double-click a specific error message and the Java Source Editor displays the erroneous line of code.

Create the CreateDoc Collaboration Business Rules

Be careful to open all nodes specified in the directions to connect to the correct items. The **CreateDoc** Collaboration contains the Business Rule displayed in Figure 6.

Figure 6 CreateDoc Collaboration Business Rules



The Collaboration in Figure 6 displays the CreateDoc Business Rules provided with the imported sample.

To create the **CreateDoc** Collaboration Business Rules do the following:

1. Open the Collaboration Editor to the **CreateDoc** Collaboration by clicking the **CreateDoc** tab if available, or by double-clicking the **CreateDoc** Collaboration from the Project Explorer tree.
2. To create Business Rule comments, from the Business Rules toolbar, click the **comment** icon. The **Enter a Comment** dialog box appears. Enter the comment and click **OK**. The comment is placed on the Business Rules tree under the last selected item. Once the Comment is created, it can be moved by clicking the comment and dragging it up or down the Business Rules tree to a new location.
3. Create the **Create uninitialized variable Status (of type boolean)** variable rule:
   A. Select the **receive** method in the Business Rule pane.
   B. From the Business Rules toolbar, click the **Local Variable** icon. The **Create Variable** dialog box appears.
   C. Enter **Status** as the Variable Name, select **Primitive** and **boolean** as the Type, and click **OK**. The boolean Status variable is added to the Business Rule tree.
4. Create the **NOTES_DOCUMENT_1.unmarshalFromString(input.Text)**:

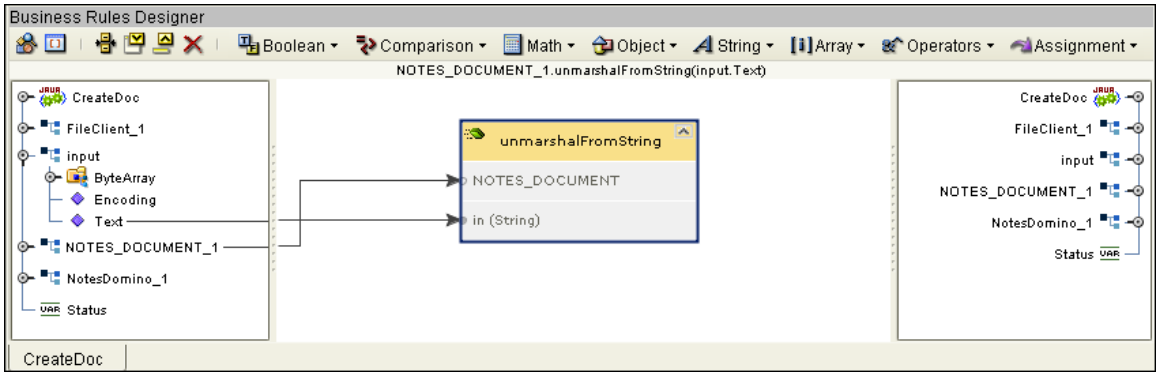
A From the Business Rules tree, select the **Empty Rule**.

B Right-click **NOTES_DOCUMENT_1** in the left pane of the Business Rules Designer, and select **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **unmarshalFromString()**. The **unmarshalFromString** method box appears.

D Map **Text** under **input** in the left pane of the Business Rules Designer to the **in (String)** input node of the **unmarshalFromString** method box. To do this, click on **Text** in the left pane of the Business Rules Designer and drag the cursor to the **in (String)** input node of the **unmarshalFromString** method box (see Figure 7).

Figure 7 Business Rules Designer - CreateDoc Collaboration Business Rules



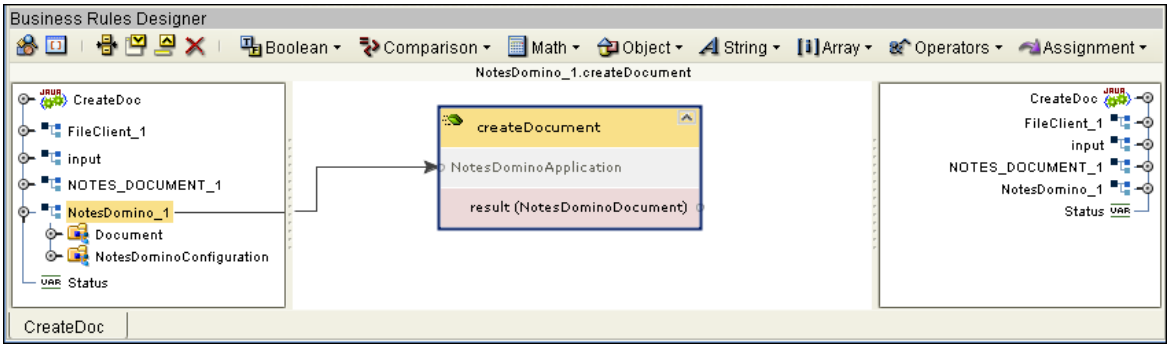
5 Create the **NotesDomino_1.createDocument** rule:

A From the Business Rules toolbar, click the **rule** icon to add a new rule.

B Right-click **NotesDomino_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **createDocument()**. The **createDocument** method box appears on the Business Rules Designer canvas (see Figure 8).

Figure 8 Business Rules Designer - CreateDoc Collaboration Business Rules

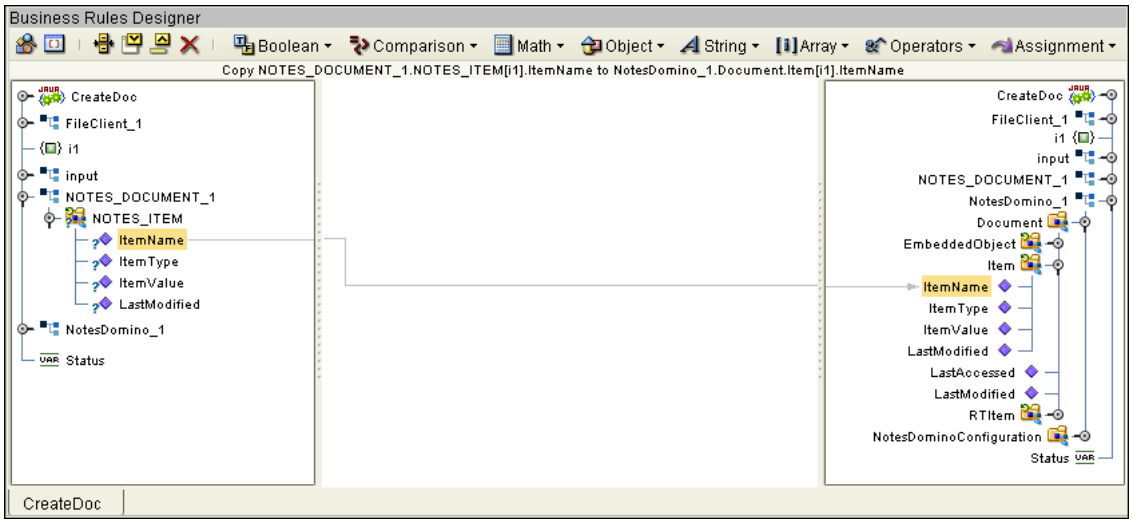


6 Create the **If** statement, **For Loop** and **Copy NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemName to NotesDomino_1.Document.Item[i1].ItemName** rule:

A From the Business Rules toolbar, click the **rule** icon to add a new rule.

B Map **ItemName** under **NOTES_DOCUMENT_1** > **NOTES_ITEM** in the left pane of the Business Rules Designer, to **ItemName** under **NotesDomino_1** > **Documents** > **Item** in the right pane of the Business rules Designer. To do this, click on **ItemName** in the left pane of the Business Rules Designer, and drag your cursor to **ItemName** under **NotesDomino_1** > **Documents** > **Item** in the right pane. A new **If** statement and **For Loop** appear in the Business Rules window and the **Copy NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemName to NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemName** rule is added under the **For Loop** (see Figure 9).

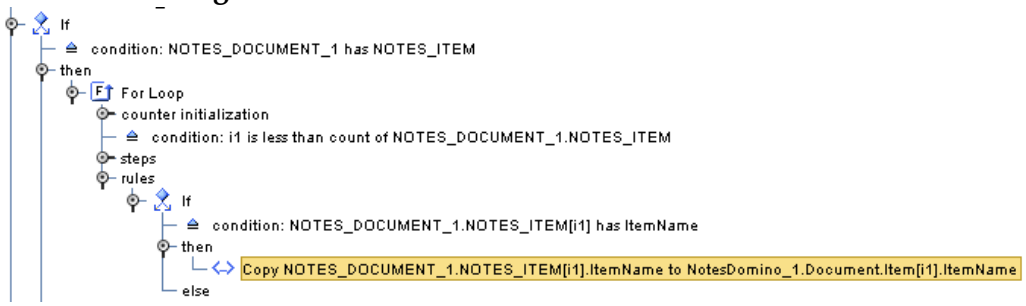
Figure 9 Business Rules Designer - CreateDoc Collaboration Business Rules



7 Create another **If** statement and the **Copy NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemValue to NotesDomino_1.Document.Item[i1].ItemValue** rule within the **For Loop**:

A From the Business Rules tree, select the **Copy NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemName to NotesDomino_1.Document.Item[i1].ItemName** rule under the **If** statement within the **For Loop** (see Figure 10).

Figure 10 Select then statement rule



B Map **ItemValue** under **NOTES_DOCUMENT_1** > **NOTES_ITEM** in the left pane of the Business Rules Designer, to **ItemValue** under **NotesDomino_1** > **Documents** > **Item** in the right pane of the Business Rules Designer. A new **If**

statement is created (as a sibling to the last **If** statement), and the **Copy NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemValue to NotesDomino_1.Document.Item[i1].ItemValue** rule is added.

8 Create the **Copy NotesDomino_1.saveDocument to Status** Business Rule:

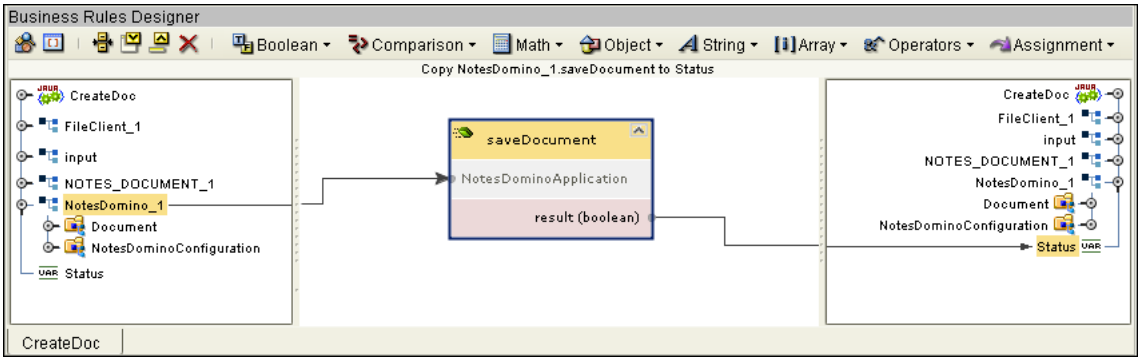
A From the Business Rules tree, select the **For Loop**. From the Business Rules toolbar, click the rule icon to add a new rule as a sibling to the **For Loop**.

B Right-click **NotesDomino_1** in the left pane of the Business Rules Designer, and select **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **saveDocument()**. The **saveDocument** method box appears.

D Map the **results (boolean)** output node of the **saveDocument** dialog box, to the **Status** variable in the right pane of the Business Rules Designer (see Figure 11).

Figure 11 Business Rules Designer - CreateDoc Collaboration Business Rules



9 Create the last **If** statement and the **not Status** condition:

A From the Business Rules toolbar, click the **If** icon. An If statement is added to the Business Rules tree.

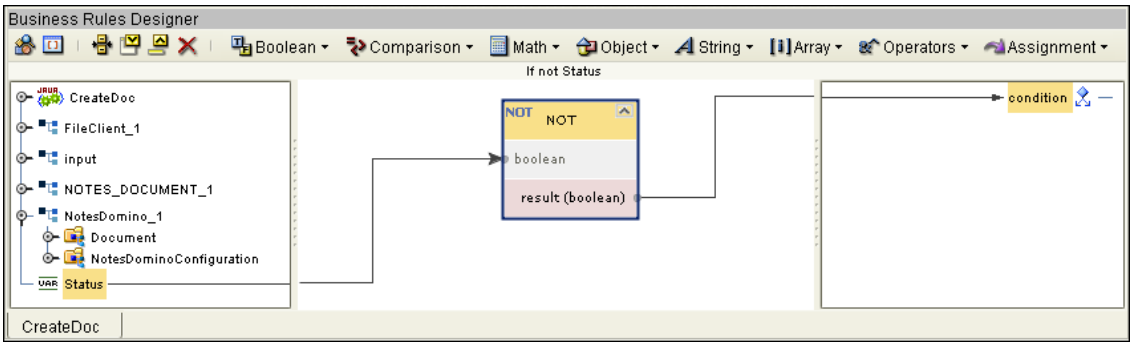
B Expand the **If** statement and select the **condition**.

C From the Business Rules Designer toolbar's **Boolean** menu, select **NOT.** The **NOT (**Boolean) operator dialog box appears.

D Map the **Status** variable in the left pane of the Business Rules Designer, to the **boolean** input node of the **NOT** operator dialog box.

E Map the **result (boolean)** output node of the **NOT** operator dialog box, to **condition** in the right pane of the Business Rules Designer (see Figure 12).

Figure 12 Business Rules Designer - CreateDoc Collaboration Business Rules



10 Create the **then** statement **Copy "Document Not Created" to FileClient_1.Text** rule:

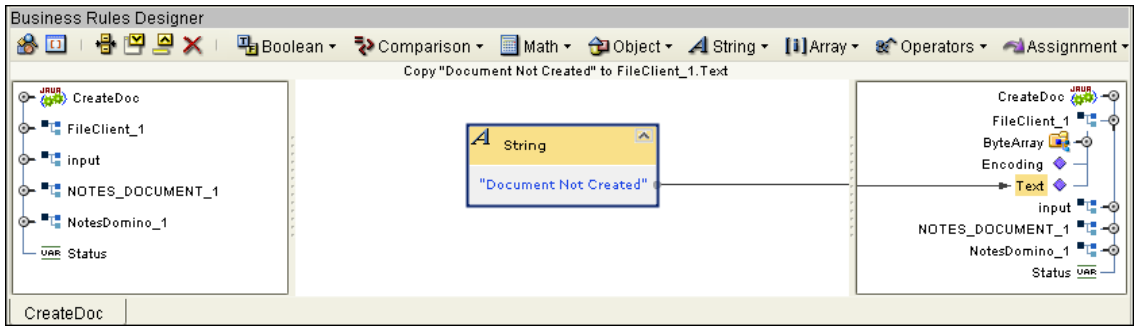
A Select **then** under the **If** statement.

B From the Business Rules toolbar, click the **rule** icon to add a new rule.

C From the Business Rules Designer toolbar's **String** menu, select **Literal String** icon. The **String** literal box appears. Enter **Document Not Created** as the value.

D Map the "**Document Not Created"** output node of the **String** literal box to **Text** under **FileClient_1** in the right pane of the Business Rules Designer (see Figure 13).

Figure 13 Business Rules Designer - CreateDoc Collaboration Business Rules



11 Create the **else** statement **Copy "Document Added" to FileClient_1.Text** rule:

A Select **else** under the **If** statement.

B From the Business Rules toolbar, click the **rule** icon to add a new rule.

C From the Business Rules Designer toolbar's **String** menu, select **Literal String** icon. The **String** literal box appears. Enter **Document Added** as the value.

D Map the "**Document Added"** output node of the **String** literal box to **Text** under **FileClient_1** in the right pane of the Business Rules Designer.

12 Create the **FileClient_1.write** Business Rule:

A Select the last **If** statement in the Business Rules tree.

B From the Business Rules toolbar, click the **rule** icon to add a new rule.

C Right-click **FileClient_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

D Select **write()**. The **write** method box appears.

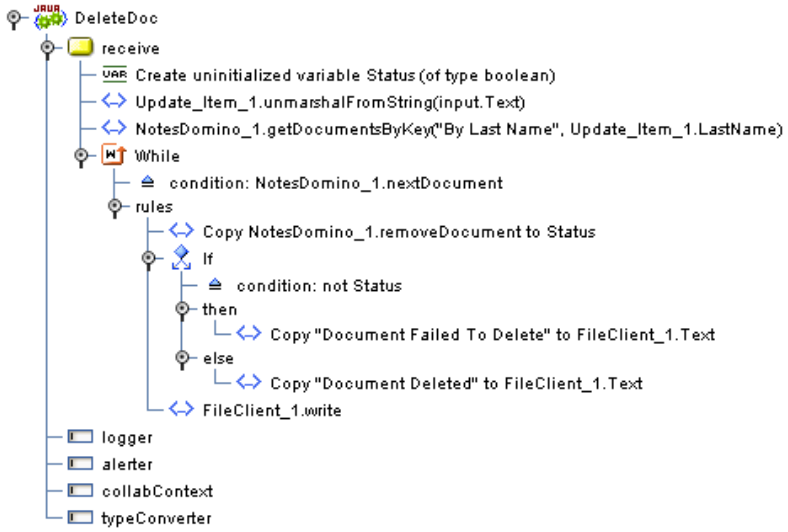
13 From the editor's toolbar, click **Validate** to check the Collaboration for errors.

14 Save your current changes to the Repository.

Creating the DeleteDoc Collaboration Business Rules

Be careful to open all nodes specified in the directions to connect to the correct items. The **DeleteDoc** Collaboration contains the Business Rule displayed in Figure 14.

Figure 14 DeleteDoc Collaboration Business Rules



To create the **DeleteDoc** Collaboration Business Rules do the following:

1 Open the Collaboration Editor to the **DeleteDoc** Collaboration by clicking the **DeleteDoc** tab if available, or by double-clicking the **DeleteDoc** Collaboration from the Enterprise Explorer tree.

2 Double-click the nodes in the Business Rules Designer panes to expand as needed.

3 Create the **Create uninitialized variable Status (of type boolean)** (variable) rule:

E From the Business Rules toolbar, click the **Local Variable** icon. The **Create Variable** dialog box appears.

F From the Create Variable dialog box. enter **Status** as the name, select **boolean** as the Type, and click **OK**. The boolean Status variable is added to the Business Rule tree.

4 Create the **UPDATE_ITEM_1.unmarshalFromString(Text)** Business Rule:

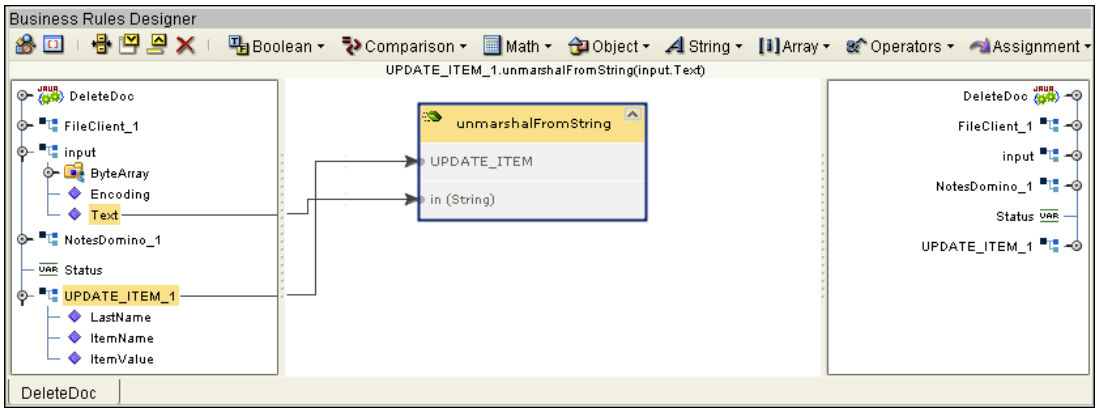
A From the Business Rules toolbar, click the **rule** icon to add a new rule.

B Right-click **UPDATE_ITEM_1** in the left pane of the Business Rules Designer, and select **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **unmarshalFromString(java.lang.String in)**. The **unmarshalFromString** method box appears.

D Map **Text** under **input** in the left pane of the Business Rules Designer, to the **in (String)** input node of the **unmarshalFromString** method box (see Figure 15).

Figure 15 Business Rules Designer - DeleteDoc Collaboration Business Rules



5 Create the **NotesDomino_1.getDocumentsByKey("By Last Name", UPDATE_ITEM_1.LastName)** rule:

A From the Business Rules toolbar, click the **rule** icon to add a new rule.

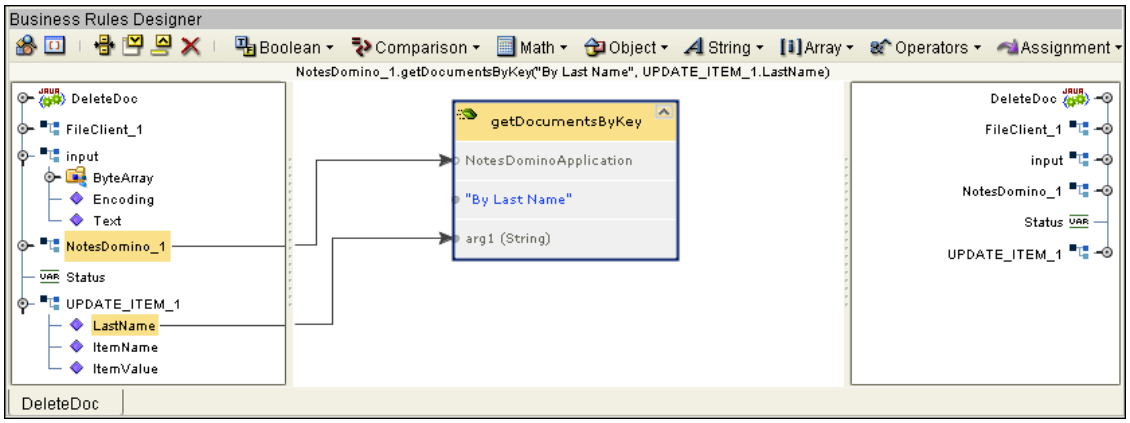
B Right-click **NotesDomino_1** in the left pane of the Business Rules Designer, and select **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **getDocumentsByKey(String arg0, String arg1)**. The **getDocumentsByKey** method box appears.

D Double-click the **arg0 (String)** input node of the **getDocumentsByKey** method UPDATE_ITEM_1box, and enter **By Last Name** as the String value.

E Map **LastName** under **UPDATE_ITEM_1** in the left pane of the Business Rules Designer, to the **arg1 (String)** input node of the **DocumentsByKey** method box (see **Figure 16 on page 37**).

Figure 16 Business Rules Designer - DeleteDoc Collaboration Business Rules



6 Create the **While** statement and **NotesDomino_1.nextDocument** condition:

A From the Business Rules toolbar, click the **While** icon to add a new **While** statement. Select the **condition** under the **While** statement.

B Right-click **NotesDomino_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **nextDocument()**. The next**Document** method box appears

D Map the **result (boolean)** output node of the **nextDocument** method box, to condition in the right pane of the Business Rules Designer.

7 To create the **Copy NotesDomino_1.removeDocument to Status** Business Rule do the following:

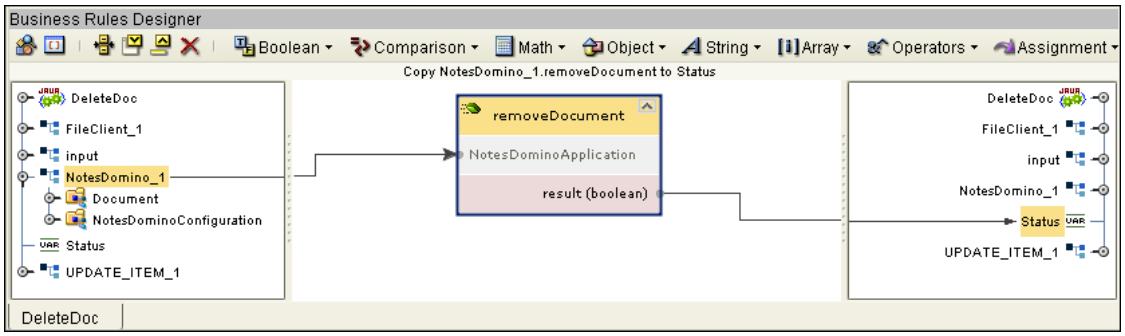
A Select **New Rule** under the **While** statement in the Business Rules widow's Business Rules tree.

B Right-click **NotesDomino_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **removeDocument()**. The **removeDocument** method box appears.

D Map the **result (boolean)** output node of the **removeDocument** method box to Status in the right pane of the Business Rules Designer (see Figure 17).

Figure 17 Business Rules Designer - DeleteDoc Collaboration Business Rules



8 Create the **If** statement and the **not Status** condition:

A From the Business Rules toolbar, click the **If** icon. An **If** statement is added to the Business Rules tree as a child to the **While** statement.

B Expand the **If** statement and select the **condition**.

C From the Business Rules Designer toolbar's **Boolean** menu, select **NOT.** The **NOT (**Boolean) operator dialog box appears.

D Map the **Status** variable in the left pane of the Business Rules Designer, to the **boolean** input node of the **NOT** operator dialog box.

E Map the **result (boolean)** output node of the **NOT** operator dialog box, to **condition** in the right pane of the Business Rules Designer

9 Create the **Copy "Document Failed To Delete" to FileClient_1.Text** Business Rule:

A Select **then** under the **if-then** statement.

B From the Business Rules toolbar, click the **rule** icon to add a new rule.

C From the Business Rules Designer's String menu, select **Literal String**. The **String** literal box appears. Enter the **Document Failed To Delete** as the String value.

D Map the "**Document Failed To Delete"** output node of the **String** literal method box, to **Text** under **FileClient_1** in the right pane of the Business Rules Designer.

10 To create the **Copy "Document Deleted" to FileClient_1.Text** Business Rule do the following:

A Select **else** under the **if-then** statement.

B From the Business Rules toolbar, click the **rule** icon to add a new rule.

C From the Business Rules Designer's String menu, select **Literal String**. The **String** literal box appears. Enter the **Document Deleted** as the String value.

D Map the "**Document Failed To Delete"** output node of the **String** literal method box, to **Text** under **FileClient_1** in the right pane of the Business Rules Designer.

11 To create the **FileClient_1.write** Business Rule do the following:

A Select the last **If** statement in the Business Rules tree.

B From the Business Rules toolbar, click the **rule** icon to add a new rule.

C Right-click **FileClient_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

D Select **write()**. The **write** method box appears.

12 From the editor's toolbar, click **Validate** to check the Collaboration for errors.

13 Save your current changes to the Repository.

From the Collaboration Editors toolbar, click the Source Code Mode icon to open the editors Java Source Editor. The Java code created from the previous steps appears as follows:

```
package prjNotesDomino_Sample_JCD;
```

```
public class CreateDoc
{

    public com.stc.codegen.logger.Logger logger;

    public com.stc.codegen.alerter.Alerter alerter;

    public com.stc.codegen.util.CollaborationContext collabContext;

    public com.stc.codegen.util.TypeConverter typeConverter;

    public void receive(
com.stc.connector.appconn.file.FileTextMessage input,
com.stc.connector.appconn.file.FileApplication FileClient_1,
dtd.Notes1876094286.NOTES_DOCUMENT NOTES_DOCUMENT_1,
com.stc.connector.notesdominoadapter.appconn.NotesDominoApplication
NotesDomino_1 )
        throws Throwable
    {
        boolean Status;
        ;
        NOTES_DOCUMENT_1.unmarshalFromString( input.getText() );
        NotesDomino_1.createDocument();
        if (NOTES_DOCUMENT_1.hasNOTES_ITEM()) {
            for (int i1 = 0; i1 < NOTES_DOCUMENT_1.countNOTES_ITEM();
i1 += 1) {
                if (NOTES_DOCUMENT_1.getNOTES_ITEM( i1
).hasItemName()) {
                    NotesDomino_1.getDocument().getItem( i1
).setItemName( NOTES_DOCUMENT_1.getNOTES_ITEM( i1 ).getItemName() );
                }
                if (NOTES_DOCUMENT_1.getNOTES_ITEM( i1
).hasItemValue()) {
                    NotesDomino_1.getDocument().getItem( i1
).setItemValue( NOTES_DOCUMENT_1.getNOTES_ITEM( i1 ).getItemValue()
);
                }
            }
        }
        Status = NotesDomino_1.saveDocument();
        if (!Status) {
            FileClient_1.setText( "Document not created" );
        } else {
            FileClient_1.setText( "Document Added" );
        }
        FileClient_1.write();
    }

}
```

Creating the RetrieveDoc Collaboration Business Rules

The **RetrieveDoc** Collaboration contains the Business Rule displayed in Figure 18.

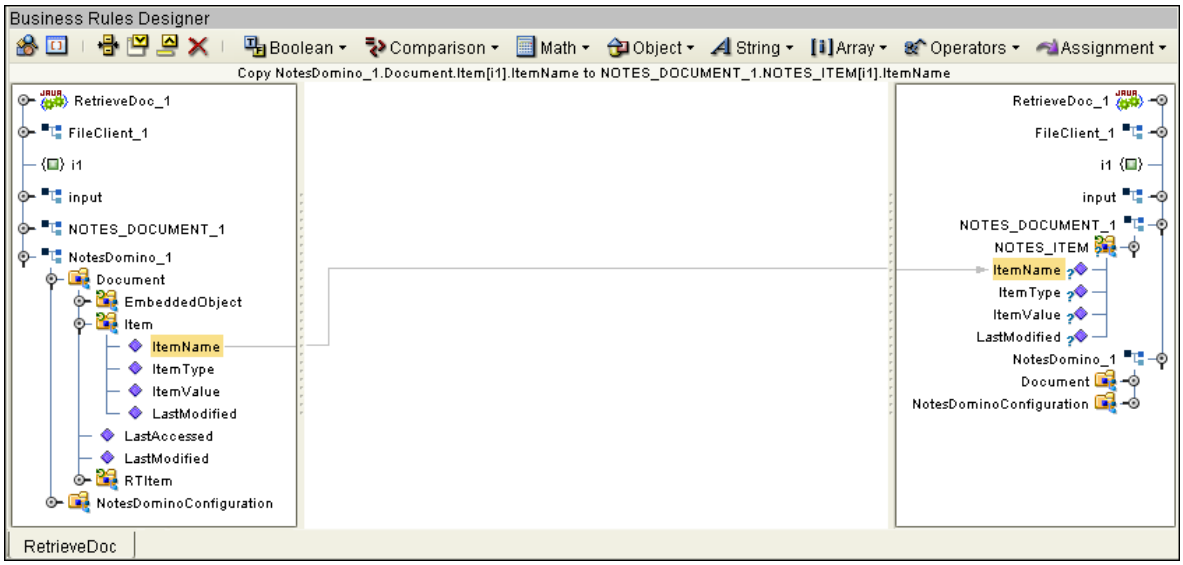
Figure 18 RetrieveDoc Collaboration Business Rules

To create the **RetrieveDoc** Collaboration Business Rules do the following:

1. Open the Collaboration Editor to the **RetrieveDoc** Collaboration by clicking the **RetrieveDoc** tab if available, or by double-clicking the **RetrieveDoc** Collaboration from the Enterprise Explorer tree.
2. Create the **NotesDomino_1.retrieveAllDocuments** Business Rule:
   - A Right-click **NotesDomino_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.
   - B Select **retrieveAllDocuments()**. The **retrieveAllDocuments** method box appears.
3. Create the **While** statement and the **NotesDomino_1.nextDocument** condition:
   - A From the Business Rules toolbar, click the **While** icon to add a new **While** statement.
   - B Select **condition** under the **While** statement.
   - C Right-click **NotesDomino_1** in the left pane of the Business Rules Designer, and select **Select a method to call** from the shortcut menu. The method selection menu appears.
   - D Select **nextDocument()**. The **nextDocument** method box appears.
   - E Map the **result (boolean)** output node of the **nextDocument** method box to **condition** in the right pane of the Business Rules Designer.
4. Create the **For Loop** and the **Copy NotesDomino_1.Document.Item[i1].ItemName to NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemName**.
   - A From the Business Rules tree, select **rules** under the **While** statement.

B Map **ItemName** under **NotesDomino_1 > Document > Item** in the left pane of the Business Rules Designer, to **ItemName** under **NOTES_DOCUMENT_1 > NOTES_ITEM** in the right pane of the Business Rules Designer (see Figure 19). The **For Loop**, it's components, and the **Copy NotesDomino_1.Document.Item[i1].ItemName to NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemName** rule are added to the Business Rules tree.

Figure 19 Business Rules Designer - RetrieveDoc Collaboration Business Rules



5 Create the **Copy String.valueOf(NotesDomino_1.Document.Item[i1].ItemType) to NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemType** rule.

A From the Business Rules toolbar, click the **Rule** icon to add a new rule to the Business Rules tree.

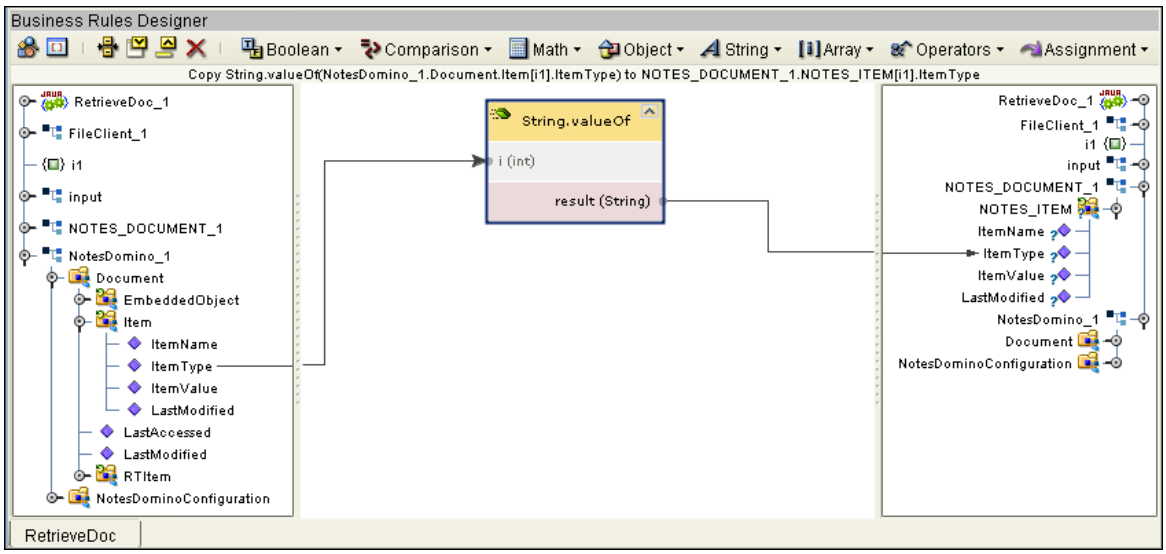
B From the Business Rules Designer toolbar, click the **Class Browser** icon. The **Class Browser** dialog box appears.

C From the **Class Browser** dialog box, select **String** as the class, select **valueOf(int i)** as the method, and click **Select**. The **String.valueOf** method box appears.

D Map **ItemType** under **NotesDomino_1 > Document > Item** in the left pane of the Business Rules Designer, to the **i (int)** input node of the **String.valueOf** method box.

E Map the **result (String)** output node of the **String.valueOf** method box, to **ItemType** under **NOTES_DOCUMENT_1 > NOTES_ITEM** in the right pane of the Business Rules Designer (see **Figure 20 on page 42**).

Figure 20 Business Rules Designer - RetrieveDoc Collaboration Business Rules



6 Create the **Copy String.valueOf(NotesDomino_1.Document.Item[i1].ItemValue) to NOTES_DOCUMENT_1.NOTES_ITEM[i1].ItemValue** rule.

A From the Business Rules toolbar, click the **Rule** icon to add a new rule to the Business Rules tree.

B From the Business Rules Designer toolbar, click the **Class Browser** icon. The **Class Browser** dialog box appears.

C From the **Class Browser** dialog box, select **String** as the class, select **valueOf(Object obj)** as the method, and click **Select**. The **String.valueOf** method box appears.

D Map **ItemValue** under **NotesDomino_1 > Document > Item** in the left pane of the Business Rules Designer, to the **obj (Object)** input node of the **String.valueOf** method box.

E Map the **result (String)** output node of the **String.valueOf** method box, to **ItemValue** under **NOTES_DOCUMENT_1 > NOTES_ITEM** in the right pane of the Business Rules Designer.

7 Create the **Copy NotesDomino_1.Document.Item[i1].LastModified to NOTES_DOCUMENT_1.NOTES_ITEM[i1].LastModified** rule.

A From the Business Rules toolbar, click the **Rule** icon to add a new rule to the Business Rules tree.

B Map **LastModified** under **NotesDomino_1 > Document > Item** in the left pane of the Business Rules Designer, to **LastModified** under **NOTES_DOCUMENT_1 > NOTES_ITEM** in the right pane of the Business Rules Designer.

8 Create the **Copy NOTES_DOCUMENT_1.marshalToString to FileClient_1.Text** rule.

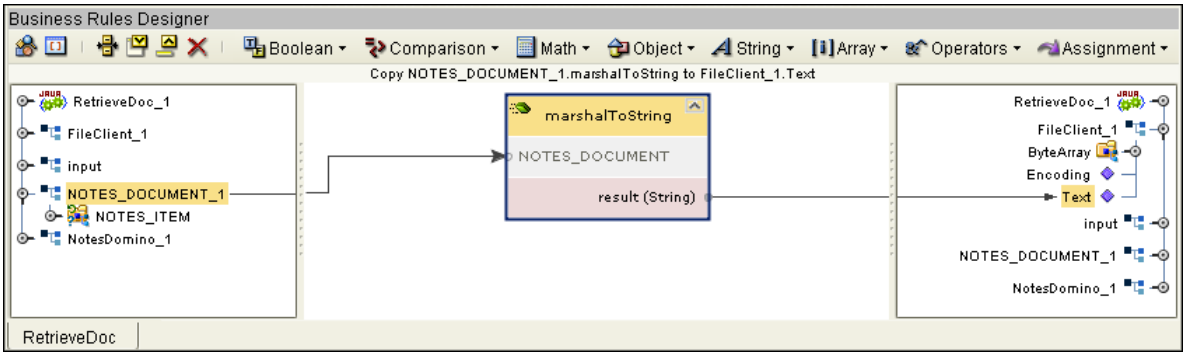
A From the Business Rules tree select the **For Loop**, and from the Business Rules toolbar, click the **rule** icon to add a new rule (as a sibling to the For Loop).

B Right-click **NOTES_DOCUMENT_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **marshalToString()**. The **marshalToString** method box appears.

D Map the **results (String)** output node of the **marshalToString** dialog box, to **Text** under **FileClient_1** in the right pane of the Business Rules Designer (see Figure 21).

Figure 21 Business Rules Designer - RetrieveDoc Collaboration Business Rules



9 Create the **FileClient_1.write** Business Rule:

A Select the last **If** statement in the Business Rules tree.

B From the Business Rules toolbar, click the **rule** icon to add a new rule.

C Right-click **FileClient_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

D Select **write()**. The **write** method box appears.

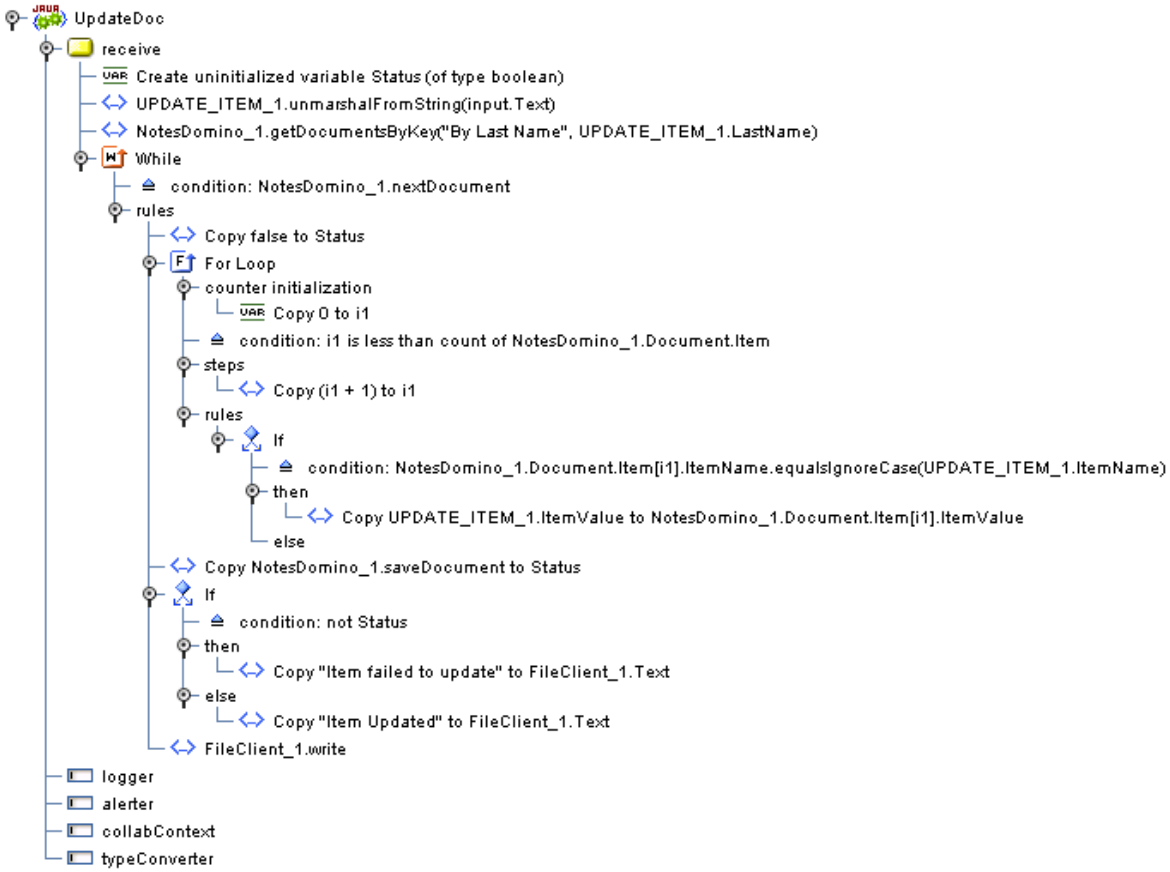
10 From the editor's toolbar, click **Validate** to check the Collaboration for errors.

11 Save your current changes to the Repository.

Creating the UpdateDoc Collaboration Business Rules

Be careful to open all nodes specified in the directions to connect to the correct items. The **UpdateDoc** Collaboration contains the Business Rule displayed in Figure 22.

Figure 22 UpdateDoc Collaboration Business Rules



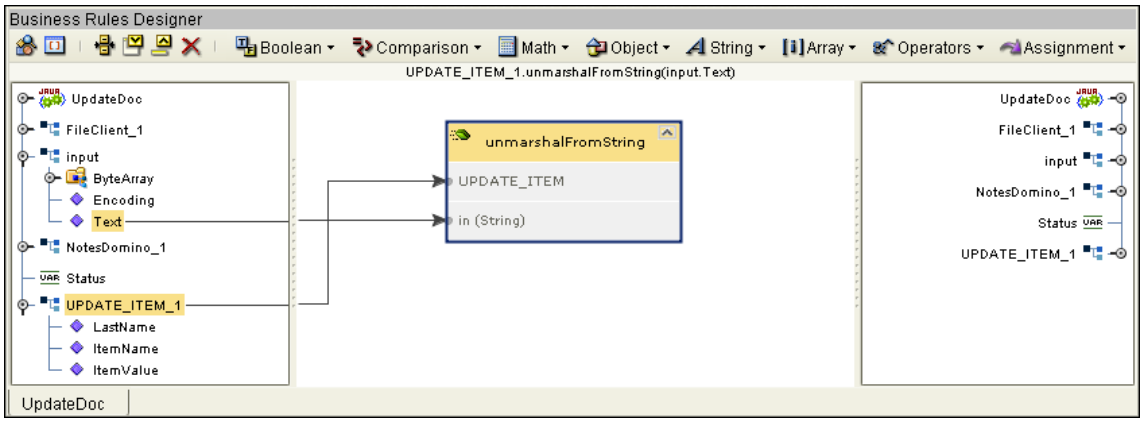
To create the UpdateDoc Collaboration Business Rules do the following:

1. Open the Collaboration Editor to the **UpdateDoc** Collaboration by clicking the **UpdateDoc** tab if available, or by double-clicking the **UpdateDoc** Collaboration from the Enterprise Explorer tree.
2. Create the **Create uninitialized variable Status (of type boolean)** variable:
   - A From the Business Rules toolbar, click the **Local Variable** icon. The **Create Variable** dialog box appears.
   - B Enter **Status** as the variable name, select **Primitive** and **boolean** as the type, and click **OK**. The boolean **Status** variable is added to the Business Rules Designer.
3. Create the **UPDATE_ITEM_1.unmarshalFromString(input.Text)** rule:
   - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
   - B Right-click **UPDATE_ITEM_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **unmarshalFromString(String in)**. The **unmarshalFromString** method box appears.

D Map **Text** under **input** in the left pane of the Business Rules Designer to the **in (String)** input node of the **unmarshalFromString** method box (see Figure 23).

Figure 23 Business Rules Designer - UpdateDoc Collaboration Business Rules



4 Create the **NotesDomino_1.getDocumentsByKey("By Last Name", UPDATE_ITEM_1.LastName)** rule:

A From the Business Rules toolbar, click the **rule** icon to add a new rule.

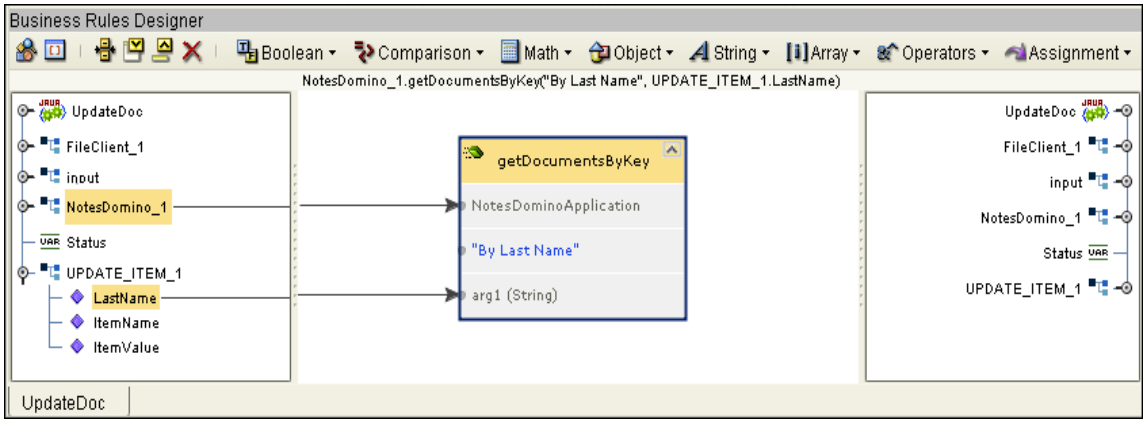
B Right-click **NotesDomino_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **getDocumentsByKey(String arg0, String arg1)**. The **getDocumentsByKey** method box appears.

D Double-click the **arg0 (String)** input node of the **getDocumentsByKey** method box and enter **By Last Name** as the String value.

E Map in **LastName** under **UPDATE_ITEM_1** the left pane of the Business Rules Designer, to the **arg1 (String)** input node of the **getDocumentsByKey** method box (see Figure 24).

Figure 24 Business Rules Designer - UpdateDoc Collaboration Business Rules



5 Create the **While** statement and **NotesDomino_1.nextDocument** Business Rule:

A From the Business Rules toolbar, click the **While** icon to add a new **While** statement. Select the **condition** under the **While** statement.

B Right-click **NotesDomino_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

C Select **nextDocument()**. The next**Document** method box appears.

6 Create the **Copy False to Status** Business Rule:

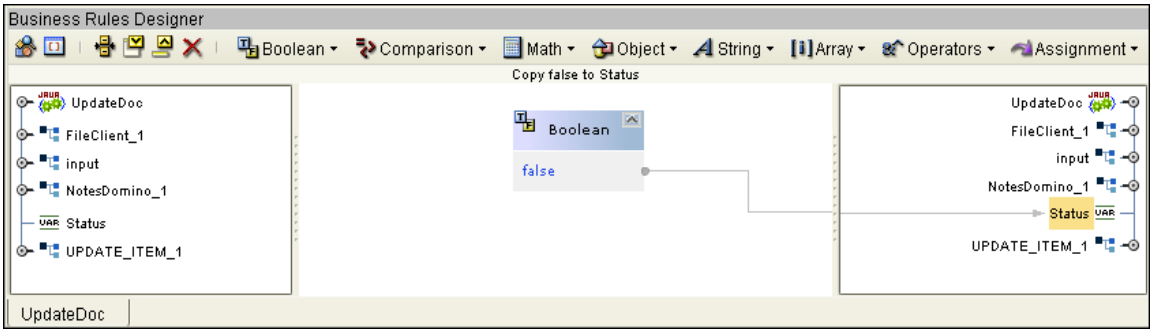
A From the Business Rules tree, select **rules** under the **While** statement.

B From the Business Rules toolbar, click the **Rule** icon. A new rule is added to the Business Rules tree under the **While** statement rules.

C From the Business Rules Designer toolbar's **Boolean** menu, select **False**. The **Boolean** method box appears

D Map the **false** output node of the **Boolean** method box to **Status** (variable) in the right pane of the Business Rules Designer (see Figure 25).

Figure 25 Business Rules Designer - UpdateDoc Collaboration Business Rules



7 Create the **For Loop** and the **Copy 0 to i1** counter initialization:

A From the Business Rules toolbar, click the **For Loop** icon. A new **For Loop** is added to the Business Rules tree under the **While** statement.

B From the Business Rules tree, select the For Loop's **counter initialization**. From the Business Rules toolbar, click the **Local Variable** icon. The **Create Variable** dialog box appears.

C From the Create Variable dialog box, enter **i1** as the name, and select **Primitive**, **int**, as the type. Click **OK**.

8 Create the **For Loop** condition: **i1 is less than count of NotesDomino_1.Document.Item**:

A From the Business Rules tree, select the For Loop's **condition**.

B From the Business Rules Designer, right-click **Document** under **NotesDomino_1**, and select **Select method to call** from the shortcut menu. The method selection box appears.

C Select **countItem**. The **countItem** method box appears.

D From the Business Rules Designer toolbar's Comparison menu, select **Less than**. The **Less than** method box appears.

E Map the **i1** field in the left pane of the Business Rules Designer, to the **number1** input node of the **Less than** method box.

F Map the **result(int)** output node of the **countItem** method box, to the **number2** input node of the **Less than** method box.

G Map the **result(boolean)** output node of the **Less than** method box to condition in the right pane of the Business Rules Designer.

9 Create the **For Loop** steps **Copy (i1 + 1) to i1** rule:

A From the Business Rules tree, select the For Loop's **steps** node.

B From the Business Rules Designer toolbar's **Math** menu, select **Add**. The **Add** method box appears.

C Map the **i1** field in the left pane of the Business Rules Designer, to the **value1** input node of the **Add** method box.

D Double-click the **value2** field of the **Add** method box, and enter **1** as the value.

E Map the **result** output node of the **Add** method box, to the **i1** field in the right pane of the Business Rules Designer.

10 Create the **If-then** statement under the For Loop rules:

A From the Business Rules tree, select **rules** under the For Loop.

B From the Business Rules toolbar, click the **If-then** icon to add an **If-then** statement under **For Loop > rules**.

11 Create the **If** statement condition: **NotesDomino_1.Document.Item[i1].ItemName.equalsIgnoreCase(UPDATE_ITEM_1.ItemName)**:

A From the Business Rules tree, select the If statement's **condition**.

B Right-click **ItemName** under **NotesDomino_1 > Document > Item** in the left pane of the Business Rules Designer and select **Select method to call** from the shortcut menu. The method selection menu appears.

C Select **equalsIgnoreCase(String anotherString)**. The **equalsIgnoreCase** method box appears.

D Map **ItemName** under **UPDATE_ITEM_1** in the left pane of the Business Rules Designer, to the **anotherString (String)** input node of the **equalsIgnoreCase** method box.

E Map the **result (boolean)** output node of the **equalsIgnoreCase** method box, to condition in the right pane of the Business Rules Designer.

12 Create the **Copy UPDATE_ITEM_1.ItemValue to NotesDomino_1.Document.Item[i1].ItemValue** rule under the **If-then** statement:

A From the Business Rules tree, select **then** under the **If** statement.

B From the Business Rules toolbar, click the **rule** icon to add a new rule under under **then**.

C Map **ItemValue** under **UPDATE_ITEM_1** in the left pane of the Business Rules Designer, to **ItemValue** under **NotesDomino_1 > Document > Item** in the right pane of the Business Rules Designer.

13 Create the **Copy NotesDomino_1.saveDocument to Status** rule:

A From the Business Rules tree, select the **For Loop**.

B From the Business Rules toolbar, click the **Rule** icon (to add a new rule as a sibling to the **For Loop**).

C Right-click NotesDomino_1 in the left pane of the Business Rules Designer, and select **Select method to call** from the shortcut menu. The method selection menu appears.

D Select **saveDocument()**. The **saveDocument** method box appears.

E Map the **result(boolean)** output node of the saveDocument method box, to **Status** (variable) in the right pane of the Business Rules Designer.

14 Create the **If-then** statement and **not Status** condition:

A From the Business Rules toolbar click the **If-then** statement icon. An **if-then** statement is added as a child to the **While** statement on the Business Rules tree.

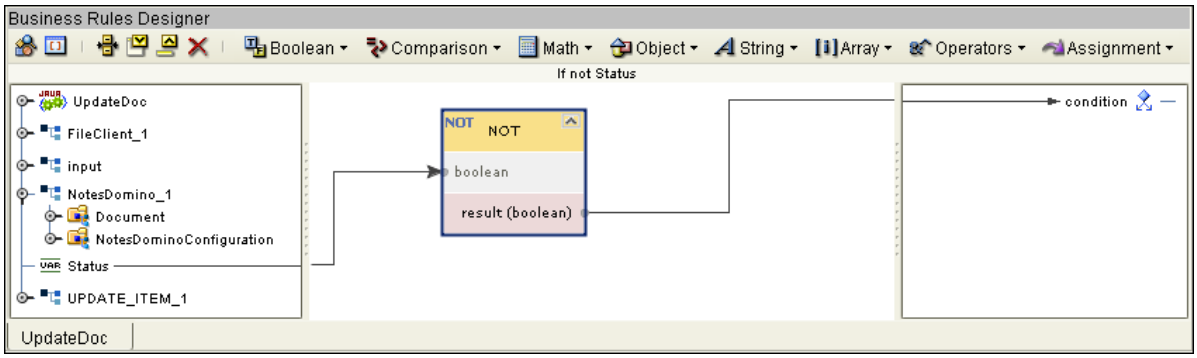
B From the Business Rules tree, select the **condition** under the new If-then statement.

C From the Business Rules Designer toolbar's Boolean menu, select **NOT**. The **Not** method box appears.

D Map **Status** (variable) in the left pane of the Business Rules Designer, to the **boolean** input node of the **Not** method box.

E Map the **result(boolean)** output node of the **Not** method box, to **condition** in the right pane of the Business Rules Designer (see Figure 26).

Figure 26 Business Rules Designer - UpdateDoc Collaboration Business Rules



15 Create the **Copy "Item failed to update" to FileClient_1.Text** rule under the **If-then** statement:

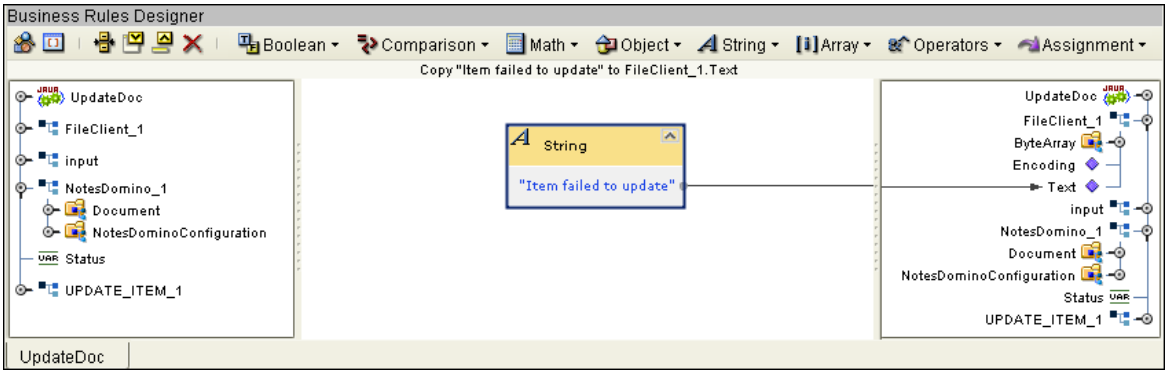
A From the Business Rules tree, select **then** under the last **If-then** statement.

B From the Business Rules toolbar, click the **Rule** icon to add a new rule.

C From the Business Rules Designer toolbar's String menu, select **Literal String**. The **String** literal box appears. Enter **Item failed to update** as the String value.

D Map the "**Item failed to update**" output node of the **String** literal box, to **Text** under FileClient_1 in the right pane of the Business Rules Designer (see Figure 27).

Figure 27 Business Rules Designer - UpdateDoc Collaboration Business Rules



16 Create the **Copy "Item Updated" to FileClient_1.Text** rule under the **If-then-else** statement:

A From the Business Rules tree, select **else** under the last **If-then** statement.

B From the Business Rules toolbar, click the **Rule** icon to add a new rule.

C From the Business Rules Designer toolbar's String menu, select **Literal String**. The **String** literal box appears. Enter **Item Updated** as the String value.

D Map the "**Item Updated**" output node of the **String** literal box, to **Text** under FileClient_1 in the right pane of the Business Rules Designer.

17 Create the **FileClient_1.write** Business Rule:

A Select the last **If** statement in the Business Rules tree.

B From the Business Rules toolbar, click the **rule** icon to add a new rule (as a sibling to the **If** statement).

C Right-click **FileClient_1** in the left pane of the Business Rules Designer, and click **Select a method to call** from the shortcut menu. The method selection menu appears.

D Select **write()**. The **write** method box appears.

18 From the editor's toolbar, click **Validate** to check the Collaboration for errors.

19 Save your current changes to the Repository.

For more information on how to create Business Rules using the Collaboration Editor see the *Sun SeeBeyond eGate Integrator User's Guide*.

4.5.5 Creating the Connectivity Maps

The Connectivity Map provides a canvas for assembling and configuring a Project's components. This sample utilizes four Connectivity Maps.

1 From the Project Explorer tree, right-click the new **prjNotesDomino_Sample_JCD** Project and select **New** > **Connectivity Map** from the shortcut menu.

2 The New Connectivity Map appears and a node for the Connectivity Map is added under the Project on the Project Explorer tree labeled **CMap1**. Rename the Connectivity Map to **CreateDoc_CM**.

3 Repeat this process to create the following three additional Connectivity Maps.

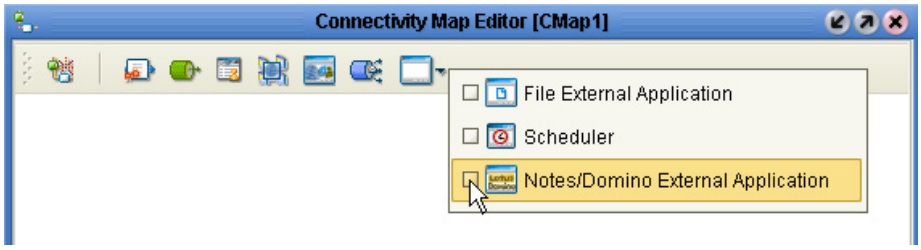
- DeleteDoc_CM
- RetrieveAll_CM
- UpdateDoc_CM

Selecting the External Applications

You now have four Connectivity Maps open, accessed by selecting the associated tabs at the bottom of the Connectivity Map screen. Select the **CreateDoc_CM** tab to display the **CreateDoc_CM** Connectivity Map.

The icons on the Connectivity Map toolbar represent the available components used to populate the Connectivity Map canvas. In a Connectivity Map, eWays are associated with External Applications. For example, to establish a connection to Lotus Notes/Domino, you must first select Notes/Domino as an External Application to use in your Connectivity Map (see Figure 28).

Figure 28 Connectivity Map - External Applications



1 Click the **External Application** icon on the Connectivity Map toolbar,

2 Select the External Applications required by your Project (for this sample, the **Notes/Domino** and **File** External Applications). Icons representing the selected External Applications are added to the Connectivity Map toolbar.

Populating the Connectivity Maps

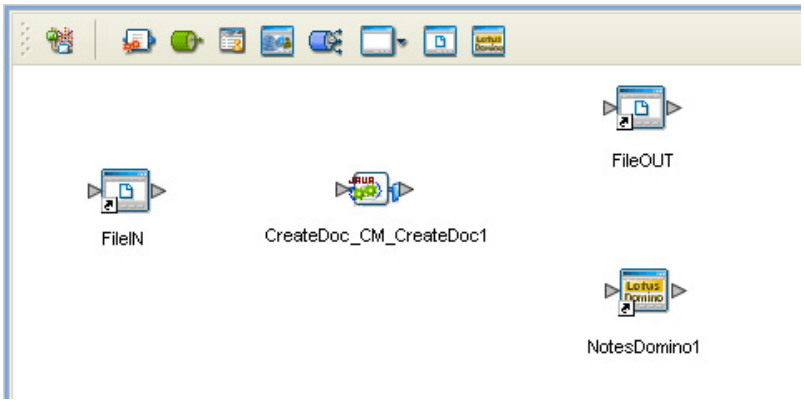
Project components to the Connectivity Maps by dragging the component icons from the Connectivity Map toolbar or existing components from the Project Explorer tree.

1 Populate the **CreateDoc_CM** Connectivity Map with the necessary components by dragging the following objects onto the Connectivity Map canvas, as displayed in **Figure 29 on page 51**:

- File External Application (2 for this sample)
- Notes/Domino External Application (1 for this sample)

2 Drag the **CreateDoc** Collaboration from the Project Explorer tree onto the Connectivity Map. The Collaboration becomes the **CreateDoc_CM_DeleteDoc1** service.

Figure 29 Connectivity Map with Components



3 Rename the File eWays as follows:

- **File1** to **FileIN**
- **File2** to **FileOUT**

as displayed in Figure 29.

DeleteDoc_CM, RetrieveAll_CM, UpdateDoc_CM

Open and populate the **DeleteDoc** Connectivity Map.

1. From the Project Explorer tree, drag and drop the **FileIN**, **FileOUT**, and **NotesDomino1** External Applications onto the Connectivity Map canvas, using the same pattern displayed in Figure 29.
2. Drag the **DeleteDoc** Collaboration from the Project Explorer tree onto the Connectivity Map. This Collaboration becomes the **DeleteDoc_CM_DeleteDoc1** service.
3. Follow these same steps to populate the **RetrieveAll_CM** and **UpdateDoc_CM** Creativity Maps.

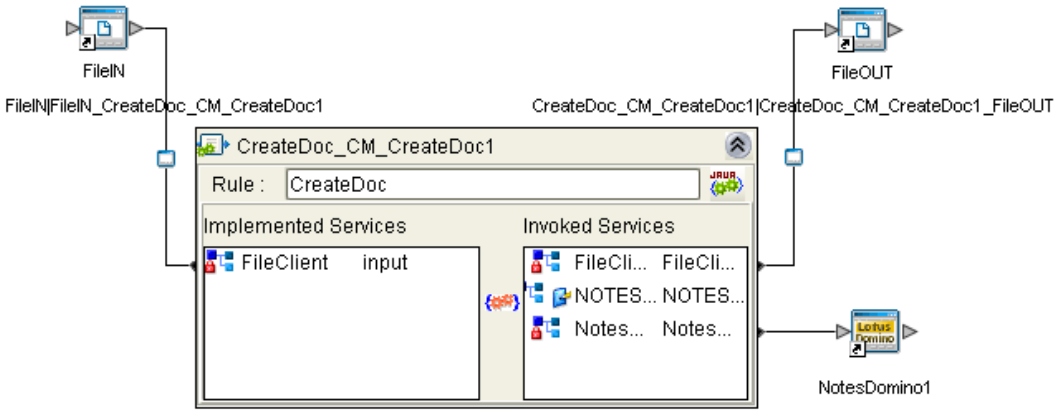
4.5.6 Binding the eWay Components

After the Collaborations have been written, and the Connectivity Maps have been populated, the components can be associated (create bindings).

Binding the CreateDoc_CM Components

1. Open the **CreateDoc_CM** Connectivity Map.
2. From the Connectivity Map canvas, double-click **CreateDoc_CM_DeleteDoc1** service. The **CreateDoc_CM_DeleteDoc1** binding dialog box appears using the **CreateDoc** Rule.
3. From the **CreateDoc_CM_DeleteDoc1** binding dialog box, map **FileClient Input** (under Implemented Services) to the inbound **FileIN** External Application. To do this, click on the binding dialog box's **FileClient** node, and drag your cursor to the output node of the **FileIN** External Application. A link now connects the two nodes.
4. Map **FileClient_1** (under Invoked Services) to the outbound **FileOUT** External Application.
5. Map **NotesDomino_1** (under Invoked Services) to the **NotesDomino1** External Application (see Figure 30).

Figure 30 Connectivity Map - Associating (Binding) the Project's Components



6. Minimize the **CreateDoc_CM_DeleteDoc1** binding dialog box and save your changes to your Repository.

Binding the DeleteDoc_CM Components

1. Open the **DeleteDoc_CM** Connectivity Map.
2. Double-click the **DeleteDoc_CM_DeleteDoc1** Service. The **DeleteDoc_CM_DeleteDoc1** binding dialog box appears using the **DeleteDoc** Rule.
3. From the binding dialog box, map **FileClient input** (under Implemented Services) to the **FileIN** External Application.
4. Map **FileClient_1** (under Invoked Services) to the **FileOUT** External Application.
5. Map **NotesDomino** (under Invoked Services) to the **NotesDomino1** External Application.
6. Minimize the **DeleteDoc_CM_DeleteDoc1** binding dialog box, and save your current changes.

Binding the RetrieveAll_CM Components

1. Open the **RetrieveAll_CM** Connectivity Map.
2. Double-click the **RetrieveAll_CM_RetrieveAllDocs1** Service. The **RetrieveAll_CM_RetrieveAllDocs1** binding dialog box appears using the **RetrieveAllDocs** Rule.
3. From the **RetrieveAll_CM_RetrieveAllDocs1** binding dialog box, map **FileClient input** (under Implemented Services) to the **FileIN** External Application.
4. Map **NotesDomino_1** (under Invoked Services) to the **NotesDomino1** External Application.
5. Map **FileClient_1** (under Invoked Services) to the **FileOUT** External Application.
6. Minimize the **RetrieveAll_CM_RetrieveAllDocs1** binding dialog box, and save your current changes.

Binding the UpdateDoc_CM Components

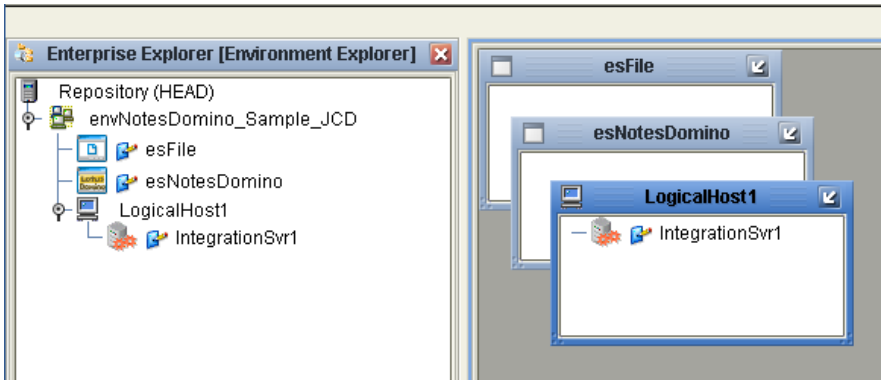
1. From the Project Explorer, double-click **UpdateDoc_CM** to display the Connectivity Map.
2. Double-click the **UpdateDoc_CM_UpdateDoc1** Service. The **UpdateDoc_CM_UpdateDoc1** binding dialog box appears using the **UpdateDoc** Rule.
3. From the **UpdateDoc_CM_UpdateDoc1** binding dialog box, map **FileClient input** (under Implemented Services) to the **FileIN** External Application.
4. Map **NotesDomino_1** (under Invoked Services) to the **NotesDomino1** External Application.
5. Map **FileClient_1** (under Invoked Services) to the **FileOUT** External Application.
6. Minimize the **UpdateDoc_CM_UpdateDoc1** binding dialog box, and save your current changes.

4.5.7 Creating an Environment

Environments include the external systems, Logical Hosts, integration servers and message servers used by a Project, and contain the configuration information for these components. Environments are created using the Enterprise Designer's Environment Editor.

1. From the Enterprise Designer's Enterprise Explorer, right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.
2. Rename the new Environment to **envNotesDomino_Sample_JCD**.
3. Right-click **envNotesDomino_Sample_JCD** and select **New File External System**. Name the External System **esFile** and click **OK**. **esFile** is added to the Environment Editor.
4. Right-click **envNotesDomino_Sample_JCD** and select **New NotesDomino External System**. Name the External System **esNotesDomino** and click **OK**. **esNotesDomino** is added to the Environment Editor.
5. Right-click **envNotesDomino_Sample_JCD** and select **New Logical Host**. **LogicalHost1** is added to the Environment Editor.
6. From the Environment Explorer tree, right-click **LogicalHost1** and select **New > Sun SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under **LogicalHost1** (see **Figure 31 on page 54**).

Figure 31 Environment Editor - envNotesDomino_Sample_JCD.



7. Save your changes to your Repository.

4.5.8 Add the Required JAR File to the Logical Host.

Prior to deploying your Lotus Notes/Domino eWay Project, copy the appropriate Jar file to your Logical Host. For directions, see **"Adding Lotus Notes/Domino JAR files" on page 14**

Note: *Add C:\Program Files\Lotus\Domino into Path and Classpath as to include the dll files else a java.library.path not found exception is generated while deploying the project.*

4.5.9 Configuring the Integration Server

You must set your Sun SeeBeyond Integration Server Password property before deploying your Project.

1. From the Environment Explorer, right-click **IntegrationSvr1** under your **Logical Host**, and select **Properties** from the shortcut menu. The Integration Server Properties Editor appears.
2. Click the **Password** property field under **Sun SeeBeyond Integration Server Configuration**. An ellipsis appears in the property field.
3. Click the ellipsis. The **Password Settings** dialog box appears. Enter **STC** as the **Specific Value** and as the **Confirm Password**, and click **OK**.

Click **OK** to accept the new property and close the Properties Editor.

4.5.10 Configuring the eWays

The **prjNotesDomino_Sample_JCD** Project contains several component eWays. The File eWay's properties are configured from the Connectivity Map and the Environment Explorer. The Lotus Notes/Domino eWay properties are set from the Environment Explorer tree (the Notes/Domino eWay contains no Connectivity Map properties).

Configuring the File eWay Connectivity Map Properties

Connectivity Map properties are specific to each component eWay. To open the Connectivity Map Properties Editor for each component File eWay, open the appropriate Connectivity Map, and double-click the node in the link between the service and the File External Application. The Properties Editor appears with the configuration for that specific eWay.

Configure your Project eWays as follows:

1. From the **CreateDoc_CM** Connectivity Map, Double-click the inbound **FileIN** eWay.
2. The **Properties Editor** opens to the inbound **FileIN** eWay properties.
3. Click the **Input file name** property field and enter **NotesCreate.txt** as the property value. Click OK to save your settings and close the Properties Editor.
4. Double-click the outbound **FileOUT** eWay.
5. Click the **Output file name** property field and enter **NotesCreate.dat** as the property value. Click **OK** to save your settings and close the Properties Editor.
6. Following the same steps presented above, modify the remaining File eWay Connectivity Map properties as follows:

- **DeleteDoc_CM** Connectivity Map File eWay Properties
  - FileIN: Input file name - NotesDelete.txt
  - FileOUT: Output file name - NotesDelete.dat
- **ReceiveAll_CM** Connectivity Map File eWay Properties

- FileIN: Input file name - NotesRetrieve.txt
- FileOUT: Output file name - NotesRetrieve.dat

- **UpdateDoc_CM** Connectivity Map File eWay Properties
  - FileIN: Input file name - NotesUpdate.txt
  - FileOUT: Output file name - NotesUpdate.dat

Configure the File eWay Environment Properties

Environment properties are specific to all of the component eWays assigned to an External System in the Environment. To configure the File eWay Environment properties, do the following:

1. From the **Environment Explorer** tree, right-click the File External System (**esFile** in this sample), and select **Properties**. The Properties Editor opens to the File eWay Environment configuration.
2. Modify the File eWay Environment configuration properties for your system, including the settings in Table 6, and click **OK**.

Table 6 File eWay Environment Explorer Properties

File eWay Environment Properties	
Inbound File eWay - Set as directed, otherwise use the default settings	
Parameter Settings > Directory	C:/temp/LotusNotes *or the input directory of your choice*
Outbound File eWay - Set as directed, otherwise use the default settings.	
Parameter Settings > Directory	C:/temp/LotusNotes *or the output directory of your choice*

Configuring the Lotus Notes/Domino eWay Properties

The Lotus Notes/Domino eWay only contains Environment properties. To configure the Lotus Notes/Domino eWay Environment properties, do the following:

1. From the **Environment Explorer** tree, right-click the Lotus Notes/Domino eWay External System (**esNotesDomino** in this sample), and select **Properties** from the shortcut menu. The Properties Editor appears.
2. Modify the Lotus Notes/Domino eWay Environment properties for your system, including the settings in Table 7, and click **OK**.

Table 7 Notes/Domino Environment Properties

Lotus Notes/Domino eWay Environment Properties	
Environment Configuration Settings - Set as directed, otherwise use the default settings	
Database Type	*Specify **Remote** or **Local**, according to your system*
Notes/Domino Server	Specify a NotesDomino server

Table 7 Notes/Domino Environment Properties

Lotus Notes/Domino eWay Environment Properties	
Notes/Domino Database	Specify a database
Notes/Domino User	Specify a NotesDomino server authorized user login
Password	Specify a NotesDomino server password for the specified user

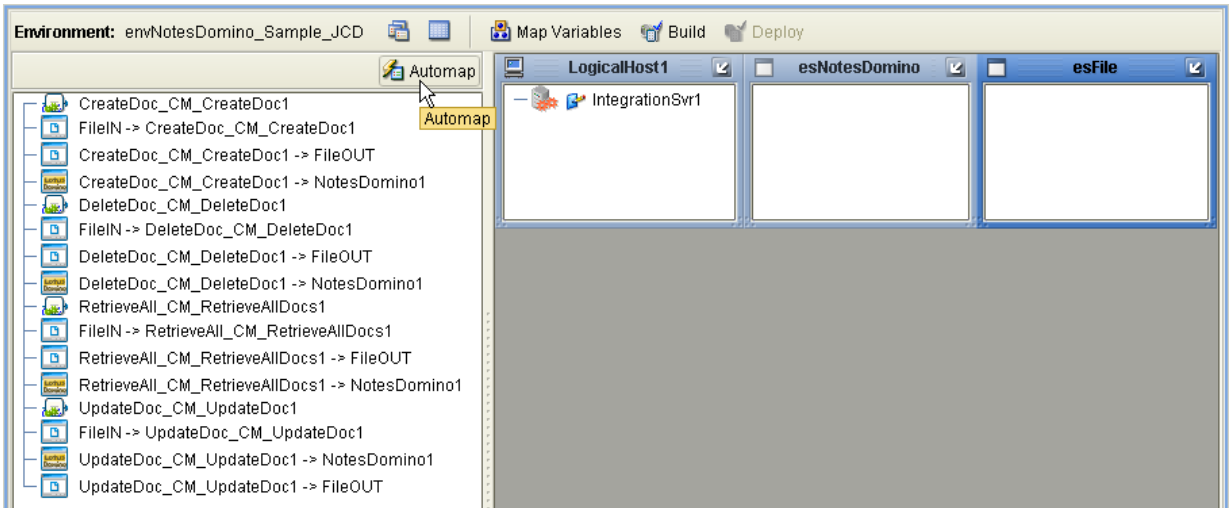
For more information on the Lotus Notes/Domino eWay properties and the Properties Editor, see **Lotus Notes/Domino eWay Properties** on page 18.

4.5.11 Creating the Deployment Profile

A Deployment Profile is used to assign Collaborations and message destinations to the integration server and message server. Deployment profiles are created using the Deployment Editor.

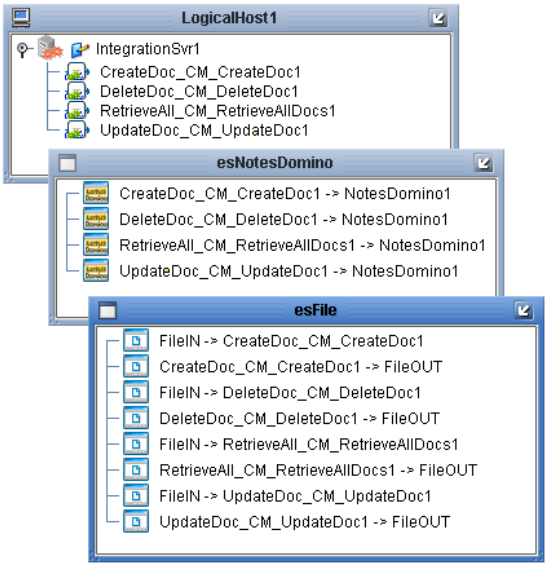
1. From the Enterprise Explorer's Project Explorer, right-click your Project and select **New** > **Deployment Profile**. The **Create Deployment Profile** dialog box appears.
2. Enter the name of your Project Deployment Profile (for this sample **dpNotesDomino_Sample_JCD**). Select **envNotesDomino_Sample_JCD** as the Environment and click **OK**. The **Deployment Editor** appears.
3. From the Deployment Editor toolbar, click the **Automap** icon (see **Figure 32 on page 57**).

Figure 32 Environment Editor - dpNotesDomino_Sample_JCD.



4. The Project's components are automatically mapped to their External Systems as displayed in Figure 33. Save your changes to the Repository.

Figure 33 Deployment - dpNotesDomino_Sample_JCD.



4.5.12 Creating and Starting the Domain

To build and deploy your Project, you must first create a domain. A domain is an instance of a Logical Host. After the domain is created, the Project is built and then deployed.

Create and Start the Domain

1. Navigate to your **<*JavaCAPS51*>\logicalhost** directory (where <*JavaCAPS51*> is the location of your Sun Java Composite Application Platform Suite installation.
2. Double-click the **domainmgr.bat** file. The **Domain Manager** appears.
3. If you have already created a domain, select your domain in the Domain Manager and click the **Start an Existing Domain** button. Once your domain is started, a green check mark indicates that the domain is running.
4. If there are no existing domains, a dialog box indicates that you can create a domain now. Click **Yes**. The **Create Domain** dialog box appears.
5. Make any necessary changes to the **Create Domain** dialog box and click **Create**. The new domain is added to the Domain Manager. Select the domain and click the **Start an Existing Domain** button. Once your domain is started, a green check mark indicates that the domain is running.

For more information about creating and managing domains see the *Sun SeeBeyond eGate™ Integrator User's Guide* and the *Sun SeeBeyond eGate™ Integrator System Administration Guide*.

4.5.13 Building and Deploying the Project

The Build process compiles and validates the Project's Java files and creates the Project EAR file.

Build the Project

1. From the Deployment Editor toolbar, click the **Build** icon.
2. If there are any validation errors, a **Validation Errors** pane will appear at the bottom of the Deployment Editor and displays information regarding the errors. Make any necessary corrections and click **Build** again.
3. After the Build has succeeded you are ready to deploy your Project.

Deploy the Project

1. From the Deployment Editor toolbar, click the **Deploy** icon. Click **Yes** when the **Deploy** prompt appears.
2. A message appears when the Project is successfully deployed. You can now test your sample.

Note: *Projects can also be deployed from the Enterprise Manager. For more information about using the Enterprise Manager to deploy, monitor, and manage your Projects, see the Sun SeeBeyond eGate™ Integrator System Administration Guide.*

4.5.14 Running the Sample

Be sure to create your Input and Output directories, matching the directory names you designated in your File eWay properties. The sample Project comes with the following input and example files:

- Sample input files:
  - NotesCreate.txt.~in
  - NotesDelete.txt.~in
  - NotesRetrieve.txt.~in
  - NotesUpdate.txt.~in
- Sample output example files:
  - NotesCreate.dat
  - NotesDelete.dat
  - NotesRetrieve.dat
  - NotesUpdate.dat

To run your deployed sample, from your configured input directory, paste (or rename) the sample input file to trigger the eWay. Run the sample in the following order:

1. RetrieveDoc
   A. Copy the **NotesRetrieve.~in** sample data file into the input directory.
   B. Change the file extension to match the file name set by the File eWay **Input file name** parameter for the **RetrieveALL_CM > FileIN** eWay.
   C. The trigger file is picked up by the eWay. The output file containing the retrieved file data appears in the output directory.

2 CreateDoc

A Copy the **NotesCreate.~in** sample data file into the input directory.

B Change the file extension to match the file name set by the Inbound File eWay **Input file name** parameter for the **CreateDoc_CM > FileIN** eWay.

C The input file is picked up by the eWay. The new document is created and added to the Lotus Notes/Domino database. The output file, containing the message "Document Added" is sent to the output directory.

3 UpdateDoc

A Copy the **NotesDelete.~in** sample data file into the input directory.

B Change the file extension to match the file name set by the Inbound File eWay **Input file name** parameter for the **UpdateDoc_CM > FileIN** eWay.

C The input file is picked up by the eWay. The document created by CreateDoc is updated. The output file, containing the message "Item Updated" is sent to the output directory.

4 DeleteDoc

A Copy the **NotesDelete.~in** sample data file into the input directory.

B Change the file extension to match the file name set by the Inbound File eWay **Input file name** parameter for the **DeleteDoc_CM > FileIN** eWay.

C The file is picked up by the eWay. The Created/Updated document is deleted. The output file, containing the message "Document Deleted" is sent to the output directory.

4.6 Creating the prjNotesDomino_EMailSample_JCD Project

The following pages provide directions that demonstrate how the sample Project and its components are created manually.

The prjNotesDomino_EMailSample_JCD Project contains one sample process: jcdMail, that demonstrates how the Lotus Notes/Domino eWay is used to send a document to recipients through the Lotus Notes/Domino server.

Note: *A Lotus Notes/Domino JAR file, Notes.jar or NSCO.jar, must be added to the Enterprise Designer prior to building your Lotus Notes/Domino eWay Project, and to your Logical Host prior to deploying your eWay Project. For more information see* **"Adding Lotus Notes/Domino JAR files" on page 14**

To manually create the prjNotesDomino_EMailSample_JCD Project, do the following:

4.6.1 Create a New Project

The first step is to create a new Project in the SeeBeyond Enterprise Designer.

1. From the Project Explorer tree, right-click the Repository and select **New Project**. A new Project (**Project1**) appears on the Project Explorer tree.
2. Rename the Project (for this sample, **prjNotesDomino_EMailSample_JCD**).

4.6.2 Create the Collaboration Definition

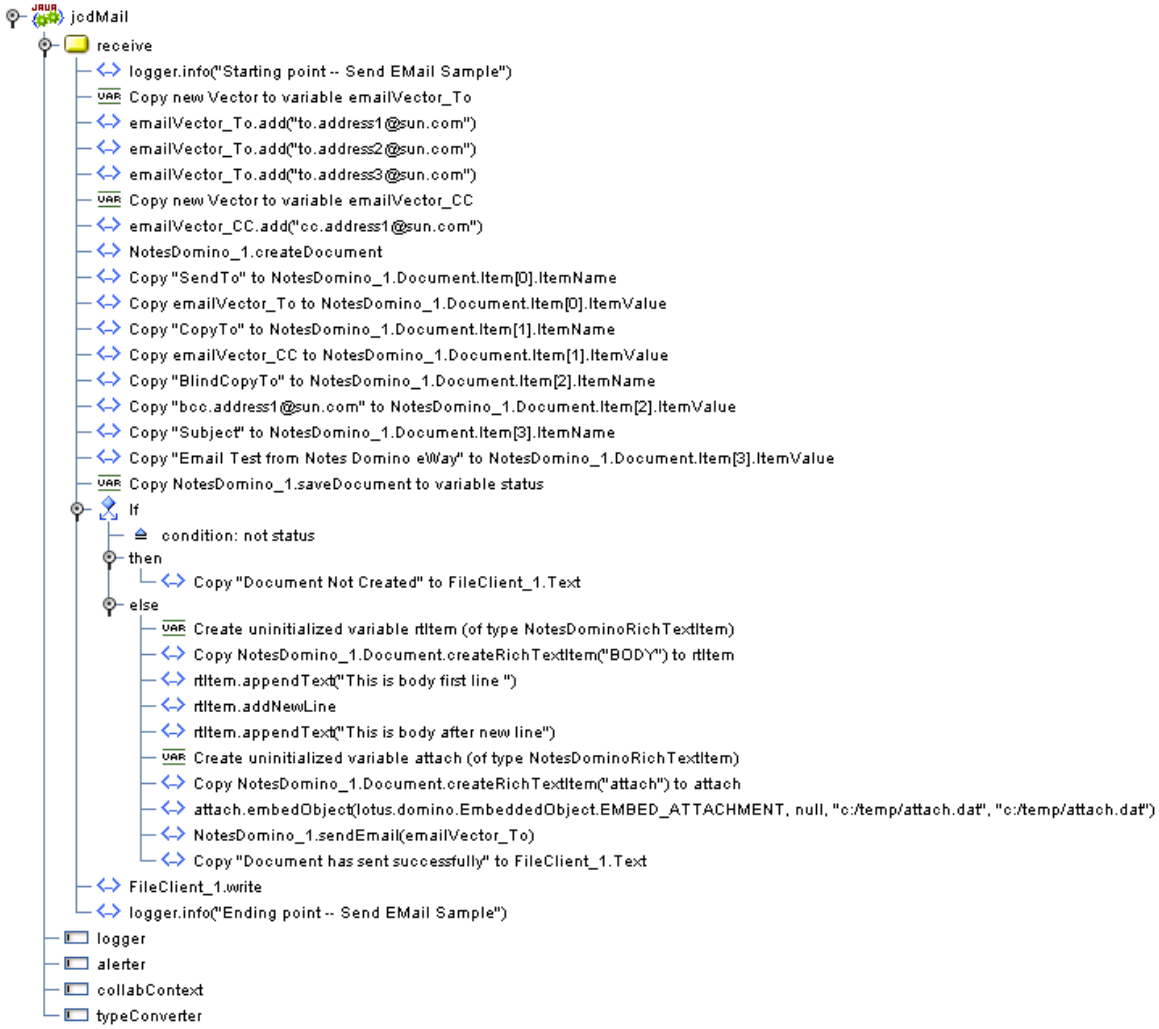
The next step is to create the **jcdMail** Collaboration using the **Collaboration Definition Wizard (Java)**. Once the Java Collaboration Definition has been created, the Business Rules of the Collaboration are written using the **Collaboration Editor (Java)**.

1. From the Project Explorer, right-click the sample Project and select **New** > **Collaboration Editor (Java)** from the shortcut menu. The **Collaboration Definition Wizard (Java)** appears.
2. Enter a Collaboration Definition name (for this sample, **jcdMail**) and click **Next**.
3. For Step 2 of the wizard, from the Web Services Interfaces selection window, double-click **Sun SeeBeyond** > **eWays** > **File** > **FileClient** > **receive**. The File Name field now displays **receive.** Click **Next**.
4. For Step 3 of the wizard, from the Select OTDs selection window, double-click **Sun SeeBeyond** > **eWays** > **File** > **FileClient**. The **FileClient_1** OTD is added to the Selected OTDs field.
5. Click the **Up One Level** button to return to the Repository. Double-click **Sun SeeBeyond** > **eWays** > **NotesDomino** > **NotesDomino**. The Selected OTDs field now lists the **NotesDomino_1** OTD.
6. Click **Finish**. The Collaboration Editor (Java) with the new Collaboration, **CreateDoc** appears, and the Collaboration is added to the Project Explorer tree.

4.6.3 Create the jcdMail Collaboration Business Rules

The next step is to create the Business Rules for **jcdMail** Collaboration using the Java Collaboration Editor. The **jcdMail** Collaboration contains the Business Rules displayed in Figure 34.

Figure 34 jcdMail Collaboration Business Rules



The previous sample Project walkthrough demonstrated how to use the Collaboration Editor's **Business Rules Designer** to create a Collaboration's Business Rules. Business Rules can also be created or edited using the Collaboration Editor's **Java Source Editor**.

The Business Rules of the **jcdMail** Collaboration perform the follow operations:

- Creates a vector object to hold the recipient email addresses.
- Creates a document using the Notes Domino **createDocument** method, and adds the SendTo, CopyTo, BlindCopyTo, and Subject items to the document.
- Creates the body and attachment for the email using **NotesDominoRichTextItem**.
- Sends the email using the **sendEmail(To_Vector)** NotesDomino method.

Sending email using the Lotus Notes/Domino eWay

To send email using the Lotus Notes/Domino eWay, you create a Notes document and compose the email fields as individual items of the document.

1. **Create a new document**: A Notes Domino document is used to compose an email message. You can create a document using the following Java code:

```
// Create a new document
NotesDomino_1.createDocument();
```

2. **Enter "To" recipient addresses**: The IBM Lotus Notes/Domino server expects a Notes document item named **SendTo** for email "To" recipients. A java.util.Vector object is needed for the SendTo item. This is a required item, and the Vector object is used when a send mail function is called. For example, you can add three To recipient email address using the following Java code

```
java.util.Vector emailVector_To = new java.util.Vector();
emailVector_To.add( "to.address1@sun.com" );
emailVector_To.add( "to.address2@sun.com" );
emailVector_To.add( "to.address3@sun.com" );
// To recipient
NotesDomino_1.getDocument().getItem( 0 ).setItemName( "SendTo" );
NotesDomino_1.getDocument().getItem( 0 ).setItemValue(
emailVector_To );
```

   Replace the sample *addresses* with real email addresses.

3. **Enter "CC" recipient addresses**: The Lotus Notes/Domino server expects a Notes document item named **CopyTo** for email "CC" recipients. This is an optional item. If a single recipient address is needed, you can assign the CC address to an item value directly using the following code:

```
// CC recipient
NotesDomino_1.getDocument().getItem( 1 ).setItemName( "CopyTo" );
NotesDomino_1.getDocument().getItem( 1 ).setItemValue(
"cc.address1@sun.com" );
```

   If multiple CC addresses are needed, another Vector object can be created as follows:

```
// CC recipient
java.util.Vector emailVector_CC = new java.util.Vector();
emailVector_CC.add( "cc.address1@sun.com" );
emailVector_CC.add( "cc.address2@sun.com" );
NotesDomino_1.getDocument().getItem( 1 ).setItemName( "CopyTo" );
NotesDomino_1.getDocument().getItem( 1 ).setItemValue(
emailVector_CC );
```

4. **Enter "BCC" recipient addresses**: The Lotus Notes/Domino server expects a Notes document item named **BlindCopyTo** for email "BCC" recipients. This is an optional item. If a single recipient address is needed, you can assign the BCC address to item value directly as follows:

```
// BCC recipient
NotesDomino_1.getDocument().getItem( 2 ).setItemName(
"BlindCopyTo" );
NotesDomino_1.getDocument().getItem( 2 ).setItemValue(
"bcc.address1@sun.com" );
```

   If multiple addresses are needed, another Vector object can be created as follows:

```
// BCC recipient
```

```
    java.util.Vector emailVector_BCC = new java.util.Vector();
    emailVector_BCC.add( "bcc.address1@sun.com" );
    emailVector_BCC.add( "bcc.address2@sun.com" );
    NotesDomino_1.getDocument().getItem( 2 ).setItemName(
"BlindCopyTo" );
    NotesDomino_1.getDocument().getItem( 2 ).setItemValue(
emailVector_BCC );
```

5 5) **Enter the email subject**: The Lotus Notes/Domino server expects a Notes document item named **Subject** for the email's subject. This is an optional item. User can assign a String value directly for the Subject field as follows:

```
    // Subject of the mail
    NotesDomino_1.getDocument().getItem( 3 ).setItemName( "Subject"
);
    NotesDomino_1.getDocument().getItem( 3 ).setItemValue( "Email
Test from Notes Domino eWay" );
```

6 **Enter the email body**: You can use a document rich text item to append the email message body line by line. For example:

```
    // body of the mail
    com.stc.connector.notesdominoadapter.appconn.NotesDominoRichTextI
tem rtItem;
    rtItem = NotesDomino_1.getDocument().createRichTextItem( "BODY"
);
    rtItem.appendText( "This is body first line " );
    rtItem.addNewLine();
    rtItem.appendText( "This is body after new line" );
```

7 **Add an attachment**: If there is an attachment for your email message, you can create another rich text item. You also need to call the **lotus.domino.EmbeddedObject.EMBED_ATTACHMENT** method. This method is found in the Notes/Domino native java class archive. For a local database, the **Notes.jar** file must be imported into the Project. For a remote connection, the **NCSO.jar** file is used instead (see **"Adding Lotus Notes/Domino JAR files" on page 14** for more information on locating these JAR files).

The following is an example of the Java code used to add an attachments:

```
    // attachment of the mail
    com.stc.connector.notesdominoadapter.appconn.NotesDominoRichTextI
tem attach;
    attach = NotesDomino_1.getDocument().createRichTextItem( "attach"
);
    attach.embedObject( lotus.domino.EmbeddedObject.EMBED_ATTACHMENT,
null, "c:/temp/attach.dat", "c:/temp/attach.dat" );
```

8 **Send the email**: Once the email document is composed completely, you can call the send email method. A Vector parameter is expected for "To" recipient addresses:

```
    // send email
    NotesDomino_1.sendEmail( emailVector_To );
```

The jcdMail Collaboration's completed Java code appears as follows:

```
package prjNotesDominoMail_Sample_JCD;


public class jcdMail
{

    public com.stc.codegen.logger.Logger logger;
```

```
    public com.stc.codegen.alerter.Alerter alerter;

    public com.stc.codegen.util.CollaborationContext collabContext;

    public com.stc.codegen.util.TypeConverter typeConverter;

    public void receive( com.stc.connector.appconn.file.FileTextMessage input,
com.stc.connector.notesdominoadapter.appconn.NotesDominoApplication NotesDomino_1,
com.stc.connector.appconn.file.FileApplication FileClient_1 )
        throws Throwable
    {
        logger.info( "Starting point -- Send EMail Sample" );
        // Add recipient addresses in data type Vector
        java.util.Vector emailVector_To = new java.util.Vector();
        emailVector_To.add( "to.address1@sun.com" );
        emailVector_To.add( "to.address2@sun.com" );
        emailVector_To.add( "to.address3@sun.com" );
        java.util.Vector emailVector_CC = new java.util.Vector();
        emailVector_CC.add( "cc.address1@sun.com" );
        // Create a new document
        NotesDomino_1.createDocument();
        // To recipient
        NotesDomino_1.getDocument().getItem( 0 ).setItemName( "SendTo" );
        NotesDomino_1.getDocument().getItem( 0 ).setItemValue( emailVector_To );
        // CC recipient
        NotesDomino_1.getDocument().getItem( 1 ).setItemName( "CopyTo" );
        NotesDomino_1.getDocument().getItem( 1 ).setItemValue( emailVector_CC );
        // BCC recipient
        NotesDomino_1.getDocument().getItem( 2 ).setItemName( "BlindCopyTo" );
        NotesDomino_1.getDocument().getItem( 2 ).setItemValue( "bcc.address1@sun.com" );
        // Subject of the mail
        NotesDomino_1.getDocument().getItem( 3 ).setItemName( "Subject" );
        NotesDomino_1.getDocument().getItem( 3 ).setItemValue( "Email Test from Notes Domino eWay"
);
        // Save document
        boolean status = NotesDomino_1.saveDocument();
        if (!status) {
            FileClient_1.setText( "Document Not Created" );
        } else {
            // body of the mail
            com.stc.connector.notesdominoadapter.appconn.NotesDominoRichTextItem rtItem;
            rtItem = NotesDomino_1.getDocument().createRichTextItem( "BODY" );
            rtItem.appendText( "This is body first line " );
            rtItem.addNewLine();
            rtItem.appendText( "This is body after new line" );
            // attachment of the mail
            com.stc.connector.notesdominoadapter.appconn.NotesDominoRichTextItem attach;
            attach = NotesDomino_1.getDocument().createRichTextItem( "attach" );
            attach.embedObject( lotus.domino.EmbeddedObject.EMBED_ATTACHMENT, null, "c:/temp/
attach.dat", "c:/temp/attach.dat" );
            // send email
            NotesDomino_1.sendEmail( emailVector_To );
            FileClient_1.setText( "Document has sent successfully" );
        }
        FileClient_1.write();
        logger.info( "Ending point -- Send EMail Sample" );
    }

}
```

4.6.4 Create the Connectivity Map

The Connectivity Map provides a canvas for assembling and configuring a Project's components. This sample utilizes one Connectivity Map.

1. From the Project Explorer tree, right-click the new **prjNotesDomino_EMailSample_JCD** Project and select **New** > **Connectivity Map** from the shortcut menu. The New Connectivity Map appears and a node for the Connectivity Map is added under the Project on the Project Explorer tree labeled **CMap1**.
2. Click the **External Application** icon on the Connectivity Map toolbar,

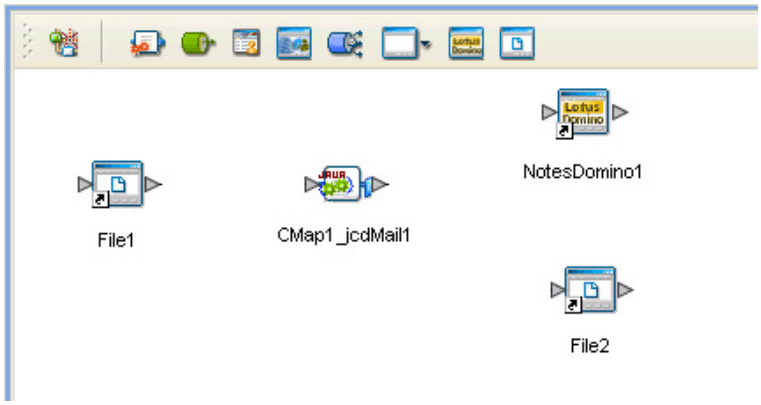
3 Select the External Applications required by your Project (for this sample, the **Notes/Domino** and **File** External Applications). Icons representing the selected External Applications are added to the Connectivity Map toolbar.

4 Populate the **CMap1** Connectivity Map with the necessary components by dragging the following objects onto the Connectivity Map canvas, as displayed in Figure 35:

- File External Application (2 for this sample)
- Notes/Domino External Application (1 for this sample)

5 Drag the **jcdMail** Collaboration from the Project Explorer tree onto the Connectivity Map. The Collaboration becomes the **CMap1_jcdMail1** service (see Figure 35).

Figure 35 Connectivity Map with Components



4.6.5 Binding the eWay Components

After the Collaboration has been written and the Connectivity Map has been populated, the components can be associated (create bindings).

1 From the Connectivity Map canvas, double-click the **CMap1_jcdMail1** service. The **CMap1_jcdMail1** binding dialog box appears using the **jcdMail** Rule.

2 From the **CMap1_jcdMail1** binding dialog box, map **FileClient Input** (under Implemented Services) to the outbound node of the inbound **File1** External Application. To do this, click on the binding dialog box's **FileClient Input** node, and drag your cursor to the output node of the **File1** External Application. A link now connects the two nodes.

3 Map **NotesDomino_1** (under Invoked Services) to the **NotesDomino1** External Application

4 Map **FileClient_1** (under Invoked Services) to the outbound **File2** External Application.

5 Minimize the **CMap1_jcdMail1** binding dialog box and save your changes to your Repository.

4.6.6 Creating an Environment

Environments include the external systems, Logical Hosts, integration servers and message servers used by a Project, and contain the configuration information for these components. Environments are created using the Enterprise Designer's Environment Editor.

1. From the Enterprise Designer's Enterprise Explorer, right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.
2. Rename the new Environment to **envNotesDomino_EMailSample_JCD**.
3. Right-click **envNotesDomino_EMailSample_JCD** and select **New File External System**. Name the External System **esFile** and click **OK**. **esFile** is added to the Environment Editor.
4. Right-click **envNotesDomino_EMailSample_JCD** and select **New NotesDomino External System**. Name the External System **esNotesDomino** and click **OK**. **esNotesDomino** is added to the Environment Editor.
5. Right-click **envNotesDomino_EMailSample_JCD** and select **New Logical Host**. **LogicalHost1** is added to the Environment Editor.
6. From the Environment Explorer tree, right-click **LogicalHost1** and select **New > Sun SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under **LogicalHost1**.
7. Save your changes to your Repository.

4.6.7 Adding the Required JAR File to the Logical Host.

Prior to deploying your Lotus Notes/Domino eWay Project, copy the appropriate Jar file to your Logical Host. For directions, see **"Adding Lotus Notes/Domino JAR files" on page 14**

4.6.8 Configuring the eWays

The **prjNotesDomino_EMailSample_JCD** Project contains three component eWays. The File eWay's properties are configured from the Connectivity Map and the Environment Explorer. The Lotus Notes/Domino eWay properties are set from the Environment Explorer tree (the Notes/Domino eWay contains no Connectivity Map properties).

Configure the File eWay Connectivity Map Properties

Connectivity Map properties are specific to each component eWay. To open the Connectivity Map Properties Editor for each component File eWay, open the appropriate Connectivity Map, and double-click the node in the link between the service and the File External Application. The Properties Editor appears with the configuration for that specific eWay.

Configure your Project eWays as follows:

1. From the **CMap1** Connectivity Map, Double-click the inbound **File1** eWay.
2. The **Properties Editor** opens to the inbound **File1** eWay properties.
3. Click the **Input file name** property field and enter **Input.txt** as the property value. Click **OK** to save your settings and close the Properties Editor.
4. Double-click the outbound **File2** eWay.
5. Click the **Output file name** property field and enter **NotesDominoMail%d.dat** as the property value. Click **OK** to save your settings and close the Properties Editor.

Configure the File eWay Environment Properties

Environment properties are specific to all of the component eWays assigned to an External System in the Environment. To configure the File eWay Environment properties, do the following:

1. From the **Environment Explorer** tree, right-click the File External System (**esFile** in this sample), and select **Properties**. The Properties Editor opens to the File eWay Environment configuration.
2. Modify the File eWay Environment configuration properties for your system, including the settings in Table 8, and click **OK**.

Table 8 File eWay Environment Explorer Properties

File eWay Environment Properties	
Inbound File eWay - Set as directed, otherwise use the default settings	
Parameter Settings > Directory	C:/temp/LotusNotes *or the input directory of your choice*
Outbound File eWay - Set as directed, otherwise use the default settings.	
Parameter Settings > Directory	C:/temp/LotusNotes *or the output directory of your choice*

Configure the Lotus Notes/Domino eWay Properties

The Lotus Notes/Domino eWay only contains Environment properties. To configure the Lotus Notes/Domino eWay Environment properties, do the following:

1. From the **Environment Explorer** tree, right-click the Lotus Notes/Domino eWay External System (**esNotesDomino** in this sample), and select **Properties** from the shortcut menu. The Properties Editor appears.
2. Modify the Lotus Notes/Domino eWay Environment properties for your system, including the settings in Table 9, and click **OK**.

Table 9 Notes/Domino Environment Properties

Lotus Notes/Domino eWay Environment Properties
Environment Configuration Settings - Set as directed, otherwise use the default settings

Table 9 Notes/Domino Environment Properties

Lotus Notes/Domino eWay Environment Properties	
Database Type	*Specify **Remote** or **Local**, according to your system*
Notes/Domino Server	Specify a NotesDomino server
Notes/Domino Database	Specify a database
Notes/Domino User	Specify a NotesDomino server authorized user login
Password	Specify a NotesDomino server password for the specified user

For more information on the Lotus Notes/Domino eWay properties and the Properties Editor, see **Lotus Notes/Domino eWay Properties** on page 18.

4.6.9 Configuring the Integration Server

You must set your Sun SeeBeyond Integration Server Password property before deploying your Project.

1. From the Environment Explorer, right-click **IntegrationSvr1** under your **Logical Host**, and select **Properties** from the shortcut menu. The Integration Server Properties Editor appears.
2. Click the **Password** property field under **Sun SeeBeyond Integration Server Configuration**. An ellipsis appears in the property field.
3. Click the ellipsis. The **Password Settings** dialog box appears. Enter **STC** as the **Specific Value** and as the **Confirm Password**, and click **OK**.

   Click **OK** to accept the new property and close the Properties Editor.

4.6.10 Creating the Deployment Profile

A Deployment Profile is used to assign Collaborations and message destinations to the integration server and message server. Deployment profiles are created using the Deployment Editor.

1. From the Enterprise Explorer's Project Explorer, right-click your Project and select **New** > **Deployment Profile**. The **Create Deployment Profile** dialog box appears.
2. Enter the name of your Project Deployment Profile (for this sample **dpNotesDomino_EMailSample_JCD**). Select **envNotesDomino_EMailSample_JCD** as the Environment and click **OK**. The **Deployment Editor** appears.
3. From the Deployment Editor toolbar, click the **Automap** icon. The Project's components are automatically mapped to their External Systems.

4.6.11 Creating and Starting the Domain

To build and deploy your Project, you must first create a domain. A domain is an instance of a Logical Host. After the domain is created, the Project is built and then deployed.

Create and Start the Domain

1. Navigate to your **<*JavaCAPS51*>\logicalhost** directory (where <*JavaCAPS51*> is the location of your Sun Java Composite Application Platform Suite installation.
2. Double-click the **domainmgr.bat** file. The **Domain Manager** appears.
3. If you have already created a domain, select your domain in the Domain Manager and click the **Start an Existing Domain** button. Once your domain is started, a green check mark indicates that the domain is running.
4. If there are no existing domains, a dialog box indicates that you can create a domain now. Click **Yes**. The **Create Domain** dialog box appears.
5. Make any necessary changes to the **Create Domain** dialog box and click **Create**. The new domain is added to the Domain Manager. Select the domain and click the **Start an Existing Domain** button. Once your domain is started, a green check mark indicates that the domain is running.

For more information about creating and managing domains see the *Sun SeeBeyond eGate™ Integrator User's Guide* and the *Sun SeeBeyond eGate™ Integrator System Administration Guide*.

4.6.12 Build and Deploy the Project

The Build process compiles and validates the Project's Java files and creates the Project EAR file.

Build the Project

1. From the Deployment Editor toolbar, click the **Build** icon.
2. If there are any validation errors, a **Validation Errors** pane will appear at the bottom of the Deployment Editor and displays information regarding the errors. Make any necessary corrections and click **Build** again.
3. After the Build has succeeded you are ready to deploy your Project.

Deploy the Project

1. From the Deployment Editor toolbar, click the **Deploy** icon. Click **Yes** when the **Deploy** prompt appears.
2. A message appears when the Project is successfully deployed. You can now test your sample.

Note: *Projects can also be deployed from the Enterprise Manager. For more information about using the Enterprise Manager to deploy, monitor, and manage your Projects, see the Sun SeeBeyond eGate™ Integrator System Administration Guide.*

4.6.13 Running the Sample

Create your Input and Output directories to match the directory names you designated in your File eWay properties. The sample Project comes with the following input and example files:

- input.txt.~in
- attach.dat
- NotesDominoMail1.dat

If you are using the optional attachment, add the attach.dat (or the attachment you designated in your Collaboration) to your C:/temp directory (or the directory you specify in your Collaboration).

To run your deployed sample, paste (or rename) the sample input file to your configured input directory, to trigger the eWay.

Index

S

T