Creating an End-to-End Scenario with e*Gate Integrator

Release 4.5.3

SeeBeyond™

Contents

Chapter 4

Creating an ELS End-to-End Scenario   41

Introduction

Welcome to the *Creating an End-to-End Scenario* guide. This chapter provides an overview of the purpose and contents of this guide. You will also find references for related documentation and the writing conventions used in this guide.

In This Chapter

- **About This Guide** on page 6
- **Contents of This Guide** on page 6
- **Writing Conventions** on page 7
- **Supporting Documents** on page 8
- **SeeBeyond Web Site** on page 9

1.1 About This Guide

This guide helps those of you who are familiar with the e*Gate Enterprise Manager to create sample e*Gate schemas called scenarios.

In this guide, you will find three step-by-step examples of creating schemas. The first two examples, or scenarios, create schemas for Java Collaboration environments. The third scenario sets up a schema for a Monk Collaboration environment.

This guide does not provide information about e*Gate or e*Gate Enterprise Manager beyond the creating of a schema. For addition information about e*Gate, refer to the documents listed in **Supporting Documents** on page 8.

1.2 Contents of This Guide

This guide includes the following information:

- **Chapter 1, "Introduction"** provides an overview of this document's purpose, contents, writing conventions, and supported documents.

- **Chapter 2, "Before You Begin"** provides a roadmap for creating scenarios. The roadmap includes, in broad outline, the steps for creating a scenario. This chapter also includes the e*Gate installation requirements for creating the scenarios in this guide.

- **Chapter 3, "Creating a Java End-to-End Scenario"** provides instructions for creating a schema that uses the Java Collaboration service to handle data transformation.

- **Chapter 4, "Creating an ELS End-to-End Scenario"** provides instructions for creating a schema that uses Event Linking and Sequencing (ELS) to combine transactions that can arrive in many parts over time.

- **Chapter 5, "Creating a Monk End-to-End Scenario"** provides instructions for creating a schema that uses the Monk Collaboration service to handle data routing.

- **Chapter 6, "Debugging and Log Files"** provides introductory information about troubleshooting your schemas with log files.

1.3 Writing Conventions

The writing conventions listed in this section are observed throughout this document.

Hypertext Links

When you are using this guide online, cross-references are also hypertext links and appear in **blue text** as shown below. Click the **blue text** to jump to the section.

For information on these and related topics, see **"Parameter, Function, and Command Names" on page 8**.

Command Line

Text to be typed at the command line is displayed in a special font as shown below.

```
java -jar ValidationBuilder.jar
```

Variables within a command line are set in the same font and bold italic as shown below.

```
stcregutil -rh host-name -rs schema-name -un user-name
-up password -ef output-directory
```

Code and Samples

Computer code and samples (including printouts) on a separate line or lines are set in Courier as shown below.

```
Configuration for BOB_Promotion
```

However, when these elements (or portions of them) or variables representing several possible elements appear within ordinary text, they are set in *italics* as shown below.

path and *file-name* are the path and file name specified as arguments to **-fr** in the **stcregutil** command line.

Notes and Cautions

Points of particular interest or significance to the reader are introduced with *Note*, *Caution*, or *Important*, and the text is displayed in *italics*, for example:

Note: The Actions menu is only available when a Properties window is displayed.

User Input

The names of items in the user interface such as icons or buttons that you click or select appear in **bold** as shown below.

Click **Apply** to save, or **OK** to save and close.

File Names and Paths

When names of files are given in the text, they appear in **bold** as shown below.

Use a text editor to open the **ValidationBuilder.properties** file.

When file paths and drive designations are used, with or without the file name, they appear in **bold** as shown below.

In the **Open** field, type **D:\setup\setup.exe** where **D:** is your CD-ROM drive.

Parameter, Function, and Command Names

When names of parameters, functions, and commands are given in the body of the text, they appear in **bold** as follows:

The default parameter **localhost** is normally only used for testing.

The Monk function **iq-put** places an Event into an IQ.

You can use the **stccb** utility to start the Control Broker.

Additional Conventions

Windows Systems — e*Gate is compliant with Windows NT, Windows 2000, and Windows XP. When this document refers to Windows, such statements apply to all three Windows platforms.

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

1.4   Supporting Documents

The following SeeBeyond documents provide additional information about the e*Gate Integrator system as explained in this guide:

- *e*Gate Integrator Collaboration Services Reference Guide*
- *e*Gate Integrator Installation Guide*
- *e*Gate Integrator Intelligent Queue Services Reference Guide*
- *e*Gate Integrator System Administration and Operations Guide*
- *e*Gate Integrator User's Guide*
- *SeeBeyond eBusiness Integration Suite Primer*
- *SeeBeyond eBusiness Integration Suite Deployment Guide*
- *Monk Developer's Reference*

- *Standard e*Way Intelligent Adapters User's Guide*

See the *SeeBeyond eBusiness Integration Suite Primer* for a complete list of e*Gate-related documentation. You can also refer to the appropriate Microsoft Windows or UNIX documents, if necessary.

1.5 SeeBeyond Web Site

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

Before You Begin

This chapter provides a roadmap for creating scenarios. The roadmap includes, in broad outline, the steps for creating a scenario. This chapter also includes the e*Gate installation requirements for creating the scenarios in this guide.

In This Chapter

- **Roadmap for Creating Scenarios** on page 10
- **e*Gate Installation Requirements** on page 11

2.1 Roadmap for Creating Scenarios

The procedure below provides a roadmap for creating the end-to-end scenarios. You can switch the order of certain items; in cases where you cannot, the applications alerts you and lets you create the required component before you continue.

To create scenarios

1. Verify the e*Gate installation and system requirements.
2. Create a new schema.
3. Create the Event Types and ETDs.
4. Create the Collaboration Rules.
5. Add the e*Ways, the JMS e*Way Connection (for Java schemas), and the IQs (for Monk schemas).
6. Add the Collaborations.
7. Review the complete schema.
8. Create the input data file.
9. Start the schema.
10. Test the schema.
11. Troubleshoot any problems.

Compared to the schema process presented in the schema diagrams in the following chapters, the roadmap procedure reverses the schema process. The roadmap calls for creating the Event Types and Collaborations before you create the e*Ways that use them. This method is more efficient because you can create all components of the same

type at the same time. It also ensures that the required components are available when you need them.

2.2 e*Gate Installation Requirements

To be able to create the scenarios described in this guide, you must have all e*Gate components installed on a single Windows system. Your system must have the following e*Gate components installed:

- Registry Host
- Participating Host
- GUIs (e*Gate Enterprise Manager and e*Gate Monitor)

Refer to the *e*Gate Integrator Installation Guide* for system requirements and installation instructions.

For the Java scenarios, you must have the following items available:

- Java 2 SDK, Standard Edition, v 1.3.1_02 (JDK 1.3.1_02)

  You can download the JDK 1.3.1_02 from **www.java.sun.com**.

- Internet Explorer 5.x or later

  The Java Collaboration Editor requires Internet Explorer 5.x or later. You can download Internet Explorer from **www.microsoft.com**.

Creating a Java End-to-End Scenario

This chapter provides step-by-step procedures for creating and testing an e*Gate schema with Java Collaboration Rules. This schema addresses a business challenge in a scenario where a company uses two systems to handle time tracking and payroll distribution. The systems use dissimilar formats which presents a challenge when one system must use data from the other system to accomplish its tasks.

This chapter describes the business challenge and how you can create a Java schema in e*Gate to allow the two systems to work together effectively.

In This Chapter

3.1 Presenting the Business Challenge

The Java end-to-end scenario described in this chapter provides a solution to the following business challenge:

System A is a timecard system and tracks the weekly hours worked by employees. The data in system A is in a delimited format. System B is a payroll system and is responsible for employee payment. The data in system B is in a fixed format of four fields: first name, last name, employee number, and amount. System A's data does not include amount information.

To integrate system A successfully with system B, the following issues must be addressed:

- System A's data in delimited format must be converted to system B's fixed format
- The value for system B's amount field must be calculated before it is sent to system B

The figure below shows the relationship between the data in system A and system B.
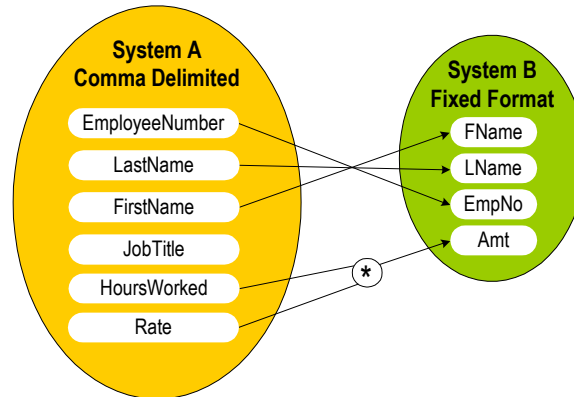
3.2 Presenting the e*Gate Solution

The e*Gate solution uses the e*Gate Java Collaboration Service to transform the data from the system A format to the system B format. e*Gate provides flexibility regarding the location of transformation processing as the data transfers from system A to system B. The solution uses the Multi-Mode e*Way as the main transformation component and two Java Pass Through File e*Ways to bring data into and move data out of e*Gate.

The figure below shows the e*Gate schema.
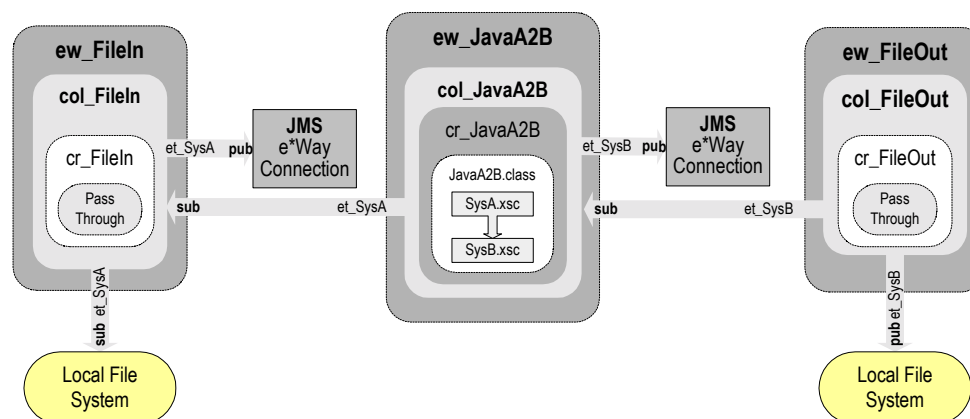
In this figure, you see the following process:

1 The **ew_FileIn** e*Way brings system A data into e*Gate as follows:

The **col_FileIn** Collaboration in the **ew_FileIn** e*Way subscribes to a location on the local file system. It polls this location for a **.fin** file, which is a text file with data from system A. If it finds a **.fin** file, it reads the message**,** packages the data as an **et_SysA** Event, and publishes the Event to the JMS e*Way Connection.

2 The **ew_JavaA2B** e*Way transforms the data format and calculates the amount.

The **col_JavaA2B** Collaboration in the **ew_JavaA2B** e*Way subscribes to **et_SysA** Events published by **col_FileIn**. It uses the Java Collaboration Rules component **cr_JavaA2B** to change **et_SysA** Events into **et_SysB** Events. This rule uses the **JavaA2B.class** which implements the transformation. **cr_JavaA2B** also computes the amount field in **et_SysB** by multiplying the hours and the rate from **et_SysA**. Finally, **col_JavaA2B** publishes the **et_SysB** Event to the JMS e*Way Connection.

3 The **ew_FileOut** e*Way writes the transformed data out to local file system.

The **col_FileOut** Collaboration in the **ew_FileOut** e*Way subscribes to **et_SysB** Events published to the JMS e*Way Connection by the **col_JavaA2B** Collaboration. The **cr_FileOut** Collaboration Rules component uses the Java Pass Through service to move the data without modifying it. When an **et_SysB** Event is retrieved, the e*Way packages it as a text file and writes it to the specified location on the local file system, completing the end-to-end scenario.

3.3 Creating the Schema

Before you start creating the schema, make sure that you have the following system requirements:

- e*Gate Registry Host, Participating Host, and GUIs installed on a single system
- The system is running the Windows operating system
- The system has Internet Explorer 5.x or later installed
- The system has Java 2 SDK, Standard Edition, v 1.3.1_02 (JDK 1.3.1_02) installed

For more information about the installation requirements, refer to **"e*Gate Installation Requirements" on page 11.**

To create the schema

1 Start the e*Gate Enterprise Manager and log in as **Administrator** (or another user with administrator privileges) to the appropriate Registry Host.

2 In the **Open Schema on Registry Host** dialog box, click **New**.

3 In the **Enter New Schema Name** box, type **JavaE2E** and click **Open**.

The e*Gate Enterprise Manager displays the new **JavaE2E** schema.

4 Click the **Components** tab. You perform all configuration steps in the **Components** tab.
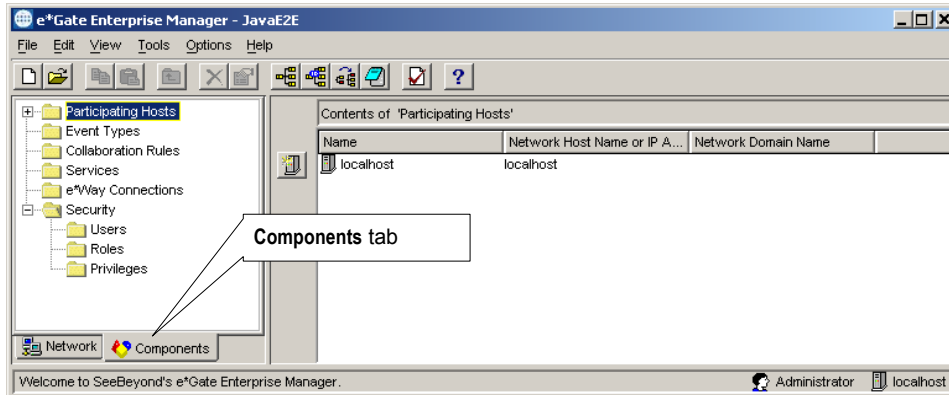
3.4 Creating the Event Types and Java ETDs

This scenario uses two Event Types. These Event Types require their own Event Type Definitions (ETDs). The first Event Type, **et_SysA**, models the comma-delimited format of the data received from system A as defined in **SysA.xsc**. The second Event Type, **et_SysB**, models the fixed-length data format required by system B as defined in **SysB.xsc**.

The figure below shows where these parts fit into the schema.
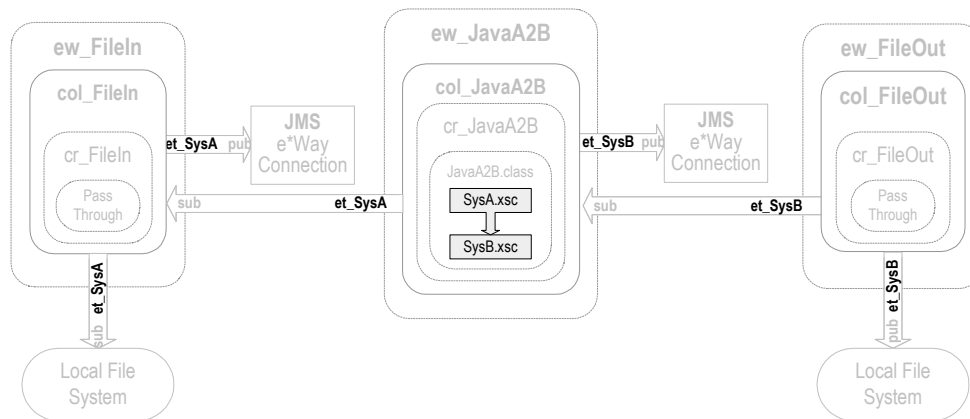
3.4.1 Creating the SysA.xsc ETD

The procedure below describes how you create the SysA.xsc ETD. Before you start, you must have already created the JavaE2E schema as described in **"Creating the Schema" on page 14**.

To create an ETD, you specify the ETD name, and the ETD's root node name and a package name.

The package name is the name of the collection of Java classes that make up the ETD. This group of Java classes is the result of the ETD creation process. The root node name and package name **cannot** be the same.

To create the SysA.xsc ETD

1 In the e*Gate Enterprise Manager, click **Default Editor** on the **Options** menu.

2 Click **Java** if necessary and click **OK**.

3 In the **Components** tab, click the **Event Types** folder.

4 Click **Create a New Event Type** .

5 In the **Name** box, type **et_SysA** and click **OK**.

This adds **et_SysA** to the Event Types pane.

6 In the Event Types pane, double-click **et_SysA**.

The **Event Type - et_SysA Properties** dialog box appears.

7 Click **New**. The **ETD Editor** window appears.

8 Click **New** .

The **New Event Type Definition** window appears.

9 Double-click **Custom ETD Wizard** .

The **Custom ETD Wizard - Introduction** dialog box appears.

10 Click **Next**.

The **Custom ETD Wizard - Step 1** dialog box appears.

11 In the **Root Node Name** box, type **SysA**.

12 In the **Package Name** box, type **SysApackage**.

13 Click **Next**, review the summary information, and click **Finish**.

The screen below shows the new SysA.xsc ETD.

Adding SysA.xsc Nodes

The procedure below describes how you add nodes to the SysA.xsc ETD. Before you start, you must have already created ETD as described in **"Creating the SysA.xsc ETD" on page 15**.

To add SysA.xsc nodes

1 In the Event Type Definition pane, right-click **SysA**, point to **Add Field**, and then click **As Child Node**.

   This adds a **Field1** node under the **SysA** root node.

2 Right-click the **Field1** node and click **Rename**.

3 Type **EmployeeNumber** and press ENTER.

4 Repeat steps 1 through 3 to add the following nodes:

   ◆ LastName

   ◆ FirstName

   ◆ JobTitle

   ◆ HoursWorked

   ◆ Rate

   The screen below shows the completed SysA.xsc ETD
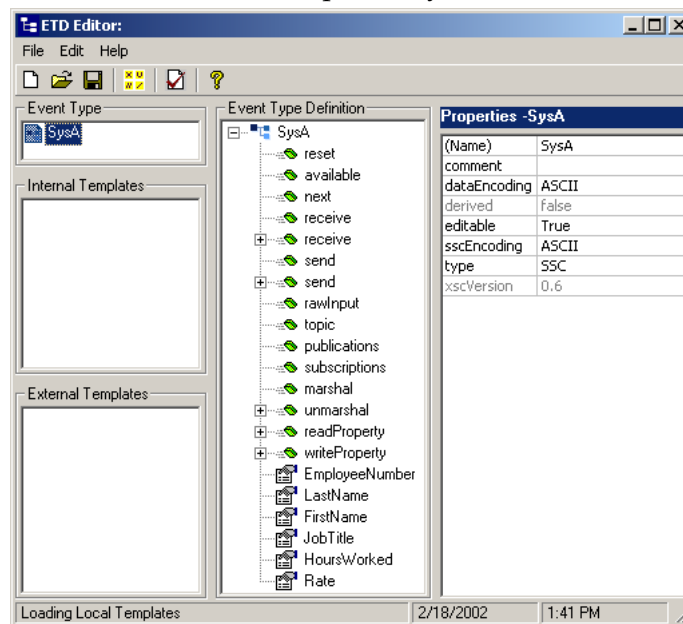
5 Click **Global Delimiters** 🔲.

6 Click **Add Level**.

7 Click **Add Single End Delimiter**.

8 Click **Value** in the Properties pane and enter "**,**" (a comma, no quotes).

9 Click **OK**.

10 On the **File** menu, click **Compile and Save**.

11 In the **File name** box, change **EventTypeDefinition1.xsc** to **SysA.xsc**.

12 Click **New Folder** .

13 In the **Enter directory to create** box, type **JavaE2E** and click **OK**.

14 Double-click the **JavaE2E** folder and click **Save**.

This compiles and saves the SysA ETD as **SysA.xsc** in the **\etd\JavaE2E** folder.

15 On the **File** menu, click **Close**.

The Java ETD Editor closes and the location of the file **SysA.xsc** now displays in the Event Type Definition pane.

16 Click **OK** to close the **Event Type - et_SysA Properties** dialog box.

3.4.2 Creating the SysB.xsc ETD

The SysB.xsc ETD corresponds to the fixed data structure used by system B.

Use the procedure in **"Creating the SysA.xsc ETD" on page 15** to create a new Event Type named **et_SysB** and a corresponding Java ETD with root node name **SysB** and package name **SysBpackage**. Then use the following procedure to add nodes to **SysB.xsc** in the ETD Editor.
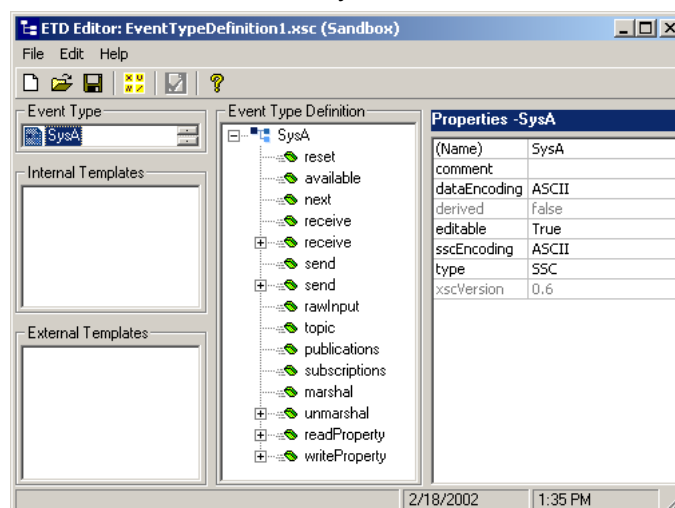
Adding SysB.xsc Nodes

The procedure below describes how you add nodes to the SysB.xsc ETD. Before you start, you must have already created ETD as described in **"Creating the SysB.xsc ETD" on page 18**.

To add SysB.xsc nodes

1 In the Event Type Definition pane in the ETD Editor, click the **SysB** node.

The SysB node properties appear in the Properties pane.

2 In the Properties pane, click **structure** and select **fixed** from the list.

3 Right-click **SysB**, point to **Add Field**, and then click **As Child Node**.

4 Right-click the **Field1** node and click **Rename**.

5 Type **FName** and press Enter.

6 With the **FName** node selected, click **length** in the Properties pane.

7 Change its value from **undefined** to **25**.

8 Click **offset** in the Properties pane, and change its value from **undefined** to **0**.

9 Repeat steps 3 through 8 to set up three additional nodes as follows:

Node Name	Structure	Length	Offset
LName	fixed	25	25
EmpNo	fixed	10	50

Node Name	Structure	Length	Offset
Amt	fixed	10	60

The screen below shows the completed SysB.xsc ETD.
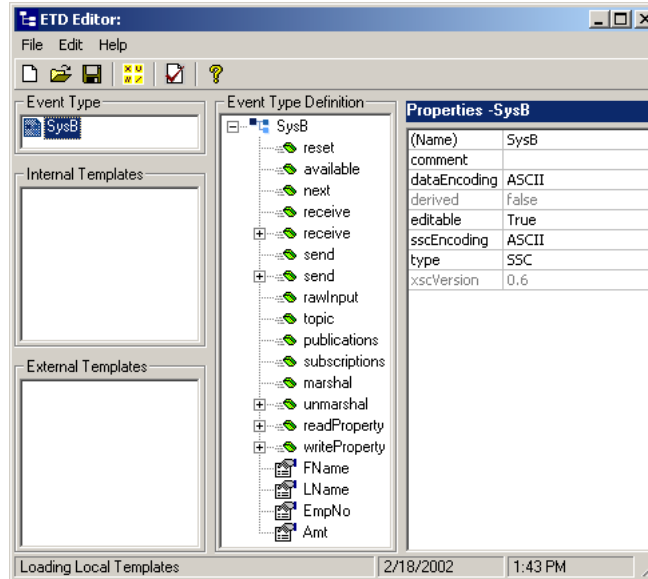
10 On the **File** menu, click **Compile and Save**.

11 In the **File name** box, change **EventTypeDefinition1.xsc** to **SysB.xsc**.

12 Double-click the **JavaE2E** folder and click **Save**.

This compiles and saves the SysB.xsc ETD as **SysB.xsc** in the **\etd\JavaE2E** folder.

13 On the **File** menu, click **Close**.

The Java ETD Editor closes and the location of the file **SysB.xsc** now displays in the Event Type Definition pane.

14 Click **OK** to close the **Event Type - et_SysB Properties** dialog box.

3.5 Creating the Collaboration Rules

The Java end-to-end scenario uses three Collaboration Rules: two Java Pass Through rules and one Java Collaboration Rules component. The Pass Through rules, **cr_FileIn** and **cr_FileOut**, route the Events through the e*Gate environment. The Collaboration Rules component **cr_JavaA2B** transforms the Event from Event Type **et_SysA** to Event Type **et_SysB**.

The figure below shows where the Collaboration Rules fit into the schema.
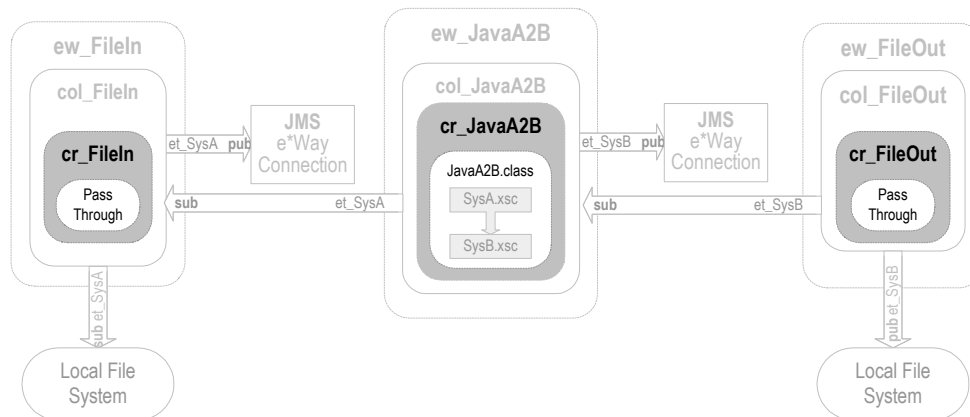
3.5.1 Creating the Java Pass Through Collaborations

The Java Pass Through Collaborations bring data into and move data out of e*Gate. The Pass Through Collaboration Rules do not change the data. The following procedure describes how you create the Java Pass Through Collaborations for this scenario.

To create the Java Pass Through Collaboration Rules

1 In the **Components** tab of the e*Gate Enterprise Manager, click the **Collaboration Rules** folder.

2 Click **Create New Collaboration Rules** .

3 In the **Name** box, type **cr_FileIn** and click **OK**.

   cr_FileIn displays in the Contents pane.

4 Double-click **cr_FileIn** in the Contents pane.

   The **Collaboration Rules - cr_FileIn Properties** dialog box appears.

5 Under **Collaboration Rules**, click **Find**.

6 Double-click **STCLibrary** and double-click **STCJavaPassThrough.class**. You see the following screen:
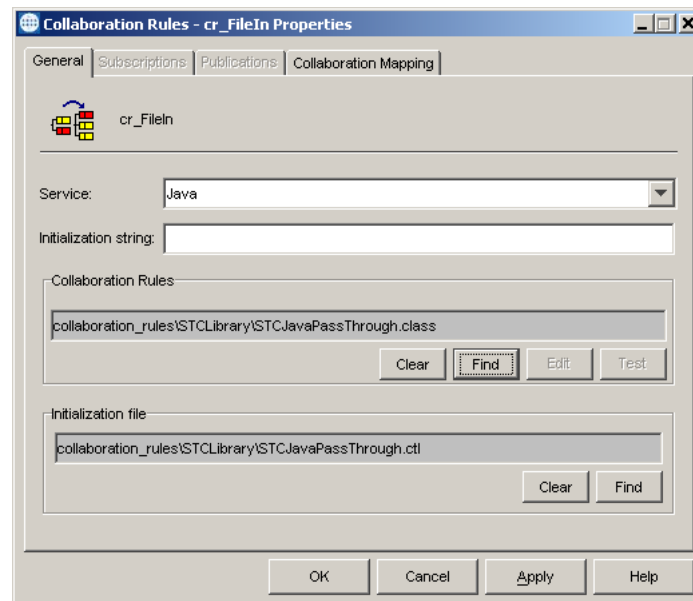
The Collaboration Rules path is set to **collaboration_rules\STCLibrary\STCJavaPassThrough.class**. The initialization file path is set to **collaboration_rules\STCLibrary\STCJavaPassThrough.ctl**. The **STCJavaPassThrough.class** file configures the Collaboration Mapping Instances.

7 Click **OK** to close the **Collaboration Rules - cr_FileIn Properties** dialog box.

8 In the Contents pane, right-click the **cr_FileIn** Collaboration Rules component.

9 Click **Copy**.

10 Right-click the Contents pane and click **Paste**.

11 Right-click the **cr_FileIn_0** Collaboration Rules component and click **Rename**.

12 In the **Name** box, type **cr_FileOut** and click **OK**.

3.5.2 Creating the cr_JavaA2B Collaboration Rules Component

This section describes how you create the Java Collaboration Rules component **cr_JavaA2B**. Creating **cr_JavaA2B** consists of the following steps:

1 Adding the component.

2 Specifying the Java script.

Adding the cr_JavaA2B Collaboration Rules Component

To add the cr_JavaA2B Collaboration Rules component

1 In the **Components** tab of the e*Gate Enterprise Manager, click the **Collaboration Rules** folder.

2 Click **Create New Collaboration Rules** .

3 In the **Name** box, type **cr_JavaA2B** and click **OK**.

4 In the Contents pane, double-click the **cr_JavaA2B** Collaboration Rules component.

5 Click the **Collaboration Mapping** tab.

6 Click **Add Instance**.

7 In the **Instance Name** box, type **Instance1** for the instance name.

8 Click **Find**, double-click the **JavaE2E** folder, and double-click **SysA.xsc**.

9 In the **Instance Name** box, type **Instance2** for the instance name.

10 Click **Find**, double-click the **JavaE2E** folder, and double-click **SysB.xsc**.

11 In the **Mode** list for Instance2**,** click **Out**.

12 Click **Apply**.

The screen below shows what the **Collaboration Mapping** tab looks like.
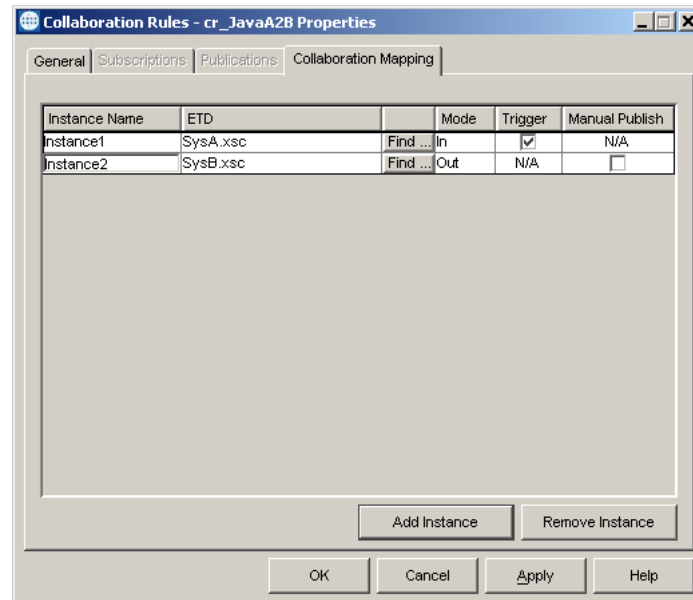
Now that you have created the Collaboration Rules component, you must follow the procedure below to specify the script behind the Collaboration Rules component.

Specifying the cr_JavaA2B Script

To specify the cr_JavaA2B script

1 In the **Collaboration Rules - cr_JavaA2B Properties** dialog box, click the **General** tab.

2 Click **New**.

The Java Collaboration Editor opens a new Collaboration Rules component with **Instance1[SysA]** as the source Event and **Instance2[SysB]** as the destination Event.
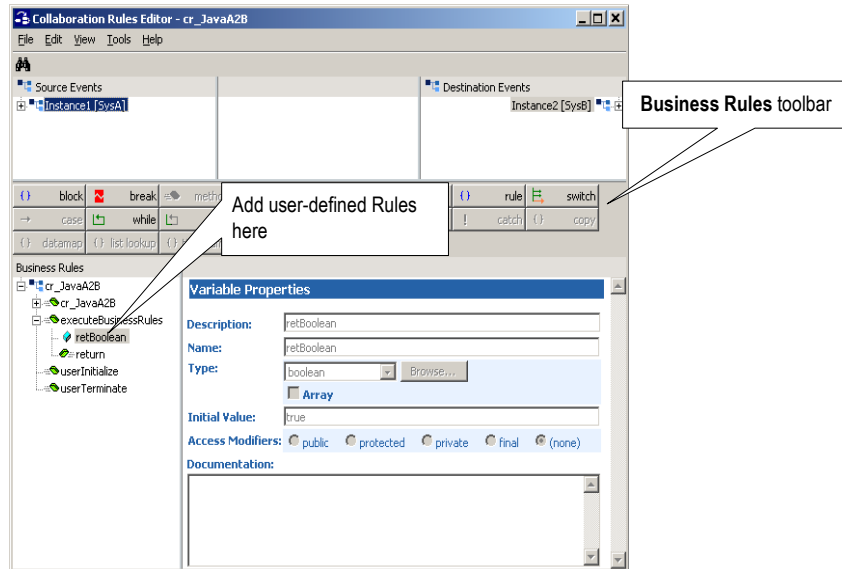
3 On the **View** menu, click **Display Code**.

The Business Rules pane now displays the Java code in addition to the code labels.
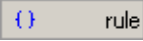
4 In the Source Events and Destination Events panes, click **+** to display the leaf nodes of the Instance1 and Instance2 ETDs.

5 In the Business Rules pane, under the **executeBusinessRules** method, click the **retBoolean** variable.

You add all rules for this scenario within the **executeBusinessRules** method between the **retBoolean** variable and the **return** rule.

6 With the **retBoolean** variable selected, drag the **EmployeeNumber** leaf node from the Source Events pane to the **EmpNo** leaf node in the Destination Events pane. The following happens:

  ◆ A line is drawn connecting the two nodes

  ◆ The **executeBusinessRules** method shows a rule between the **retBoolean** variable and the **return** rule; this rule copies the contents of the **EmployeeNumber** node to the **EmpNo** node

  ◆ The text of the Java code displays in the Rule Properties pane in the **Rule** box

The screen below shows the changes.
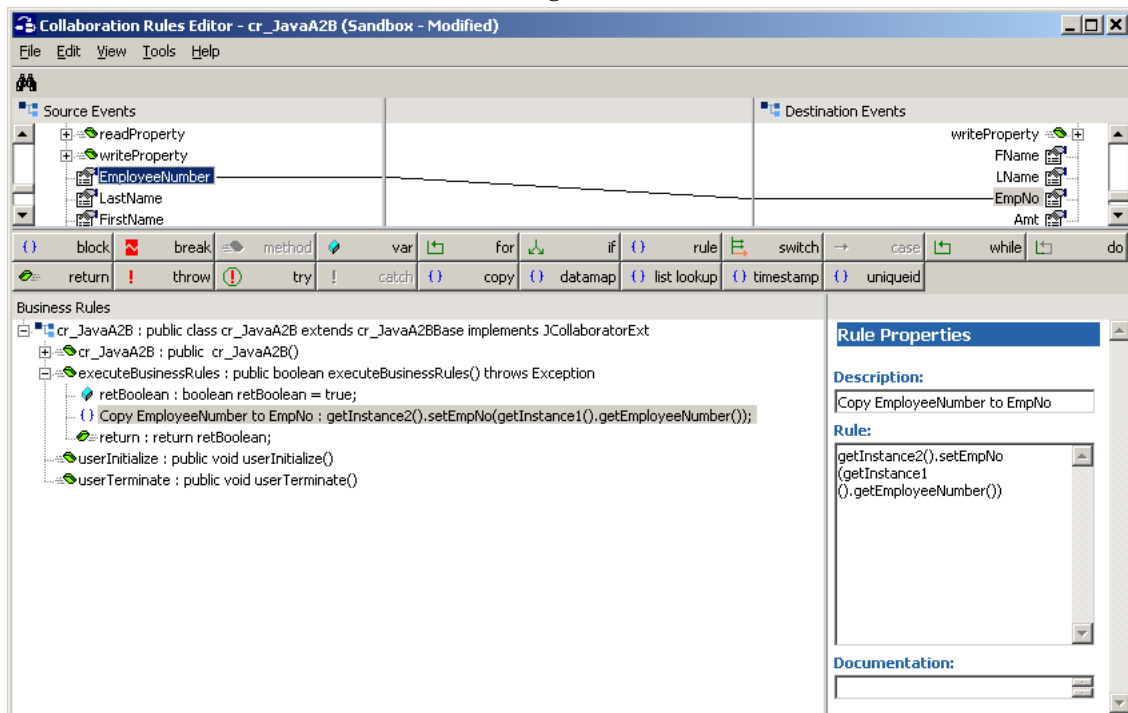
7 Drag the **LastName** leaf node in the Source Events pane to the **LName** leaf node in the Destination Events pane.

8 Drag the **FirstName** leaf node in the Source Events pane to the **FName** leaf node in the Destination Events pane.

9 Click the **Copy FirstName** rule.

10 On the **Business Rules** toolbar, click **Rule** `{}      rule`.

This adds an empty rule to the **executeBusinessRules** method.

11 From the Destination Events pane, drag the **Amt** leaf node to the **Rule** box in the Rule Properties pane.

The **Rule** box displays the following script:

```
getInstance2().setAmt()
```

12 Make sure your keyboard is in **INSERT** mode, click between the ending parentheses (), and type:

```
""+(Integer.parseInt(
```

The **Rule** box displays the following script:

```
getInstance2().setAmt(""+(Integer.parseInt())
```

*If you do not see the closing parenthesis, your keyboard was probably not in **INSERT** mode. Just add a closing parenthesis.*

13 From the Source Events pane, drag the **HoursWorked** leaf node to the location in between the ending parentheses in the **Rule** box.

The **Rule** box displays the following script:

```
getInstance2().setAmt(""+(Integer.parseInt(getInstance1().getHours
Worked())
```

14 Click at the end of the script and type

```
*Integer.parseInt(
```

15 From the Source Events pane, drag the **Rate** leaf node to the end of the script in the **Rule** box.

16 Click in the Rule box at the end of the script and type:

```
)))
```

The screen below shows the completed rule.

Rule:

getInstance2().setAmt(""+(Integer.parseInt(getInstance1().getHoursWorked())*Integer.parseInt(getInstance1().getRate()))))
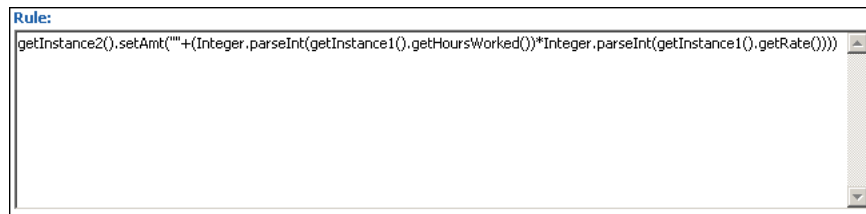
This rule multiplies the amounts from **HoursWorked** and **Rate** in the Source Event pane and copies the result to the **Amt** leaf node in the Destination Event pane.

17 On the **File** menu, click **Save**.

18 Click **Save** to save the **cr_JavaA2B.xpr** file.

19 On the **File** menu, click **Compile**.

The **Compile** box shows "Compile Completed" if the script compiled without errors as shown below.
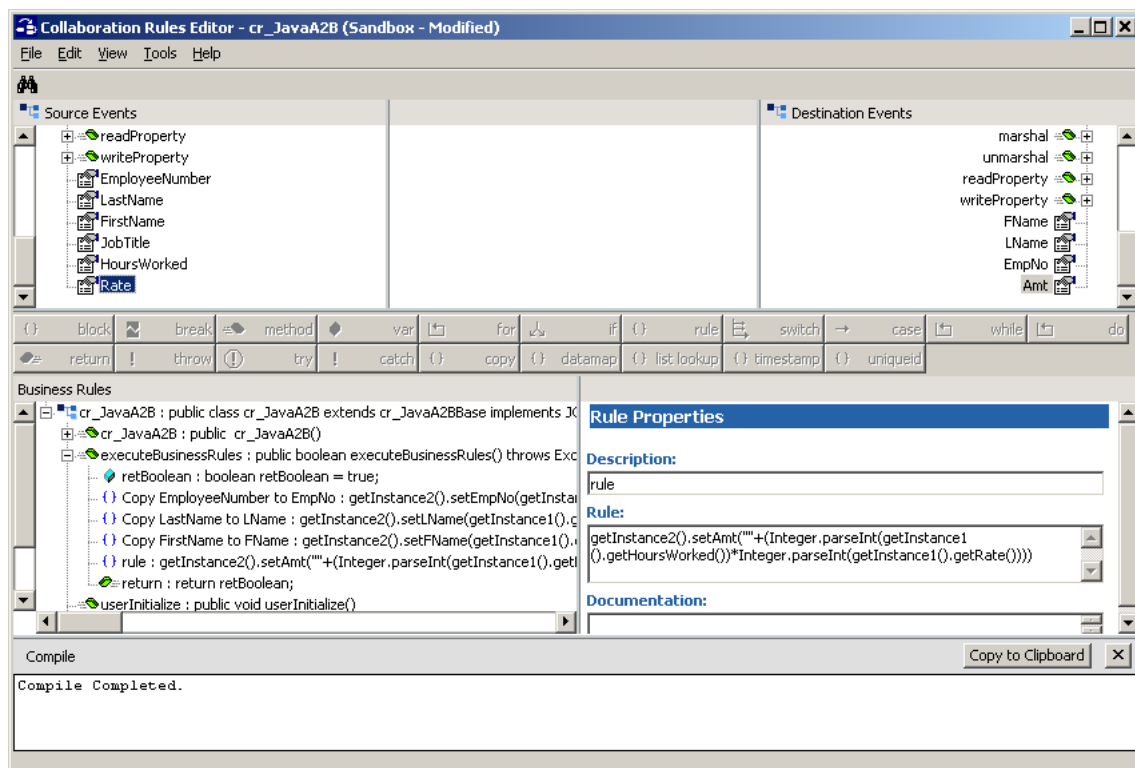
If the Compile box lists errors, fix them before you continue.

20 On the **File** menu, click **Exit**.

The Collaboration Editor closes. The **Collaboration Rules - cr_JavaA2B Properties** dialog box shows the paths for the Collaboration Rules component and the initialization file as shown in the screen below.
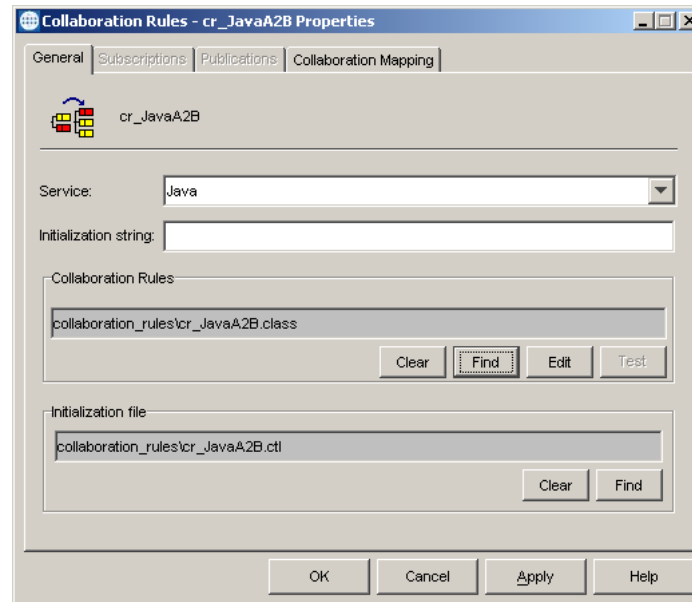
21 Click **OK**.

3.6 Adding the e*Ways and JMS e*Way Connection

After creating the ETDs and Collaborations, you add the e*Ways and the JMS e*Way Connection. This section describes how you add those components.

As shown in the figure below, the JavaE2E schema has the following e*Ways and e*Way Connections:

- **ew_FileIn** File e*Way
- **ew_JavaA2B** Multi-Mode e*Way
- **ew_FileOut** File e*Way

- JMS e*Way Connection
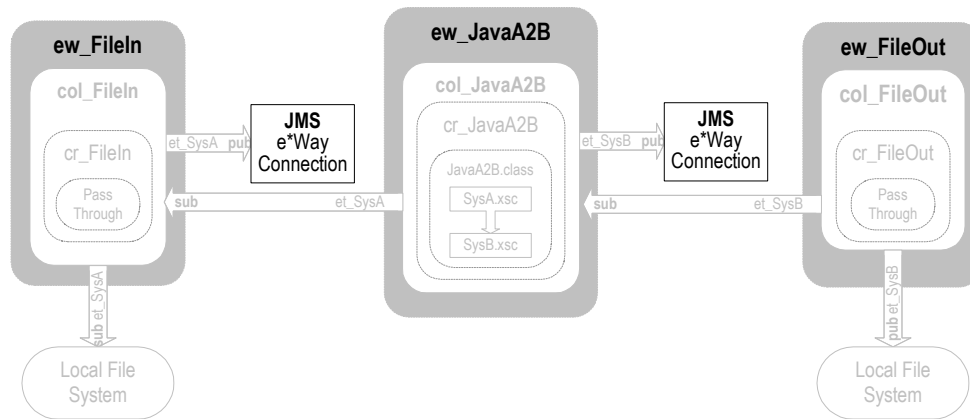
File e*Ways connect to external file systems to read and write files.

3.6.1 Adding the ew_FileIn File e*Way

To add the ew_FileIn File e*Way

1. In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder, and then click the Control Broker (*hostname*_**cb**).

2. Click **Create a New e*Way** .

3. In the **New e*Way Component** box, type **ew_FileIn** and click **OK**.

   This adds the **ew_FileIn** e*Way to the Contents pane.

4. In the **Components** tab, double-click the **ew_FileIn** e*Way.

   The **ew_FileIn - Properties** dialog box appears.

5. Under **Executable file**, click **Find**.

6. In the **File Selection** dialog box, double-click **stcewfile.exe**.

   This adds **bin\stcewfile.exe** file as the executable file, making the component a File e*Way.

7. Under **Configuration file**, click **New**.

The **Edit Settings** dialog box appears as shown below.

8 In the **Goto Section** list, click **Poller (inbound) settings**.

9 In the **PollDirectory** box, type *drive:\eGatefolder***Client\Data\JavaE2E** and press ENTER.

where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client\Data\JavaE2E.

This adds the **JavaE2E** directory as the directory to be polled as shown in the screen below.
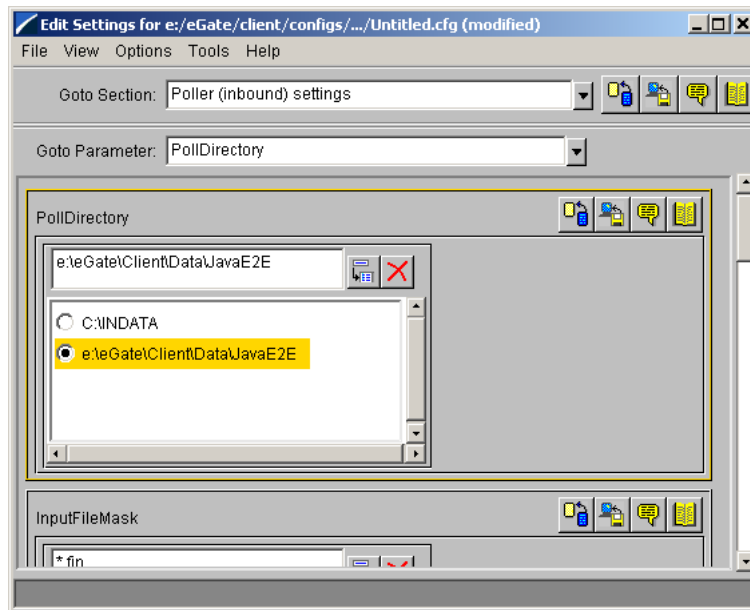
10 On the **File** menu, click **Save**.

11 In the **Save As** dialog box, click **Save** to accept the default file name **ew_FileIn.cfg** and save the file.

12 On the **File** menu, click **Close** to close the **Edit Settings** dialog box.

This adds **configs\stcewfile\ew_FileIn.cfg** under **Configuration file** as shown below.
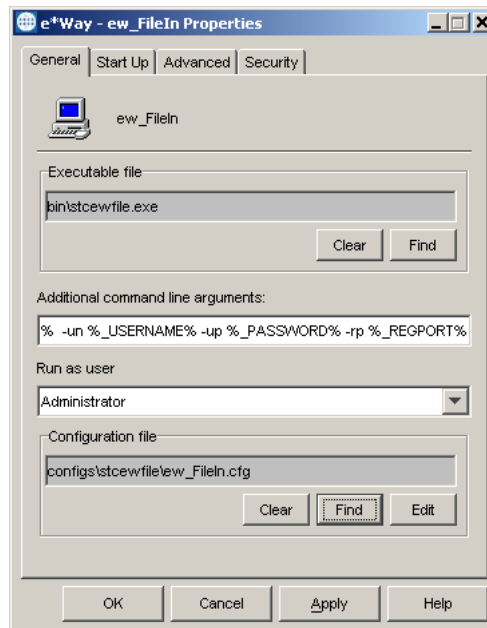
13 Click the **Start Up** tab and click **Start automatically**.

14 Click **OK** to close the **e*Way - ew_FileIn Properties** dialog box.

3.6.2 Adding the ew_FileOut File e*Way

To add the ew_FileOut File e*Way

1 Use steps 1 through 7 from **"Adding the ew_FileIn File e*Way" on page 27** to add another File e*Way named **ew_FileOut**.

The **Edit Settings** dialog box appears.

2 Under **General Settings**, click **NO** under **AllowIncoming**, and **YES** under **AllowOutgoing**.

3 In the **Goto Section** list, click **Outbound (send) settings**.

4 In the **OutputDirectory** box, type *drive:\eGatefolder***\Client\Data\JavaE2E** and press ENTER.

where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client\Data\JavaE2E.

5 In the **OutputFileName** box, type **JavaE2Eoutput%d.dat** and press ENTER.
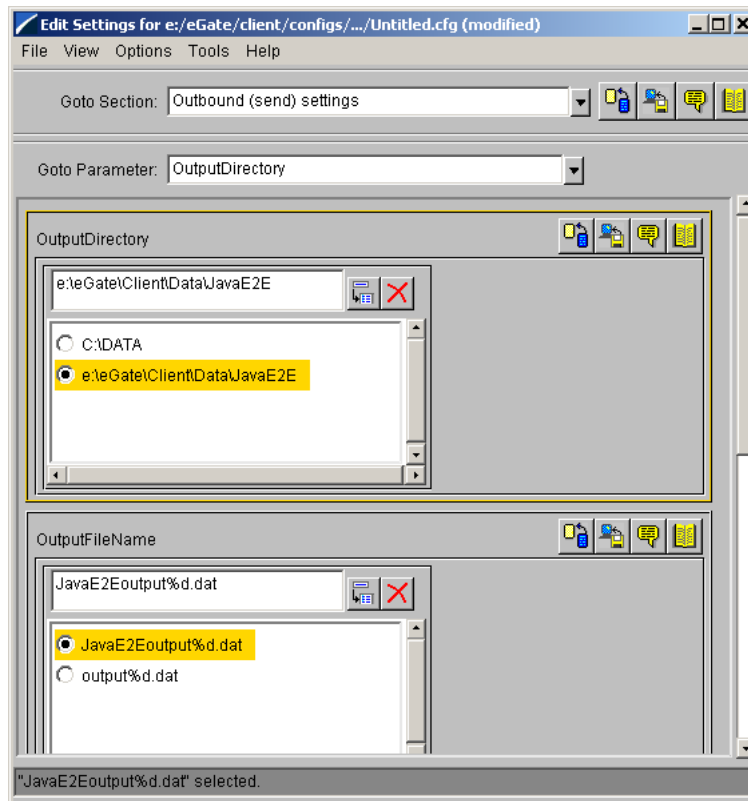
6 On the **File** menu, click **Save**.

7 In the **Save As** dialog box, click **Save** to accept the default file name
(**ew_FileOut.cfg**) and save the file.

8 On the **File** menu, click **Close** to close the **Edit Settings** dialog box.

9 Click the **Start Up** tab, and then select the **Start automatically** check box.

10 Click **OK** to close the **e*Way - ew_FileOut Properties** dialog box.

3.6.3 Adding the ew_JavaA2B Multi-Mode e*Way

The procedure below describes how you add the ew_JavaA2B Multi-Mode e*Way. You
configure this e*Way with a remote debugging port number. For remote debugging to
work, you must enable in-schema debugging on the Participating Host. For details,
refer to the *e*Gate Integrator Installation Guide*.

To add the ew_JavaA2B Multi-Mode e*Way

1 In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating
Hosts** folder, and then click the Control Broker (*hostname_cb*).

2 Click **Create a New e*Way** .

3 In the **New e*Way Component** box, type **ew_JavaA2B** and click **OK**.

This adds the **ew_JavaA2B** e*Way to the Contents pane.

4 In the **Components** tab, double-click the **ew_JavaA2B** e*Way.

The **ew_JavaA2B - Properties** dialog box appears.

5 Under **Configuration file**, click **New**.

The **Edit Settings** dialog box appears.

6 In the **Goto parameters** list, click **Remote debugging port number**.

7 In the **Remote debugging port number** box, type **8000** and press ENTER.
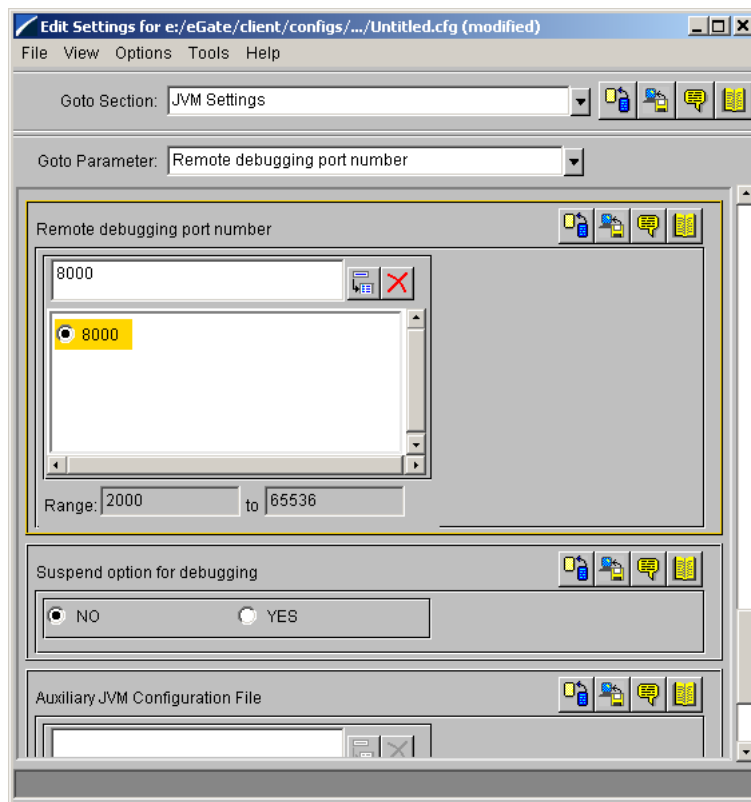
8 On the **File** menu, click **Save**.

9 In the **Save As** dialog box, click **Save** to accept the default file name **ew_JavaA2B.cfg** and save the file.

10 On the **File** menu, click **Close** to close the **Edit Settings** dialog box.

11 Click the **Start Up** tab and click **Start automatically**.

12 Click **OK** to close the **e*Way - ew_JavaA2B Properties** dialog box.

3.6.4 Configuring the IQ Manager

To configure the IQ Manager

1 In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder, then double-click the IQ manager (*hostname*_**iqmgr**).

The **IQ Manager - Properties** dialog box appears.

2 Click the **Start Up** tab and click **Start automatically**.

3 Click **OK** to close the **IQ Manager - Properties** dialog box.

3.6.5 Adding the JMS e*Way Connection

To add the JMS e*Way Connection

1 In the **Components** tab of the Enterprise Manager, click the **e*Way Connections** folder.

2 Click **Create a New e*Way Connection** [icon].

3 In the **Name** box, type **JMS** and click **OK**.

4 Double-click **JMS**.

The **e*Way Connection - JMS Properties** dialog box displays.
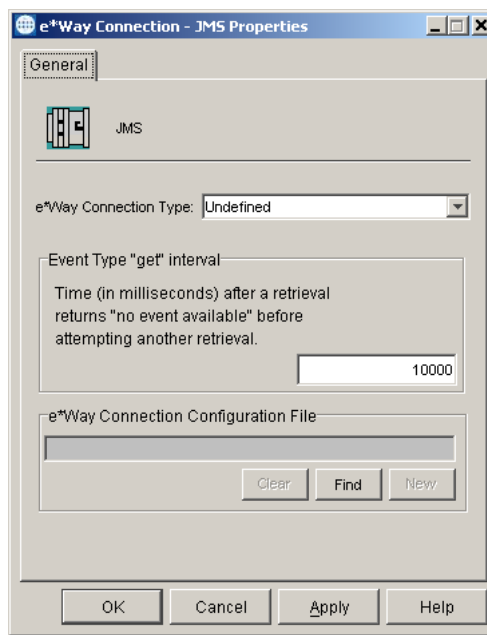
5 In the **e*Way Connection Type** list, click **SeeBeyond JMS**.

6 Click **New**.

7 In the **JMS IQ Manager** box, click your IQ Manager (*hostname*_**iqmgr**).

The screen below shows an example

8   Click **OK** to close the **New JMS e*Way Connection** dialog box.

9   Click **OK** to close the **e*Way Connection - JMS Properties** dialog box.

3.7   Adding the Collaborations

e*Ways use Collaborations to route the data through e*Gate. Typically, you configure the Collaborations in upstream to downstream order: the **col_FileIn** Collaboration must be configured before configuring either the **col_JavaA2B** or **col_FileOut** Collaborations.

The figure below shows the Collaborations to the remainder of the parts that make up the complete schema.
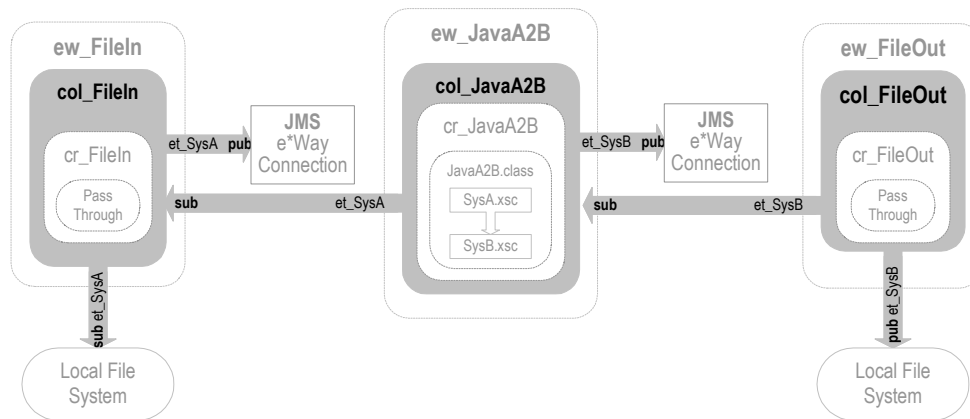
3.7.1   Adding the col_FileIn Collaboration

The procedure below describes how you add the **col_FileIn** Collaboration. The **col_FileIn** Collaboration brings the data into e*Gate.

To add the col_FileIn Collaboration

1   In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder and the Control Broker (*hostname_cb*).

2   Click the **ew_FileIn** e*Way.

3   Click **Create a New Collaboration** .

4   In the **New Collaboration Component** dialog box, type **col_FileIn** and click **OK**.

5   Double-click **col_FileIn**.

The **Collaboration - col_FileIn Properties** dialog box displays.

6   In the **Collaboration Rules** list, click **cr_FileIn**.

7   Under **Subscriptions**, click **Add**.

This adds a row under **Subscriptions**.

8   Double-click the new row in the **Instance Name** column and click **JavaPassThroughIn**.

9   Double-click the row in the **Event Type** column and click **et_SysA**.

10   Double-click the row in the **Source** column and click **<EXTERNAL>**.

11   Under **Publications**, click **Add**.

12   If a message appears regarding journal expiration time, click **No**.

13   Double-click the row in the **Event Type** column and click **et_SysA**.

14   Double-click the row in the **Destination** column and click **JMS**.

The **Collaboration - col_FileIn Properties** dialog box now displays as follows:

15   Click **OK** to close the **Collaboration - col_FileIn Properties** dialog box.

3.7.2 Adding the col_JavaA2B Collaboration

The procedure below describes how you add the **col_JavaA2B** Collaboration. The **col_JavaA2B** Collaboration changes the data from the **et_SysA** Event Type to the **et_SysB** Event Type.

To add the col_JavaA2B Collaboration

1   In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder and the Control Broker (*hostname*_**cb**).

2   Click the **ew_JavaA2B** e*Way.

3   Click **Create a New Collaboration** .

4   In the **New Collaboration Component** dialog box, type **col_JavaA2B** and click **OK**.

5   Double-click **col_JavaA2B**.

6   In the **Collaboration Rules** list, click **cr_JavaA2B**.

7   Under **Subscriptions**, click **Add**.

8   Double-click the new row in the **Instance Name** column and click **Instance1**.

9   Double-click the **Event Type** column and click **et_SysA**.

10 Double-click the **Source** column and click **JMS**.

11 Under **Publications**, click **Add**.

12 Double-click the new row in the **Instance Name** column and click **Instance2**.

13 Double-click the **Event Type** column and click **et_SysB**.

14 Double-click the **Destination** column and click **JMS**.

The **Collaboration - col_JavaA2B Properties** dialog box now displays as follows:
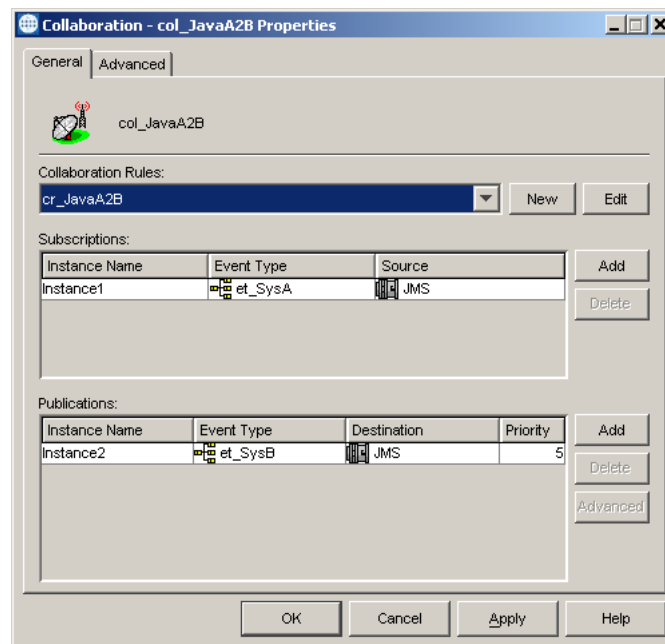
15 Click **OK** to close the **Collaboration - col_JavaA2B Properties** dialog box.

3.7.3 Adding the col_FileOut Collaboration

The procedure below describes how you add the **col_FileOut** Collaboration. The **col_FileOut** Collaboration sends the transformed data out of e*Gate.
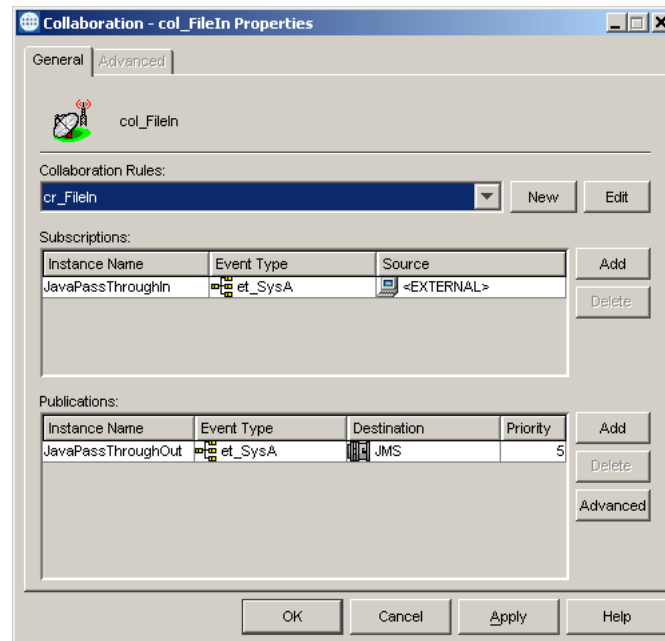
To add the col_FileOut Collaboration

1   In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder and the Control Broker (*hostname_***cb**).

2   Click the **ew_FileOut** e*Way.

3   Click **Create a New Collaboration** .

4   In the **New Collaboration Component** dialog box, type **col_FileOut** and click **OK**.

5   Double-click **col_FileOut**.

6   In the **Collaboration Rules** list, click **cr_FileOut**.

7   Under **Subscriptions**, click **Add**.

8   Double-click the new row in the **Instance Name** column and click **JavaPassThroughIn**.

9   Double-click the **Event Type** column and click **et_SysB**.

10   Double-click the **Source** column and click **JMS**.

11   Under **Publications**, click **Add**.

12   Double-click the **Event Type** column and click **et_SysB**.

13   Double-click the **Destination** column and click **<EXTERNAL>**.

The **Collaboration - col_FileOut Properties** dialog box now displays as follows:
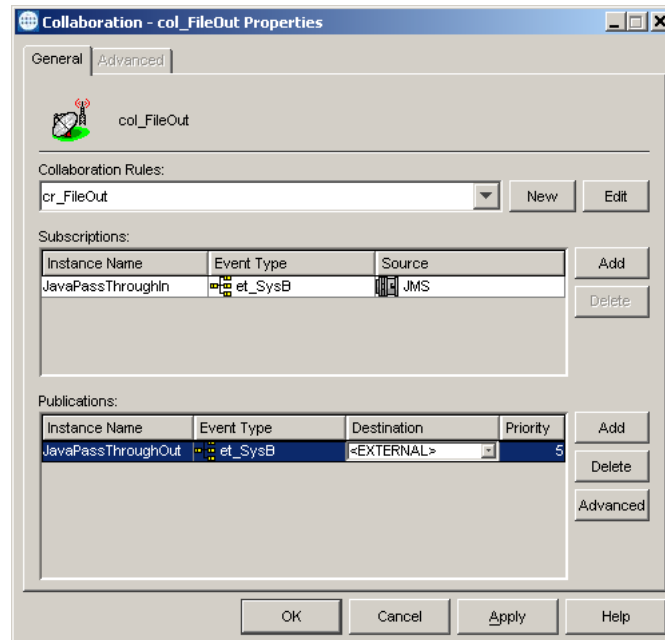
14   Click **OK** to close the **Collaboration - col_FileOut Properties** dialog box.

3.8   Reviewing the Complete Schema

Now that you have completed the schema, use the table below to verify that you have created all the components and that all settings for the components are correct.

Component	Logical Name	Settings
Schema	JavaE2E	
Control Broker	*hostname*_cb	

Component	Logical Name	Settings
IQ Manager	*hostname*_iqmgr	Service = SeeBeyond JMS Config file = *hostname*_iqmgr.cfg Start Up = Auto
Event Type	et_SysA	SysA.xsc
	et_SysB	SysB.xsc
Java ETD	SysA.xsc	Package Name = SysApackage
	SysB.xsc	Package Name = SysBpackage
Collaboration Rules	cr_FileIn	Service = Java JavaPassThroughIn GenericInEvent.ssc Trigger JavaPassThroughOut GenericOutEvent.ssc
	cr_JavaA2B	Service = Java Instance1 SysA.xsc In Trigger Instance2 SysB.xsc Out
	cr_FileOut	Service = Java JavaPassThroughIn GenericInEvent.ssc Trigger JavaPassThroughOut GenericOutEvent.ssc
Java Collaboration Rules Class	cr_JavaA2B.class	Source = Instance1 Destination = Instance2
Inbound File e*Way	ew_FileIn	Executable = stcewfile.exe Config file = ew_FileIn.cfg Start Up = Auto Collaboration = col_FileIn
Outbound File e*Way	ew_FileOut	Executable = stcewfile.exe Config file = ew_FileOut.cfg Start Up = Auto Collaboration = col_FileOut
Multi-Mode e*Way	ew_JavaA2B	Executable = stceway.exe Config file = ew_JavaA2B.cfg Start Up = Auto Collaboration = col_JavaA2B
JMS e*Way Connection	JMS	Service = SeeBeyond JMS Config file = jms.cfg
Collaborations	col_FileIn	Collab Rule = cr_FileIn Subscription = et_SysA from <EXTERNAL> Publication = et_SysA to JMS
	col_JavaA2B	Collab Rule = cr_JavaA2B Subscription = et_SysA from JMS Publication = et_SysB to JMS
	col_FileOut	Collab Rule = cr_FileOut Subscription = et_SysB from JMS Publication = et_SysB to <EXTERNAL>

3.9   # Creating the Input Data File

Follow the procedure below to create the sample input data file. The schema uses this file to bring data into e*Gate.

To create the input data file

1 Use a text editor such as Notepad to create a text file with the following information:

```
10000,Contrary,Mary,President,60,100
10001,Horner,Jack,Line Operator,40,20
10002,Seashells,Sally,Consultant,20,75
10003,Dumpty,Humpty,Manager,50,40
10004,Winky,William,Marketing VP,40,60
```

2 In Windows Explorer, open the *drive:\eGatefolder* **Client** folder, where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client.

3 In the **Client** folder, create a folder named **Data**.

4 Open the data folder and create a folder named **JavaE2E**.

5 Save the file as **JavaE2Einput.txt**.

3.10   # Starting the Schema

To start the schema

1 Use the following command to start the Control Broker from a command line.

```
stccb.exe -rh hostname -rs JavaE2E -ln hostname_cb -un username
-up password
```

2 Start the e*Gate Monitor and open the **JavaE2E** schema.

3 Expand the **Schema - JavaE2E** folder, the **Participating Hosts** folder, and then click the Control Broker (*hostname_cb*).

4   Verify that all the components are running by checking that the Status Name for
     each component lists **Up** as shown in the screen below.
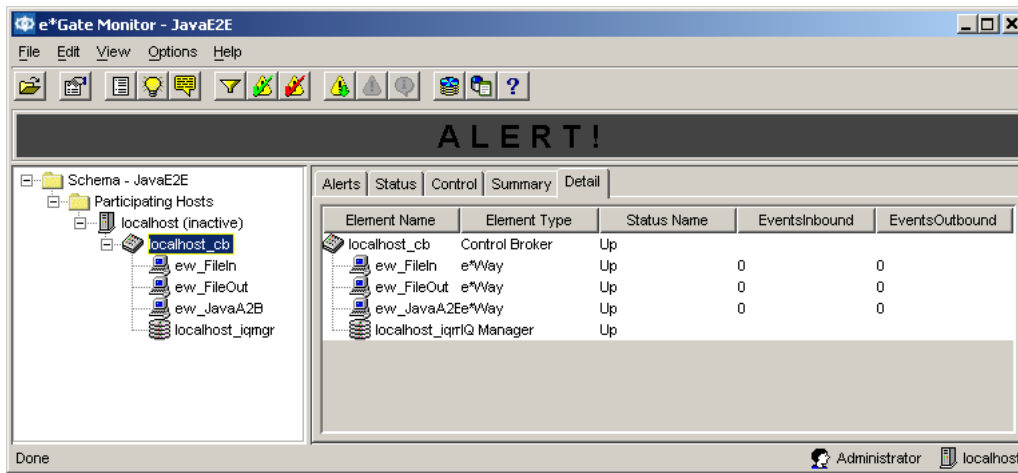
3.11 Testing the Schema

To test the schema

1   In Windows Explorer, open the *drive***:***eGatefolder***\\Client\\Data\\JavaE2E** folder,
     where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder
     where e*Gate is installed. For example: c:\\eGate\\Client\\Data\\JavaE2E.

2   Change the file extension of **JavaE2Einput.txt** to **.fin**.

3   Click **Yes** when the confirmation message appears.

4   Verify that the extension changes to **.~in** indicating that the **ew_FileIn** e*Way
     retrieved the file.

     Almost immediately, the output file **JavaE2Eoutput#.dat** appears in the folder,
     indicating the schema ran successfully.

5   Verify that the output looks like that shown in the screen below.
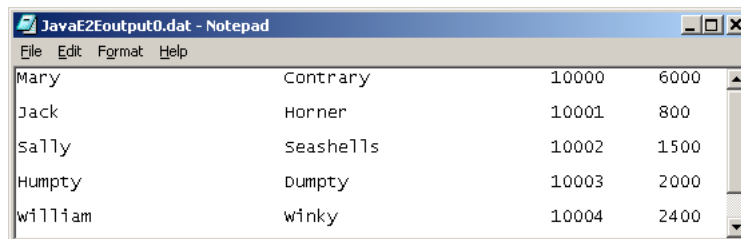
3.12 Troubleshooting

If you encounter a problem and are unable to solve it with **"Reviewing the Complete Schema" on page 36**, refer to **"Debugging and Log Files" on page 92** to start the trace or error log component in your schema.

Error	Possible Solution
unreachable statement	In the **executeBusinessRules** method, place the **return retBoolean** rule below the user-defined rules

Creating an ELS End-to-End Scenario

This chapter provides step-by-step procedures for creating and testing an end-to-end e*Gate scenario that uses Event Linking and Sequencing (ELS). This schema addresses the following scenario: a company needs its receiving system to send a message to its purchasing system when all items of an order with multiple items are received. At that point, the purchasing system issues payment for the purchase order.

This chapter describes the business challenge and how you can create an ELS schema in e*Gate to have the receiving system track the receipt of order items and to send a message to the purchasing system when the order is received in its entirety.

In This Chapter

4.1 Presenting the Business Challenge

The ELS end-to-end scenario described in this chapter provides a solution to the following business challenge:

Purchasing buys office supplies for the company. The purchasing systems generates orders with multiple line items. Receiving processes the items as they are delivered. Typically, only individual line items are delivered, not entire purchase orders.

Purchasing only pays purchase orders that are completely fulfilled. The company needs a way to track whether an entire purchase order has been filled.

The figure below shows the relationship between the receiving and purchasing systems.
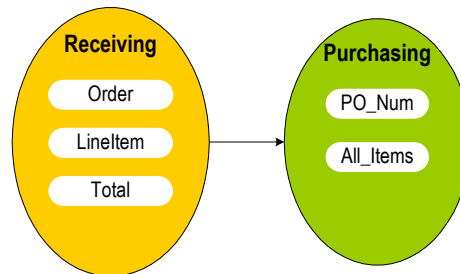
4.2 Presenting the e*Gate Solution

The ELS scenario shows you how e*Gate's ELS feature can correlate the individual line items taken in by receiving so that when all line items for a specific purchase order have been received, a combined Event is sent to purchasing. Purchasing uses the receipt of this Event to trigger payment of the purchase order.

The figure below shows the e*Gate solution to this business challenge.
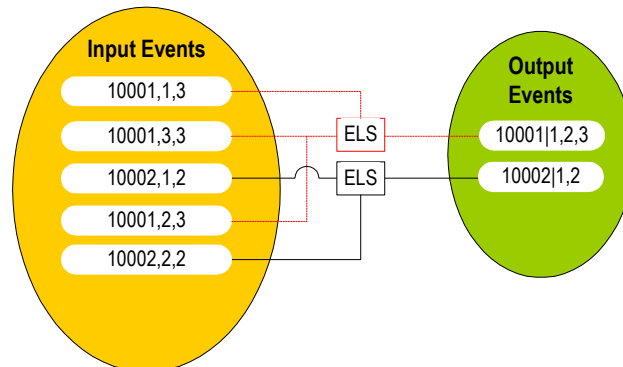
The proposed e*Gate solution makes use of e*Gate enhanced Java Collaboration Service and the ELS methods added to the Java Collaboration to combine the data from Events that have the same order number into a single Event. e*Gate is very flexible about where the actual ELS processing can occur as the data moves from the receiving system to the purchasing system. Our solution uses a Java Collaboration as the main transformation component and two File e*Ways to bring data into and move data out of e*Gate.

The figure below shows the complete e*Gate schema.
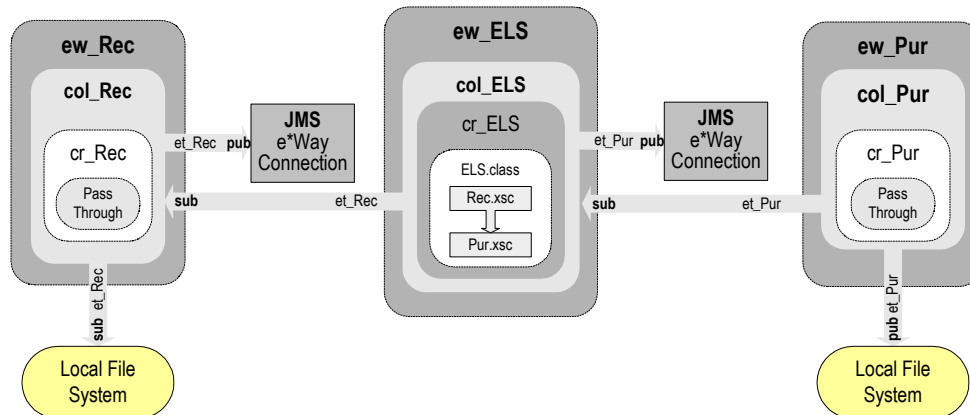
In this figure, you see the following process:

1 The **ew_Rec** e*Way brings data from the Receiving System into e*Gate.

The **col_Rec** Collaboration in the **ew_Rec** e*Way subscribes to a location on the local file system. It polls this location for a text file with extension **.fin** containing data from the Receiving System, packages the data as an **et_Rec** Event, and then publishes the Event to the JMS e*Way Connection.

2 The **ew_ELS** e*Way combines Events with matching order numbers into a single Event.

The **col_ELS** in the **ew_ELS** subscribes to **et_Rec** Events published by the **JMS** e*Way Connection. It uses an ELS-enabled Java Collaboration Rules component **cr_ELS** to temporarily store **et_Rec** Events. Once all the Events associated with a specific order are received, **cr_ELS** publishes a combined Event, **et_Pur,** to the JMS e*Way Connection. This Collaboration Rules component uses the **ELS.class** file that implements the business logic.

3 The **ew_Pur** e*Way writes the combined Event out to local file system.

The **col_Pur** Collaboration in the **ew_Pur** e*Way subscribes to **et_Pur** Events published by the JMS e*Way Connection. The **cr_Pur** Collaboration Rules component uses the Java Pass Through service to move the data without modifying it. When a **et_Pur** Event is retrieved, the e*Way packages it as a text file and writes it to the specified location on the local file system, completing the end-to-end scenario.

4.3 Creating the Schema

Before you start creating the schema, make sure that you have the following system requirements:

- e*Gate Registry Host, Participating Host, and GUIs installed on a single system
- The system is running the Windows operating system

- The system has Internet Explorer 5.x or later installed

- The system has Java 2 SDK, Standard Edition, v 1.3.1_02 (JDK 1.3.1_02) installed

For more information about the installation requirements, refer to **"e*Gate Installation Requirements" on page 11.**

To create the schema

1   Start the e*Gate Enterprise Manager and log in as **Administrator** (or another user with administrator privileges) to the appropriate Registry Host.

2   In the **Open Schema on Registry Host** dialog box, click **New**.

3   In the **Enter New Schema Name** box, type **ELSE2E**, and click **Open**.

    The Enterprise Manager displays the new **ELSE2E** schema.

4   Click the **Components** tab. You perform all configuration steps in the **Components** tab.
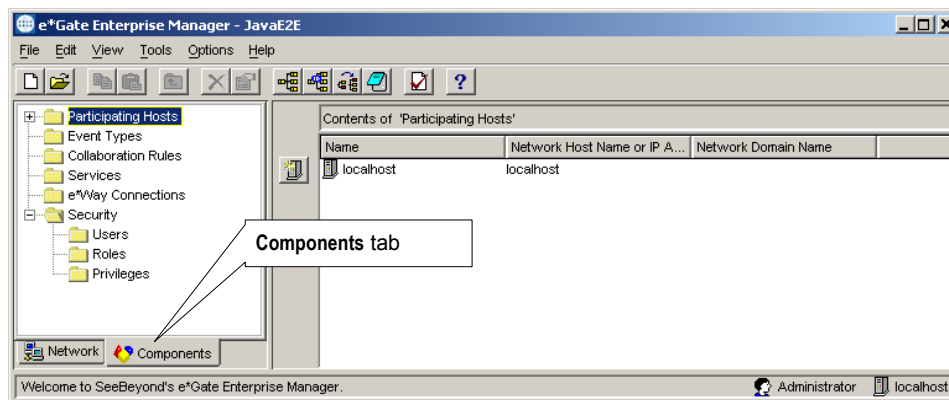
4.4   Creating the Event Types and Java ETDs

This scenario uses two Event Types. These Event Types require their own ETDs. The first Event Type, **et_Rec**, models the format of the data sent by the receiving system. The second Event Type, **et_Pur**, models the format used by the purchasing system.

The figure below shows where these parts fit into the schema.
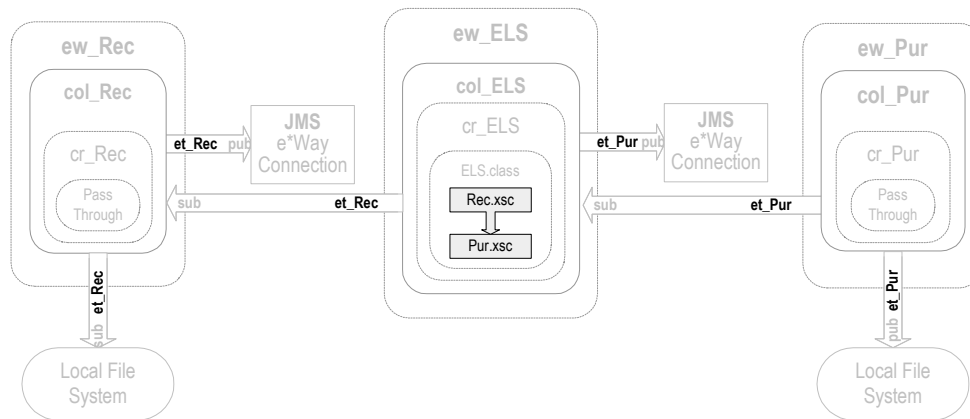
4.4.1 Creating the Rec.xsc ETD

The procedure below describes how you create the Rec.xsc ETD. Before you start, you must have already created the ELSE2E schema as described in **"Creating the Schema" on page 43**.

To create an ETD, you specify the ETD name, and the ETD's root node name and a package name.

The package name is the name of the collection of Java classes that make up the ETD. This group of Java classes is the result of the ETD creation process. The root node name and package name **cannot** be the same.

To create the Rec.xsc ETD

1 In the e*Gate Enterprise Manager, click **Default Editor** on the **Options** menu.

2 Click **Java** if necessary and click **OK**.

3 In the **Components** tab, click the **Event Types** folder.

4 Click **Create a New Event Type** .

5 In the **Name** box, type **et_Rec** and click **OK**.

This adds **et_Rec** to the Event Types pane.

6 In the Event Types pane, double-click **et_Rec**.

The **Event Type - et_Rec Properties** dialog box appears.

7 Click **New**. The **ETD Editor** window appears.

8 Click **New** .

The **New Event Type Definition** window appears.

9 Double-click **Custom ETD Wizard** .

The **Custom ETD Wizard - Introduction** dialog box appears.

10 Click **Next**.

The **Custom ETD Wizard - Step 1** dialog box appears.

11   In the **Root Node Name** box, type **Rec**.

12   In the **Package Name** box, type **Recpackage**.

13   Click **Next**, review the summary information, and click **Finish**.

The screen below shows the new Rec.xsc ETD.
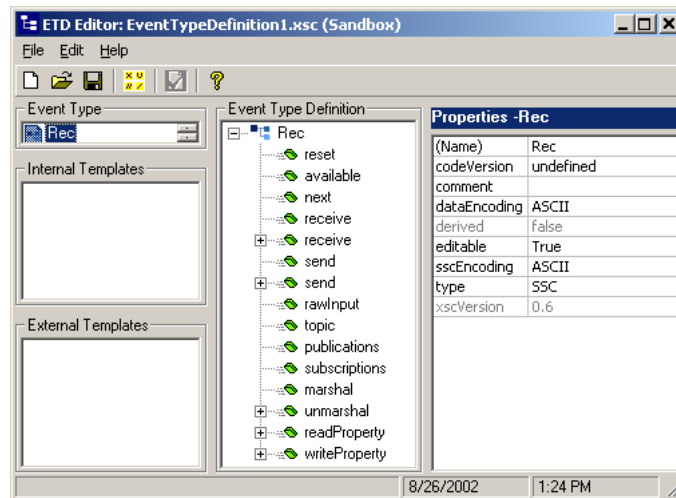
Adding Rec.xsc ETD Nodes

The procedure below describes how you add nodes to the Rec.xsc ETD. Before you start, you must have already created ETD as described in **"Creating the Rec.xsc ETD" on page 45**.

To add Rec.xsc ETD nodes

1   In the Event Type Definition pane, right-click **Rec**, point to **Add Field**, and then click **As Child Node**.

This adds a **Field1** node under the **Rec** root node.

2   Right-click the **Field1** node and click **Rename**.

3   Type **Order** and press **ENTER**.

4   Repeat steps 1 through 3 to add the following nodes:

   ◆ LineItem

   ◆ Total

5   With the **Total** node selected, click **javaType** in the Properties pane and click **int**.

The screen below shows the completed Rec.xsc ETD
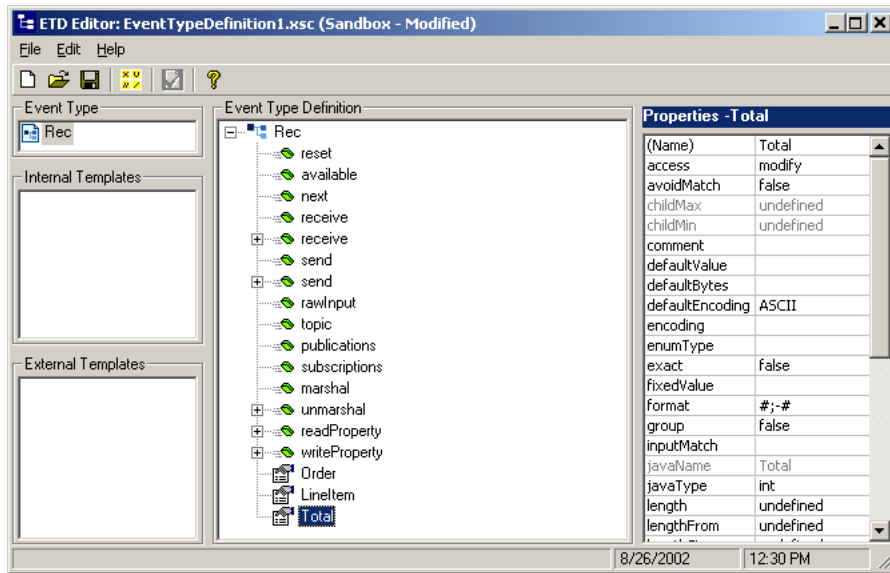
6   Click **Global Delimiters** [icon].

7   Click **Add Level**.

8   Click **Add Single End Delimiter**.

9   Click **Value** in the Properties pane and enter ",", (a comma, no quotes) as shown below.
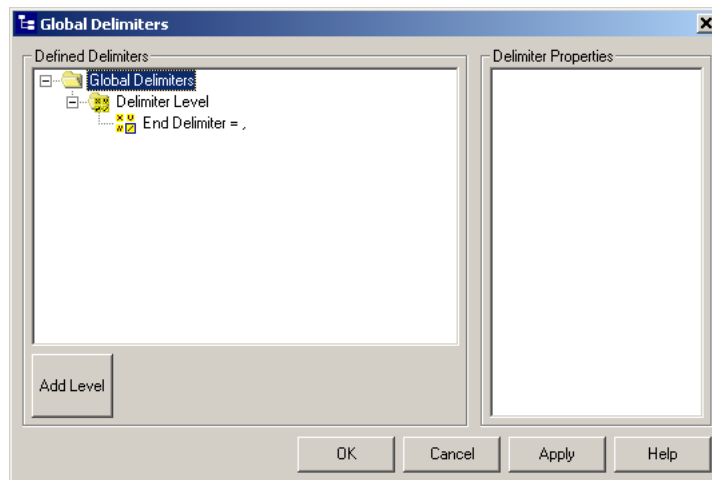
10   Click **OK**.

11   On the **File** menu, click **Compile and Save**.

12   In the **File name** box, change **EventTypeDefinition1.xsc** to **Rec.xsc** and click **Save**.

This compiles and saves the Rec.xsc ETD as **Rec.xsc** in the **etd** folder.

13   On the **File** menu, click **Close**.

The Java ETD Editor closes and the location of **Rec.xsc** displays the Event Type Definition pane.

14   Click **OK** to close the **Event Type - et_Rec Properties** dialog box.

4.4.2   ## Creating the Pur.xsc ETD

This ETD corresponds to the fixed data structure used by the purchasing system.

Use the procedure in **"Creating the Rec.xsc ETD" on page 45** to create a new Event Type named **et_Pur** and a corresponding Java ETD named Pur.xsc with root node name **Pur** and package name **Purpackage**. Then use the following procedure to add nodes to **Pur.xsc** in the ETD Editor.

Adding Pur.xsc ETD Nodes

The procedure below describes how you add nodes to the Pur.xsc ETD. Before you start, you must have already created ETD as described above.

To add Pur.xsc ETD nodes

1   In the Event Type Definition pane in the ETD Editor, right-click the **Pur** node, point to **Add Field**, and then click **As Child Node**.

2   Right-click the **Field1** node and click **Rename**.

3   Type **PONum** and press ENTER.

4   With the **PONum** node selected, click **local delimiters** in the Properties pane and click the ellipsis ⚏ button.

The **Local Delimiters** dialog box appears.

5   Click **Add Single End Delimiter**.

6   Click **Value** in the Properties pane and enter "|" (pipe, no quotes).

7   Click **OK** to close the **Local Delimiter** dialog.

8   In the Event Type Definition pane, right-click **Pur**, point to **Add Field**, and then click **As Child Node**.

9   Right-click the **Field1** node and click **Rename**.

10   Type **AllItems** and press ENTER.

11   With the **AllItems** node selected, click **local delimiters** in the Properties pane and click the ellipsis ⚏ button.

The **Local Delimiters** dialog box appears.

12   Click **Add Single End Delimiter**.

13   Click **Value** in the Properties pane and enter "," (comma, no quotes).

14   Click **OK** to close the **Local Delimiter** dialog.

15   Click **maxOccurs**, and then enter "-l" (minus one, no quotes) to change **AllItems** to a repeating node.

The screen below shows the completed ETD.
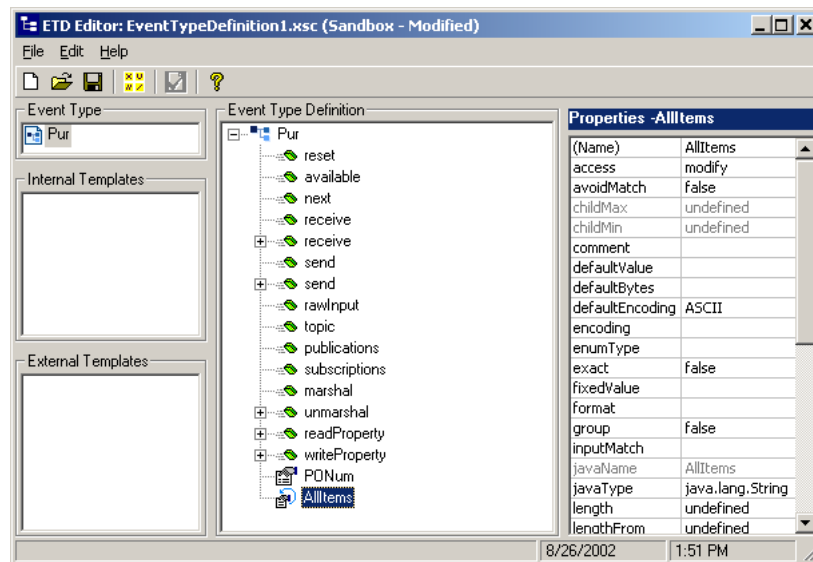
16 On the **File** menu, click **Compile and Save**.

17 In the **File name** box, change **EventTypeDefinition1.xsc** to **Pur.xsc** and click **Save**.

This compiles and saves the Pur.xsc ETD as **Pur.xsc** in the **etd** folder.

18 On the **File** menu, click **Close**.

The Java ETD Editor closes and the location of **Pur.xsc** displays the Event Type Definition pane.

19 Click **OK** to close the **Event Type - et_Pur Properties** dialog box.

4.5   Creating the Collaboration Rules

The ELS end-to-end scenario uses three Collaboration Rules: two Java Pass Through Collaboration Rules and one Java Collaboration Rules component. The Pass Through rules, **cr_Rec** and **cr_Pur**, route the Events through the e*Gate environment. The Java Collaboration Rules component **cr_ELS** combines the individual line item Events into a single complete purchase order Event.

The figure below shows where the Collaboration Rules fit into the schema.
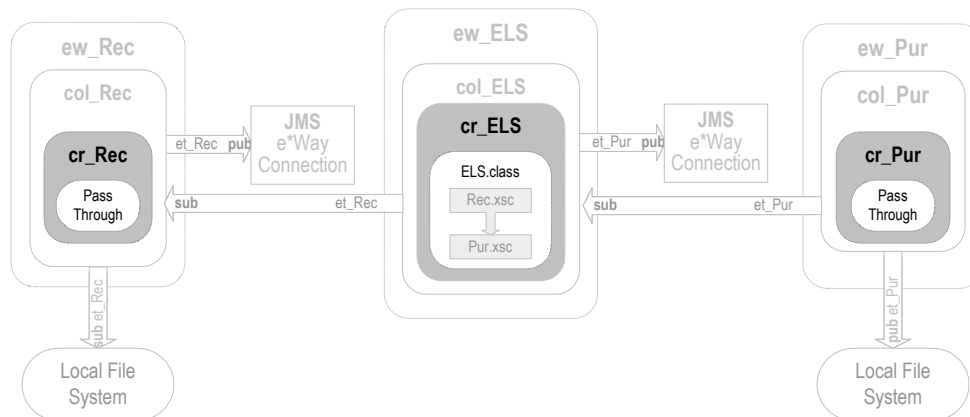
4.5.1 Creating the Java Pass Through Collaboration Rules

The Java Pass Through Collaboration Rules are used to bring data into and move data out of e*Gate. The Pass Through Collaboration Rules do not change the data. The following procedure describes how you create the Java Pass Through Collaborations for this scenario.

To create the Java Pass Through Collaboration Rules

1 In the **Components** tab of the e*Gate Enterprise Manager, click the **Collaboration Rules** folder.

2 Click **Create New Collaboration Rules** .

3 In the **Name** box, type **cr_Rec** and click **OK**.

   cr_Rec displays in the Collaboration Rules pane.

4 Double-click **cr_Rec** in the Collaboration Rules pane.

   The **Collaboration Rules - cr_Rec Properties** dialog box appears.

5 Under **Collaboration Rules**, click **Find**.

6 Double-click **STCLibrary** and double-click **STCJavaPassThrough.class**.

The Collaboration Editor closes. The **Collaboration Rules - cr_Rec Properties** dialog box shows the paths for the Collaboration Rules component and the initialization file as shown in the screen below:
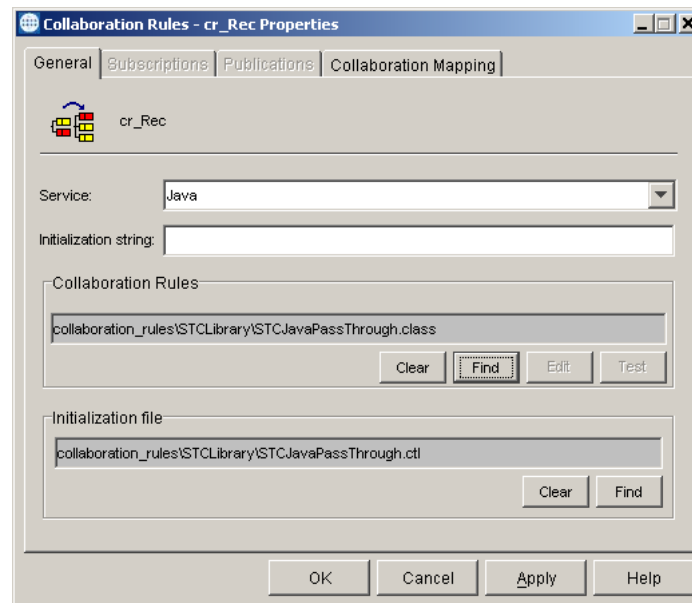
7 Click **OK** to close the **Collaboration Rules - cr_Rec Properties** dialog box.

8 In the Contents pane, right-click the **cr_Rec** Collaboration Rules component.

9 Click **Copy**.

10 Right-click the Contents pane and click **Paste**.

11 Right-click the **cr_Rec_0** Collaboration Rules component and click **Rename**.

12 In the **Name** box, type **cr_Pur** and click **OK**.

4.5.2 Creating the cr_ELS Collaboration Rules Component

This section describes how you create the Java Collaboration Rules component **cr_ELS**. Creating **cr_ELS** consists of the following steps:

1 Adding the component.

2 Configuring the component.

3 Specifying the Java script.

4 Compiling and saving the component.

Adding the cr_ELS Collaboration Rules Component

To add the cr_ELS Collaboration Rules component

1 In the **Components** tab of the e*Gate Enterprise Manager, click the **Collaboration Rules** folder.

2 Click **Create New Collaboration Rules** .

3   In the **Name** box, type **cr_ELS** and click **OK**.

4   Double-click **cr_ELS** in the Collaboration Rules pane.

5   Click the **Collaboration Mapping** tab.

6   Click **Add Instance**.

7   In the **Instance Name** box, type **RootIn** for the instance name.

8   Click **Find** and double-click **Rec.xsc**.

9   Click **Add Instance**.

10   In the **Instance Name**, type **RootOut** for the instance name.

11   Click **Find** for RootOut and double-click **Pur.xsc**.

12   In the **Mode** list for RootOut, click **Out**.

13   Click **Apply**.

The screen below shows what the Collaboration Mapping tab looks like.
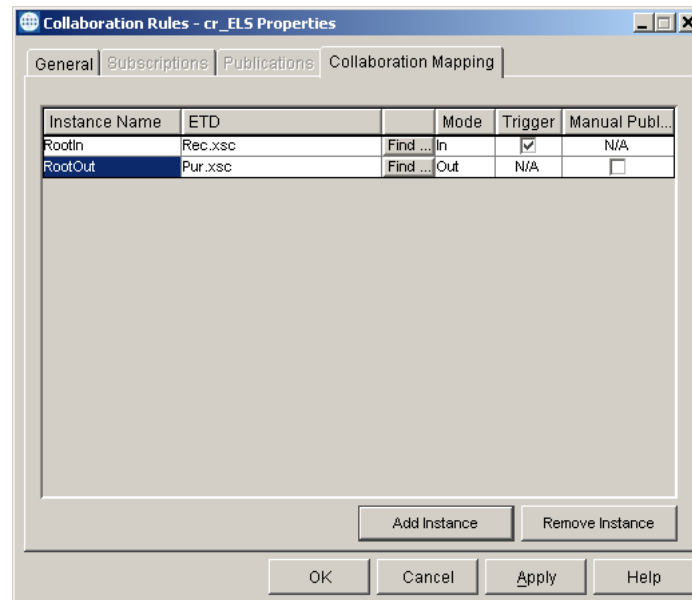
Now that you have created the Collaboration Rules component, you must follow the procedure below to specify the script behind the Collaboration Rules component.

Configuring the cr_ELS Collaboration Rules Component

To configure the cr_ELS Collaboration Rules component

1   In the **Collaboration Rules - cr_ELS Properties** dialog box, click the **General** tab.

2   Click **New**.

The Java Collaboration Editor opens a new Collaboration Rules component with RootIn (Rec) as the source Event and RootOut(Pur) as the destination Event.

3 On the **File** menu, click **Enable ELS**.

4 Click **Yes** to confirm that you want to enable ELS.

This adds the ELS methods **retrieveLinkIdentifier**, **isLinkingComplete**, and **onExpire** to the business rules as shown below.
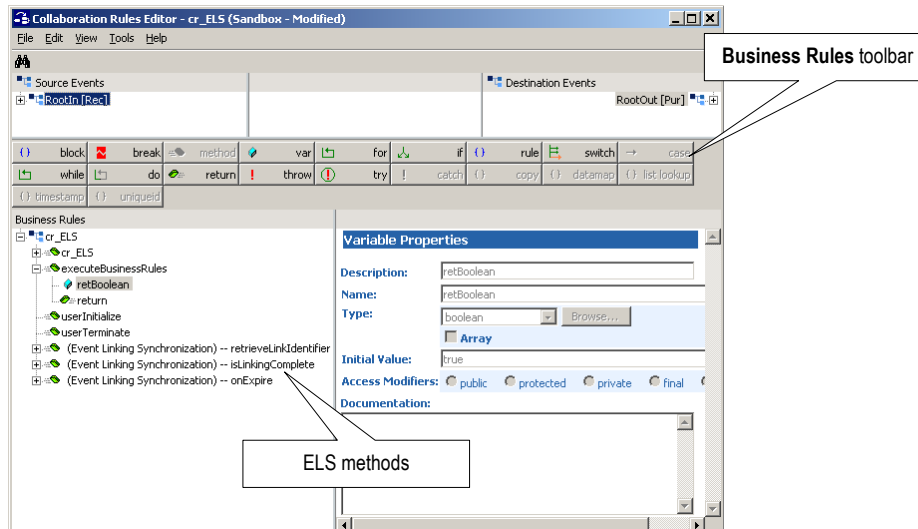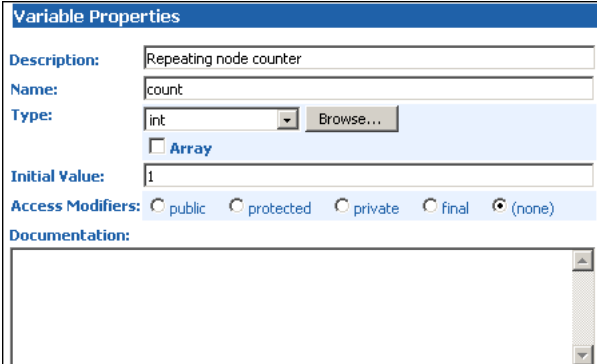
5 In the Source Events and Destination Events panes, click **+** to expand the leaf nodes of the RootIn and RootOut ETDs.

6 On the **Tools** menu, click **ELS Wizard.**

The **ELS Wizard - Introduction** dialog box appears.

7 Click **Next**.

The **ELS Wizard - Step 1** dialog box appears.

8 In the Collaboration Rules Editor behind the ELS wizard, click **Order** in the Source Events pane and drag it into the **Key Value (linkidentifier)** box in the **ELS Wizard - Step 1** dialog box.

9 Click **Next**.

10 Click **Total** in the Source Events pane and drag it into the **Message Count** box in the **ELS Wizard - Step 2** dialog box.

11 Click **Next** to continue.

12 In the **Expiration Time** box, type **10000**.

13 Click **Finish**. This closes the ELS wizard.

Specifying the executeBusinessRules Method

The **executeBusinessRules** method is called whenever the **isLinkingComplete** method returns a Boolean true. It is used to create the destination Events.

To specify the executeBusinessRules method

1 In the Business Rules pane of the Collaboration Rules Editor, click **retBoolean** under **executeBusinessRules**.

2 On the **Business Rules** toolbar, click **Var** ⬥ var .

   This adds a variable below **retBoolean**.

3 In the variable properties pane, enter the following information:

Variable Properties	
Description:	Repeating node counter
Name:	count
Type:	int ▼ Browse...
	☐ **Array**
Initial Value:	1
Access Modifiers:	○ public ○ protected ○ private ○ final ◉ (none)
Documentation:	

4 With the **Repeating node counter** variable selected, click **Rule** {} rule on the **Business Rules** toolbar.

5 In the **Description** box in the Rule Properties pane, type **Copy order number**.

6 From the Destination Events pane, drag the **PONum** leaf node to the **Rule** box in the Rule Properties pane.

   The **Rule** box displays the following script:

   ```
   getRootOut().setPONum()
   ```

7 From the Source Events pane, drag the **Order** leaf node to the **Rule** box in between the last parentheses.

   The **Rule** box displays the following script:

   ```
   getRootOut().setPONum(getRootIn().getOrder())
   ```

   This rule copies the purchase order number to the destination Event.

8 With the **Copy order number** rule selected, click **Rule** {} rule on the **Business Rules** toolbar.

9 In the **Description** box in the Rule Properties pane, type **Copy first line item number**.

10 From the Destination Events pane, drag the **AllItems** leaf node to the **Rule** box in the Rule Properties pane.

11 Click **OK** to set the destination node repetition to 0 to indicate where in the repetition the value for **AllItems** falls.

   The **Rule** box displays the following script:
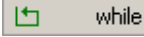
   ```
   getRootOut().setAllItems(0, )
   ```

12 From the Source Events pane, drag the **LineItem** leaf node to the **Rule** box in between the comma and the closing parenthesis. Delete the extra space.

The **Rule** box displays the following script:

```
getRootOut().setAllItems(0,getRootIn().getLineItem())
```

This rule copies the first line item number to the **AllItems** repeating node in the destination Event. Subsequent line item numbers are copied to this node from within the following while rule.

13 With the **Copy first line item number** rule selected, click **While** 🔄 while on the **Business Rules** toolbar.
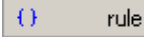
14 In the **Description** box in the While Properties pane, type **Copy additional line item numbers**.

15 From the Source Events pane, drag the **next** leaf node to the **Condition** box in the While Properties pane.

The **Condition** box displays the following script:

```
getRootIn().next()
```

The while rule processes as long as there is another matching Event to retrieve.

16 With the **Copy additional line item number** rule selected, click **Rule** {} rule on the **Business Rules** toolbar.

17 Click **Child**.

18 In the **Description** box in the Rule Properties pane, type **Copy next line item number**.

19 From the Destination Events pane, drag the **AllItems** leaf node to the **Rule** box in the Rule Properties pane.

20 In the Select Repetition Instance, change 0 to **count**.

21 Click **OK**.

The **Rule** box displays the following script:

```
getRootOut().setAllItems(count, )
```

22 From the Source Events pane, drag the **LineItem** leaf node to the **Rule** box in between **count** and the closing parenthesis. Delete the extra space.

The **Rule** box displays the following script:

```
getRootOut().setAllItems(count,getRootIn().getLineItem())
```

This rule copies all remaining line item numbers to the **AllItems** repeating node in the destination Event.

23 With the **Copy next line item number** rule selected, click **Rule** {} rule on the **Business Rules** toolbar.

24 In the **Description** box in the Rule Properties pane, type **Add one to repeating node counter**.

25 In the **Rule** box, type **count++**.

This rule increments the repeating node counter. The count variable is used in the **Copy next line item number** rule to set values in the **AllItems** repeating node.

The screen below shows with completed executeBusinessRules Method (with **Display Code** on the View menu selected).

Compiling and Saving the cr_ELS Collaboration Rules Component

To compile and save the cr_ELS Collaboration Rules component

1   On the **File** menu, click **Save**.

2   In the **Save** dialog box, click Save to save the **cr_ELS.xp**r file to the **collaboration_rules** folder.

3   On the **File** menu, click **Compile**.

The **Compile** box shows "Compile Completed."

4   On the **File** menu, click **Exit**.

The Collaboration Editor closes. The **Collaboration Rules - cr_JavaA2B Properties** dialog box shows the paths for the Collaboration Rules component and the initialization file as shown in the screen below.
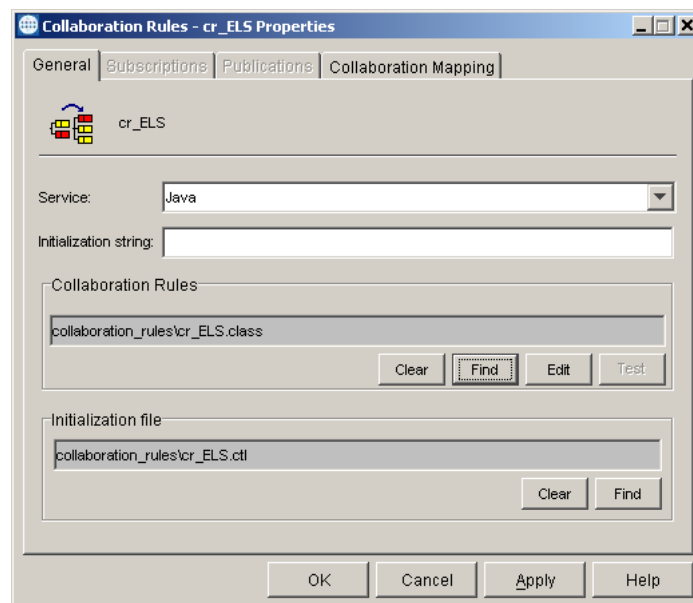
5   Click **OK** to close the **Collaboration Rules - cr_ELS Properties** dialog box.

4.6 Adding the e*Ways and the JMS e*Way Connection

After creating the ETDs and Collaborations, you add the e*Ways and the JMS e*Way Connection. This section describes how you add those components.

As shown in the figure below, the ELSE2E schema has the following e*Ways and e*Way Connections:

- **ew_Rec** File e*Way
- **ew_ELS** e*Way
- **ew_Pur** File e*Way
- JMS e*Way Connection
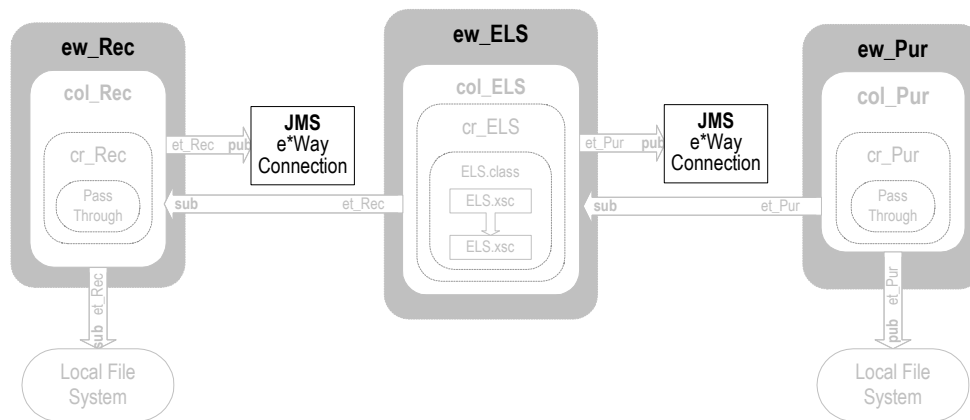
File e*Ways connect to external file systems to read and write files.

4.6.1 Adding the ew_Rec File e*Way

To add the ew_Rec File e*Way

1 In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder, and then click the Control Broker (*hostname*_**cb**).

2 Click **Create a New e*Way** .

3 In the **New e*Way Component** box, type **ew_Rec** and click **OK**.

   This adds the **ew_Rec** e*Way to the Control Broker pane.

4 In the **Components** tab, double-click the **ew_Rec** e*Way.

   The **ew_Rec - Properties** dialog box appears.

5 Under **Executable file**, click **Find**.

6 In the **File Selection** dialog box, double-click **stcewfile.exe**.

   This adds **bin\stcewfile.exe** file as the executable file, making the component a File e*Way.

7 Under **Configuration file**, click **New**.

The **Edit Settings** dialog box appears as shown below.
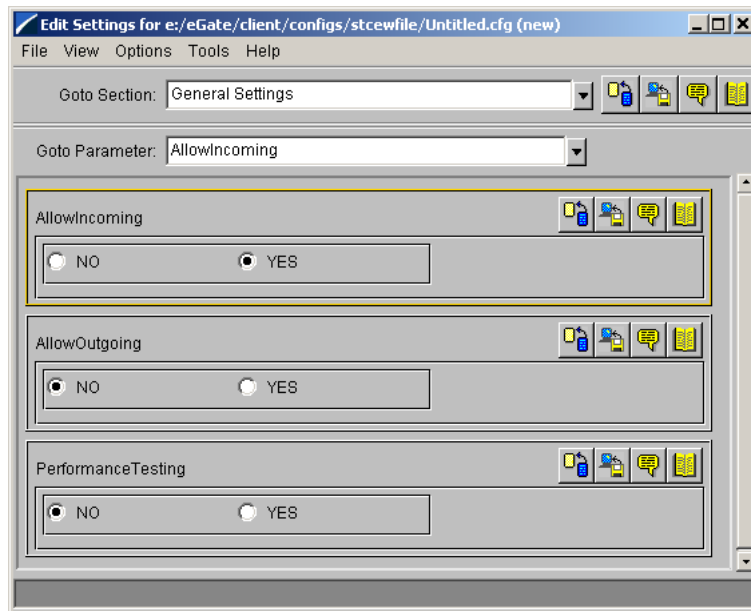
8 In the **Goto Section** list, click **Poller (inbound) settings**.

9 In the **PollDirectory** box, type *drive:\eGatefolder***\Client\Data** and press ENTER.

where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client\Data.

This adds the **Data** directory as the directory to be polled as shown in the screen below.
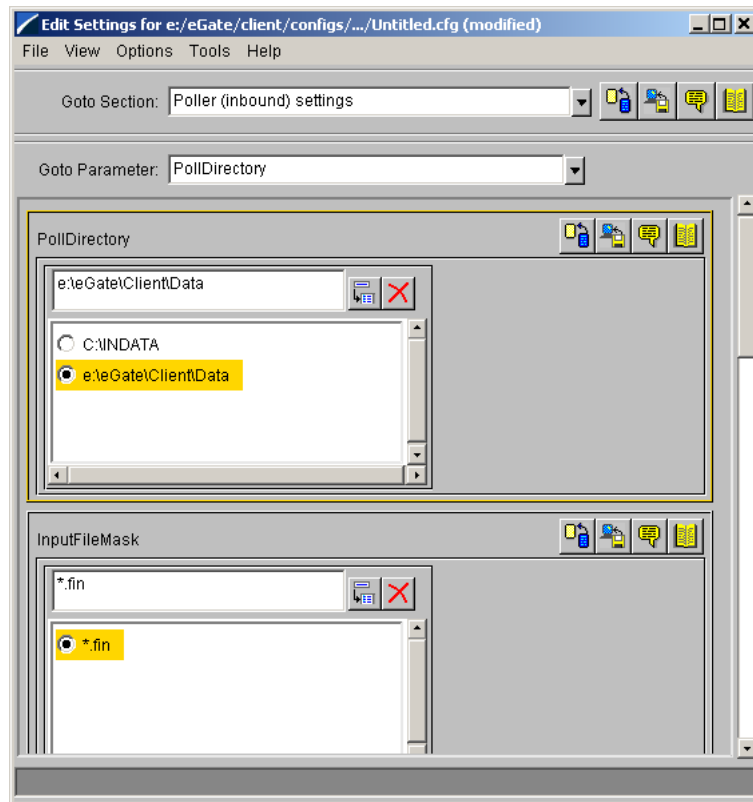
10 On the **File** menu, click **Save**.

11 In the **Save As** dialog box, click **Save** to accept the default file name **ew_Rec.cfg** and save the file.

12 On the **File** menu, click **Close** to close the Edit Settings dialog box.

This adds **configs\stcewfile\ew_Rec.cfg** under **Configuration file** in the **e*Way - ew_Rec Properties** dialog box.

13 Click the **Start Up** tab and click **Start automatically**.

14 Click **OK** to close the **e*Way - ew_Rec Properties** dialog box.

4.6.2 Adding the ew_Pur File e*Way

To add the ew_Pur File e*Way

1 Use steps 1 through 7 from **"Adding the ew_Rec File e*Way" on page 57** to add another File e*Way named **ew_Pur**.

The **Edit Settings** dialog box appears.

2 Under **General Settings**, click **NO** under **AllowIncoming**, and **YES** under **AllowOutgoing**.

3 In the **Goto Section** list, click **Outbound (send) settings**.

4 In the **OutputDirectory** box, type *drive:\eGatefolder***Client\Data** and press ENTER.

where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client\Data.

5 In the **OutputFileName** box, type **ELSE2Eoutput%d.dat** and press ENTER.
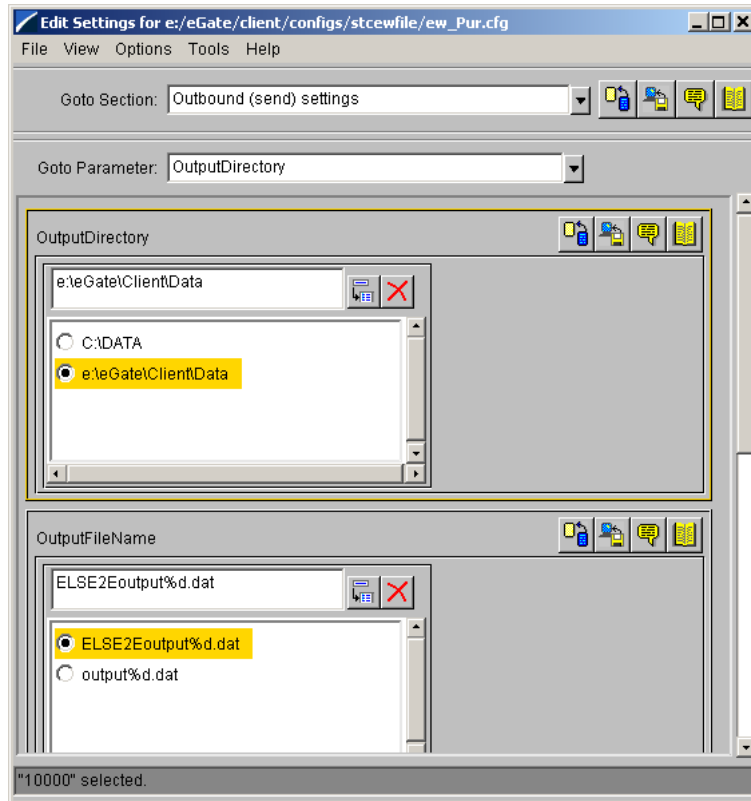
6 On the **File** menu, click **Save**.

7 In the **Save As** dialog box, click **Save** to accept the default file name **ew_Pur.cfg** and save the file.

8 On the **File** menu, click **Close** to close the **Edit Settings** dialog box.

9 Click the **Start Up** tab and click **Start automatically**.

10 Click **OK** to close the **e*Way - ew_Pur Properties** dialog box.

4.6.3 Adding the ew_ELS Multi-Mode e*Way

The procedure below describes how you add the ew_ELS Multi-Mode e*Way. You configure this e*Way with a remote debugging port number. For remote debugging to work, you must enable in -schema debugging on the Participating Host. For details, refer to the *e*Gate Integrator Installation Guide*.

To add the ew_ELS Multi-Mode e*Way

1 In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder, and then click the Control Broker (*hostname_cb*).

2 Click **Create a New e*Way** 🔧.

3 In the **New e*Way Component** box, type **ew_ELS** and click **OK**.

This adds the **ew_ELS** e*Way to the Control Broker pane.

4 In the **Components** tab, double-click the **ew_ELS** e*Way.

The **ew_ELS - Properties** dialog box appears.

5 Under **Configuration file**, click **New**.

The **Edit Settings** dialog box appears.

6 In the **Goto parameters** list, click **Remote debugging port number**.

7 In the **Remote debugging port number** box, type **8000** and press ENTER.
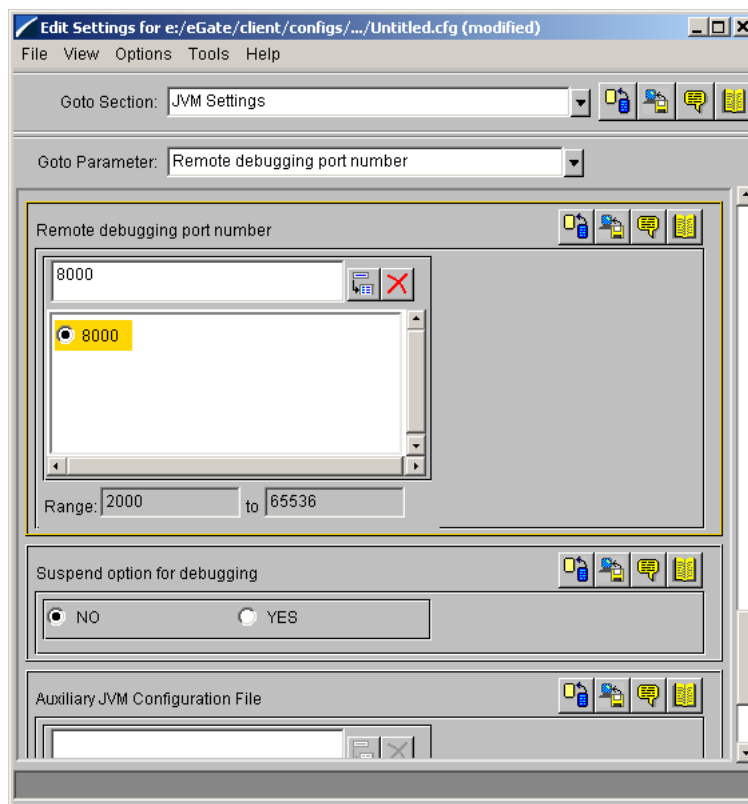
8 On the **File** menu, click **Save**.

9 In the **Save As** dialog box, click **Save** to accept the default file name **ew_ELS.cfg** and save the file.

10 On the **File** menu, click **Close** to close the **Edit Settings** dialog box.

11 Click the **Start Up** tab and click **Start automatically**.

12 Click **OK** to close the **e*Way - ew_ELS Properties** dialog box.

4.6.4 **Configuring the IQ Manager**

To configure the IQ Manager

1 In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder, then double-click the IQ manager (*hostname*_**iqmgr**).

The **IQ Manager - Properties** dialog box appears.

2 Click the **Start Up** tab and click **Start automatically**.

3 Click **OK** to close the **IQ Manager - Properties** dialog box.

4.6.5 **Adding the JMS e*Way Connection**

To add the JMS e*Way Connection

1 In the **Components** tab of the Enterprise Manager, click the **e*Way Connections** folder.

2 Click **Create a New e*Way Connection** [icon].

3 In the **Name** box, type **JMS** and click **OK**.

4 Double-click **JMS**.

The **e*Way Connection - JMS Properties** dialog box displays.
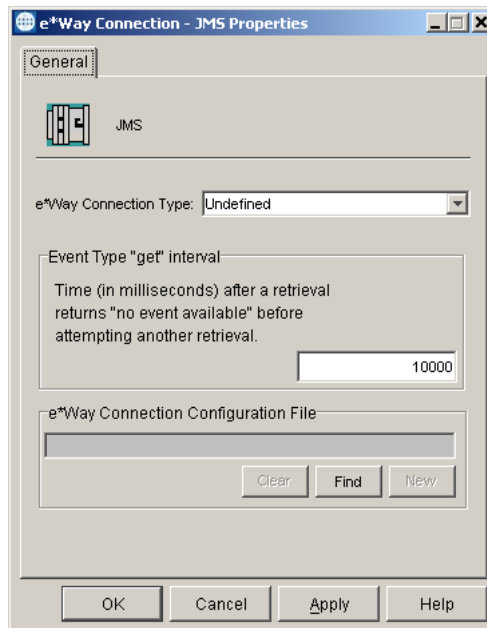
5 In the **e*Way Connection Type** list, click **SeeBeyond JMS**.

6 Click **New**.

7 In the **JMS IQ Manager** box, click your IQ Manager (*hostname*_**iqmgr**).

The screen below shows an example

8 Click **OK** to close the **New JMS e*Way Connection** dialog box.

9 Click **OK** to close the **e*Way Connection - JMS Properties** dialog box.

4.7 Adding the Collaborations

e*Ways use Collaborations to route the data through e*Gate. Typically, you configure the Collaborations in upstream to downstream order: the **col_Rec** Collaboration must be configured before configuring either the **col_ELS** or **col_Pur** Collaborations.

The figure below shows the Collaborations to the remainder of the parts that make up the complete schema.
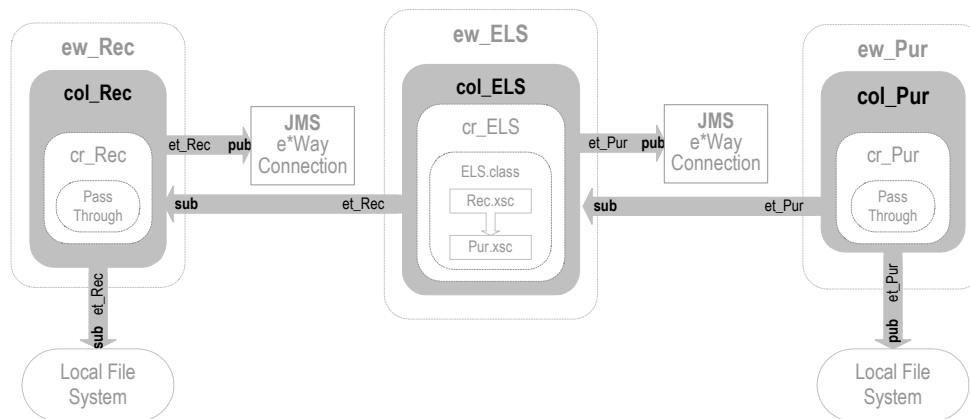
4.7.1 Adding the col_Rec Collaboration

The procedure below describes how you add the **col_Rec** Collaboration. The **col_Rec** Collaboration brings the data into e*Gate.

To add the col_Rec Collaboration

1 In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder and the Control Broker (*hostname_cb*).

2 Click the **ew_Rec** e*Way.

3 Click **Create a New Collaboration** .

4 In the **New Collaboration Component** dialog box, type **col_Rec** and click **OK**.

5 Double-click **col_Rec**.

The **Collaboration - col_Rec Properties** dialog box displays.

6 In the **Collaboration Rules** list, click **cr_Rec**.

7 Under **Subscriptions**, click **Add**.

This adds a row to the **Subscriptions** area.

8 Double-click the new row in the **Instance Name** column and click **JavaPassThroughIn**.

9 Double-click the row in the **Event Type** column and click **et_Rec**.

10 Double-click the row in the **Source** column and click **<EXTERNAL>**.

11 If a message appears regarding journal expiration time, click **No**.

12 Under **Publications**, click **Add**.

13 Double-click the row in the **Event Type** column and click **et_Rec**.

14 Double-click the row in the **Destination** column and click **JMS**.

The **Collaboration - col_Rec Properties** dialog box now displays as follows:
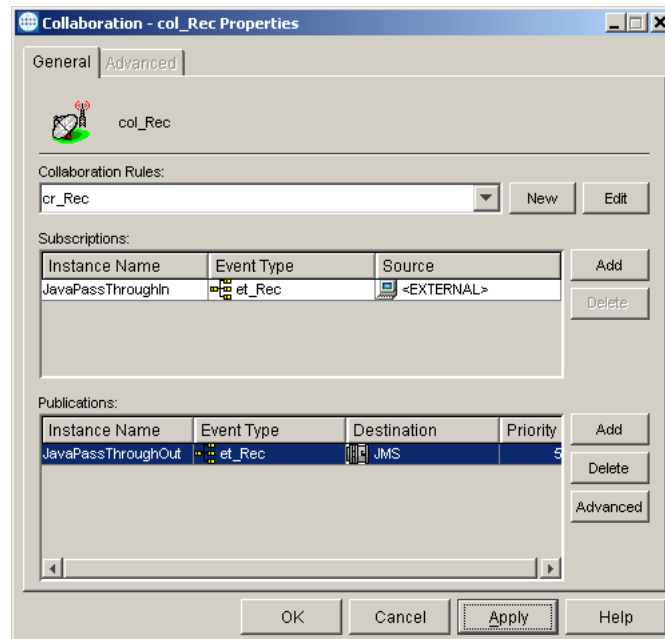
15 Click **OK** to close the **Collaboration - col_Rec Properties** dialog box.

4.7.2 Adding the col_ELS e*Way

The procedure below describes how you add the **col_ELS** Collaboration. The **col_ELS** Collaboration changes the data from the **et_Rec** Event Type to the **et_Pur** Event Type.

To add the col_ELS Collaboration

1   In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder and the Control Broker (*hostname_cb*).

2   Click the **ew_ELS** e*Way.

3   Click **Create a New Collaboration** .

4   In the **New Collaboration Component** dialog box, type **col_ELS** and click **OK**.

5   Double-click **col_ELS**.

6   In the **Collaboration Rules** list, click **cr_ELS**.

7   Under **Subscriptions**, click **Add**.

8   Double-click the new row in the **Instance Name** column and click **RootIn**.

9   Double-click the **Event Type** column and click **et_Rec**.

10 Double-click the **Source** column and click **JMS**.

11 Under **Publications**, click **Add**.

12 Double-click the new row in the **Instance Name** column and click **RootOut**.

13 Double-click the **Event Type** column and click **et_Pur**.

14 Double-click the **Destination** column and click **JMS**.

The **Collaboration - col_ELS Properties** dialog box now displays as follows:

15 Click **OK** to close the **Collaboration - col_ELS Properties** dialog box.

4.7.3 Adding the col_Pur Collaboration

The procedure below describes how you add the **col_Pur** Collaboration. The **col_Pur** Collaboration sends the transformed data out of e*Gate.

To add the col_Pur Collaboration

1   In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder and the Control Broker (*hostname_cb*).

2   Click the **ew_Pur** e*Way.

3   Click **Create a New Collaboration** .

4   In the **New Collaboration Component** dialog box, type **col_Pur** and click **OK**.

5   Double-click **col_Pur**.

6   In the **Collaboration Rules** list, click **cr_Pur**.

7   Under **Subscriptions**, click **Add**.

8   Double-click the new row in the **Instance Name** column and click **JavaPassThroughIn**.

9   Double-click the **Event Type** column and click **et_Pur**.

10   Double-click the **Source** column and click **JMS**.

11   Under **Publications**, click **Add**.

12   Double-click the **Event Type** column and click **et_Pur**

13   Double-click the **Destination** column and click **<EXTERNAL>**.

The **Collaboration - col_Pur Properties** dialog box now displays as follows:

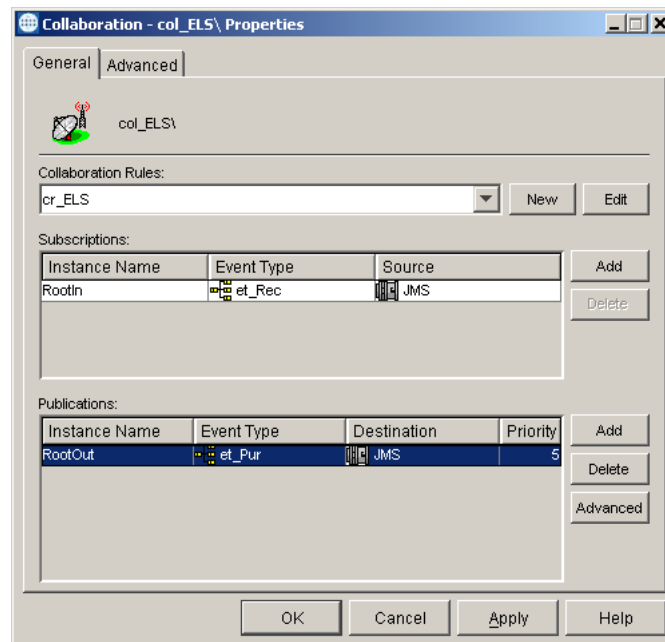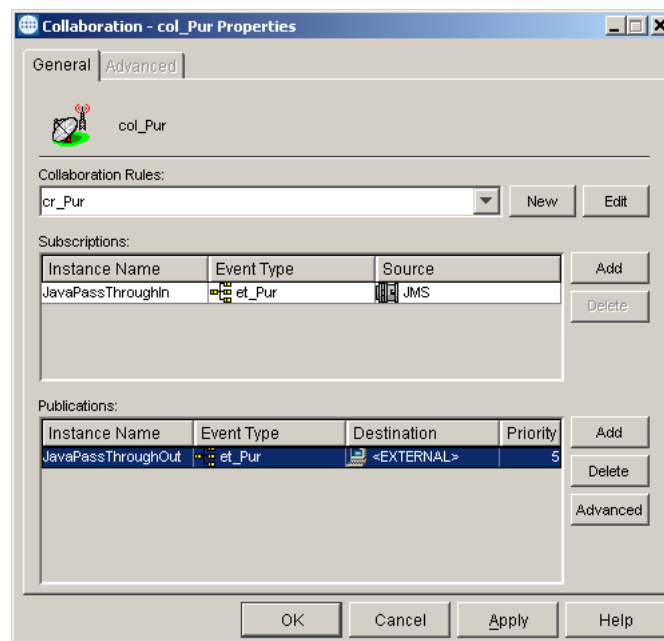14   Click **OK** to close the **Collaboration - col_Pur Properties** dialog box.

4.8   Reviewing the Complete Schema

Now that you have completed the schema, use the table below to verify that you have created all the components and that all settings for the components are correct.

Component	Logical Name	Settings
Schema	ELSE2E	
Control Broker	*hostname*_cb	
IQ Manager	*hostname*_iqmgr	Service = SeeBeyond JMS Config file = *hostname*_iqmgr.cfg Start Up = Auto
Event Type	et_Rec	et_Rec.xsc
	et_Pur	et_Pur.xsc
Java ETD	Rec.xsc	Package Name = RecPackage
	Pur.xsc	Package Name = PurPackage
Collaboration Rules	cr_Rec	Service = Java JavaPassThroughIn GenericInEvent.ssc Trigger JavaPassThroughOut GenericOutEvent.ssc
	cr_ELS	Service = Java RootIn Rec.xsc In Trigger RootOut Pur.xsc Out
	cr_Pur	Service = Java JavaPassThroughIn GenericInEvent.ssc Trigger JavaPassThroughOut GenericOutEvent.ssc
Java Collaboration Rules Class	ELS.class	Source = RootIn (Rec) Destination = RootOut (Pur)
Inbound File e*Way	ew_Rec	Executable = stcewfile.exe Config file = ew_Rec.cfg Start Up = Auto Collaboration = col_Rec
Outbound File e*Way	ew_Pur	Executable = stcewfile.exe Config file = ew_Pur.cfg Start Up = Auto Collaboration = col_Pur
Multi-Mode e*Way	ew_ELS	Executable = stceway.exe Config file = ew_ELS.cfg Start Up = Auto Collaboration = col_ELS
JMS e*Way Connection	JMS	Type: SeeBeyond JMS Config file = jms.cfg

Component	Logical Name	Settings
Collaborations	col_Rec	Collaboration Rules = cr_Rec Sub = et_Rec from <EXTERNAL> Pub = et_Rec to JMS
	col_ELS	Collaboration Rules = cr_ELS Sub = et_Rec from JMS Pub = et_Pur to JMS
	col_Pur	Collaboration Rules = cr_Pur Sub = et_Pur from JMS Pub = et_Pur to <EXTERNAL>

4.9 Creating the Input Data File

Follow the procedure below to create the sample input data file. The schema uses this file to bring data into e*Gate.

To create the input data file

1 Use a text editor such as Notepad to create a text file with the following information:

```
Order01,this is message 1 for Order01,3,
Order01,this is message 2 for Order01,3,
Order02,this is message 1 for Order02,2,
Order03,this is message 1 for Order03,4,
Order03,this is message 2 for Order03,4,
Order01,this is message 3 for Order01,3,
Order02,this is message 2 for Order02,2,
Order03,this is message 3 for Order03,4,
Order03,this is message 4 for Order03,4,
```

2 In Windows Explorer, open the *drive*:*eGatefolder***Client** folder, where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client.

3 In the **Client** folder, create a folder named **Data**.

4 Save the file as **ELSE2Einput.txt**.

4.10 Starting the Schema

To start the schema

1 Use the following command to start the Control Broker from a command line.

```
stccb.exe -rh hostname -rs ELSE2E -ln hostname_cb -un username
-up password
```

2 Start the e*Gate Monitor and open the **ELSE2E** schema.

1 Expand the **Schema - ELSE2E** folder, the **Participating Hosts** folder, and then click the Control Broker (*hostname_cb*).

2 Verify that all the components are running by checking that the Status Name for each component lists **Up** as shown in the screen below.
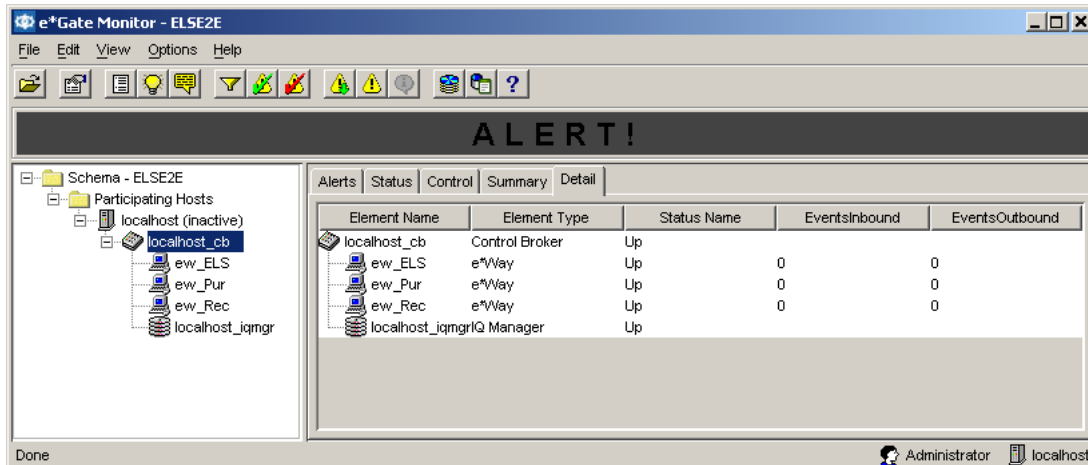
4.11 Testing the Schema

To test the schema

1 In Windows Explorer, open the *drive*:***eGatefolder***\\Client\\Data** folder, where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client\Data.

2 Change the file extension of **ELSE2Einput.txt** to **.fin**.

3 Click **Yes** when the confirmation message appears.

4 Verify that the extension changes to **.~in** indicating that the **ew_Rec** e*Way retrieved the file.

Almost immediately, the output file **ELSE2Eoutput#.dat** appears in the folder, indicating the schema ran successfully.

5 Verify that the output looks like that shown in the screen below.
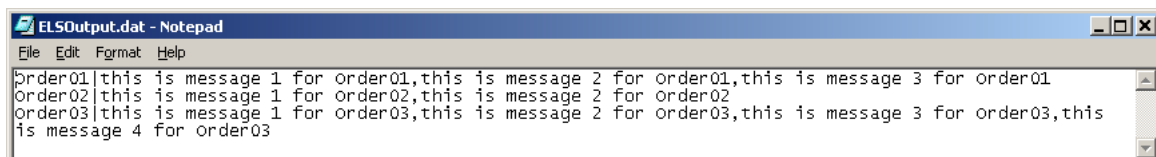
4.12 Troubleshooting

If you encounter a problem and are unable to solve it with **"Reviewing the Complete Schema" on page 67**, refer to **"Debugging and Log Files" on page 92** to start the trace or error log component in your schema.

Error	Possible Solution
unreachable statement	In the **executeBusinessRules** method, place the **return retBoolean** rule below the user-defined rules.

Creating a Monk End-to-End Scenario

This chapter provides step-by-step procedures for creating and testing a Monk end-to-end scenario that handles data transformation.

In This Chapter

5.1 Presenting the Business Challenge

The business needs a way to determine if incoming data is valid or not and route the data accordingly. Valid data is sent to one system while invalid data is sent to another.

5.2 Presenting the e*Gate Solution

The e*Gate solution uses a Monk Collaboration in the inbound e*Way to separate and route the data. The figure below shows the component relationships of e*Gate that implements this business logic.

The figure below shows the complete e*Gate schema.
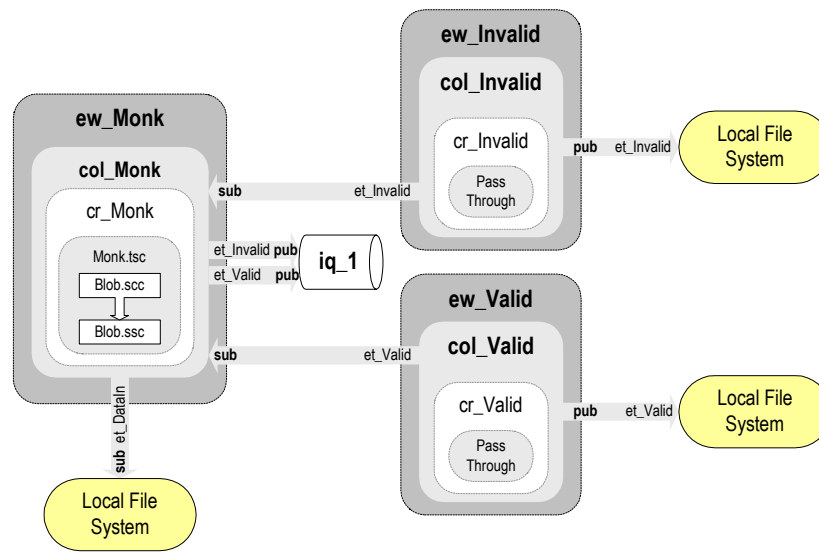
In this figure, you see the following process:

1 The **ew_Monk** e*Way brings the data into e*Gate and separates it into valid and invalid types.

The **col_Monk** Collaboration in the **ew_Monk** e*Way subscribes to a location on the local file system containing data to process. The data is packaged as Event Type (**et_DataIn**) and processed by the Collaboration Rules Script (**Monk.tsc**). The Monk script determines whether the data is valid or invalid. Invalid data is repackaged as Event Type **et_Invalid**. Valid data is repackaged as Event Type **et_Valid**. The **col_Monk** Collaboration published both Event Types to **iq_1**.

2 The **ew_Invalid** e*Way retrieves the invalid data and publishes it to a location on the local file system.

The **col_Invalid** in the **ew_Invalid** subscribes to **et_Invalid** Events published by **col_Monk**. The **cr_Invalid** Collaboration Rule uses the Pass Through service to move the data without modifying it. When an **et_Invalid** Event is retrieved, the **ew_Invalid** file e*Way packages it as a text file and writes it to the specified location on the local file system.

3 The **ew_Valid** e*Way retrieves the valid data and publishes it to a location on the local file system.

The **col_Valid** in the **ew_Valid** subscribes to **et_Valid** Events published by **col_Monk**. When an **et_Valid** Event is retrieved, the **ew_Valid** file e*Way packages it as a text file and writes it to the specified location on the local file system.

5.3 Creating the Schema

Before you start creating the schema, make sure that you have the following system requirements:

- e*Gate Registry Host, Participating Host, and GUIs installed on a single system

- The system is running the Windows operating system

For more information about the installation requirements, refer to **"e*Gate Installation Requirements" on page 11.**

To create the schema

  1 Start the e*Gate Enterprise Manager and log in as **Administrator** (or another user with administrator privileges) to the appropriate Registry Host.

  2 In the **Open Schema on Registry Host** dialog box, click **New**.

  3 In the **Enter New Schema Name** box, type **MonkE2E**, and click **Open**.

     The Enterprise Manager displays the new **MonkE2E** schema.

  4 Click the **Components** tab. You perform all configuration steps in the **Components** tab.
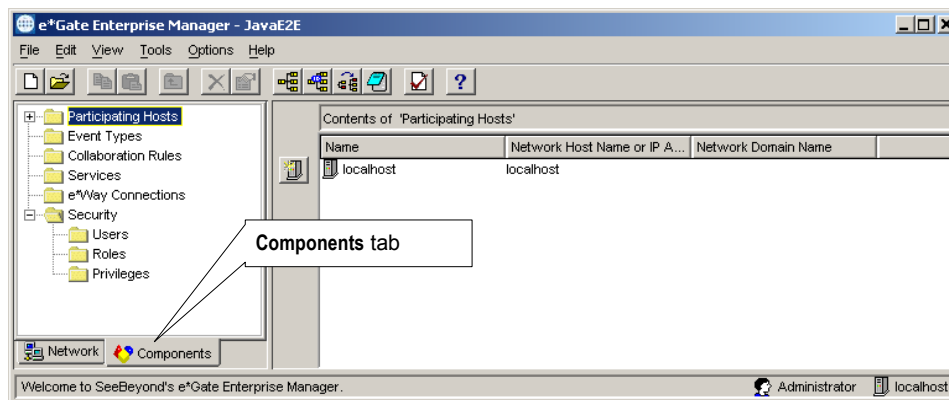
5.4   Creating the Monk ETD and the Event Types

The ETD is created first using the Monk ETD Editor and then the three Event Types are created in the e*Gate Enterprise Manager.

This scenario uses three Event Types that share one common Monk ETD called **Blob.ssc**. The Event Types are:

- **et_DataIn**

- **et_Invalid**

- **et_Valid**

Event Types group together similar Events. In this scenario, the similarity between the Events named by a particular Event Types is their routing. For example, all the Events routed to the **ew_Valid** e*Way use the Event Type **et_Valid**.

The sections below describe how you create the Monk ETD and the Event Types for the MonkE2E schema.

The figure below shows where these parts fit into the schema.
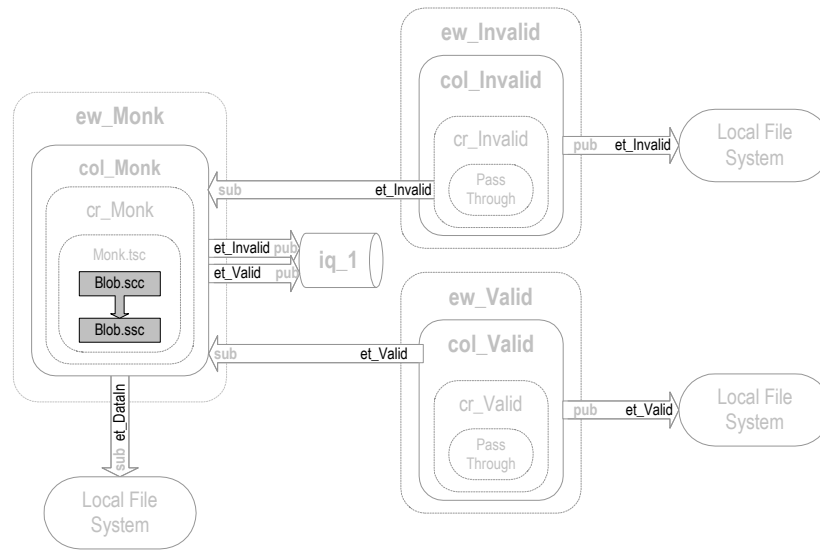
5.4.1 Creating the Blob.ssc ETD

The procedure below describes how you create the Blob.ssc ETD. Before you start, you must have already created the MonkE2E schema as described in **"Creating the Schema" on page 72**.

ETD structures can be complex, with many nodes and levels depending on the data type modeled. For simplicity and brevity, the data in this scenario has an uncomplicated structure with one node.

To create the Blob.xsc ETD

1 In the e*Gate Enterprise Manager, click **Default Editor** on the **Options** menu.

2 Click **Monk** if necessary and click **OK**.

3 On the toolbar, click **ETD Editor** .

4 In the ETD Editor, click **New** on the toolbar.

5 In the **File name** box, type **Blob.ssc**. and click **OK**.

6 Click **Node** on the toolbar.

This adds a single root node named Blob to the ETD work space as shown below.
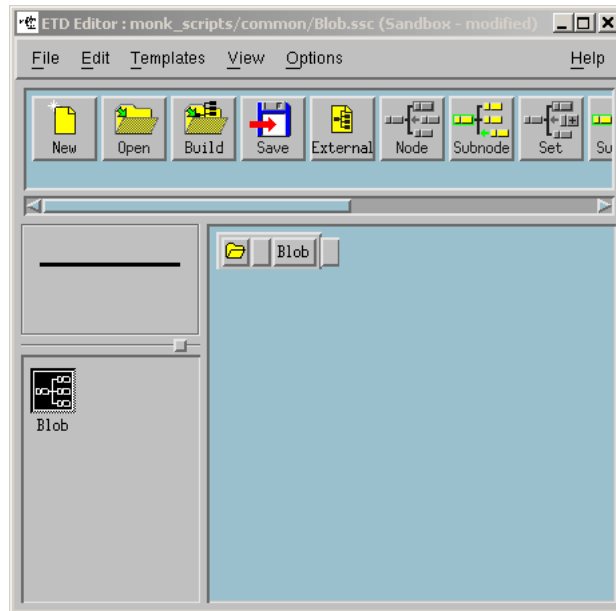
7 On the **File** menu, click **Save**.

8 On the **File** menu, click **Close** to close the ETD Editor.

5.4.2 Creating the Event Types

The procedure below describes how you create the MonkE2E schema Event Types. The et_DataIn Event Type brings the data into e*Gate. The **et_Valid** Event Type routes data to the **ew_Valid** e*Way. The **et_Invalid** Event Type routes data to the **ew_Invalid** e*Way.

To create the Event Types

1 In the **Components** tab of the Enterprise Manager, click the **Event Types** folder.

2 Click **Create a New Event Type** 🖼️.

3 In the **Name** box, type **et_DataIn** and click **OK**.

This adds **et_DataIn** to the Event Types pane.

4 In the Event Types pane, double-click **et_DataIn**.

The **Event Type - et_DataIn Properties** dialog box appears.

5 Click **Find**.

6 Double-click the **common** folder if necessary and double-click **Blob.ssc**.

7 Click **OK** to close the **Event Type - et_DataIn Properties** dialog box.

8 Repeat steps 1 through 7 to create the **et_Valid** and **et_Invalid** Event Types. Substitute **et_Valid** and **et_Invalid** for the Event Type name.

The Event Types pane now shows the newly created Event Types.
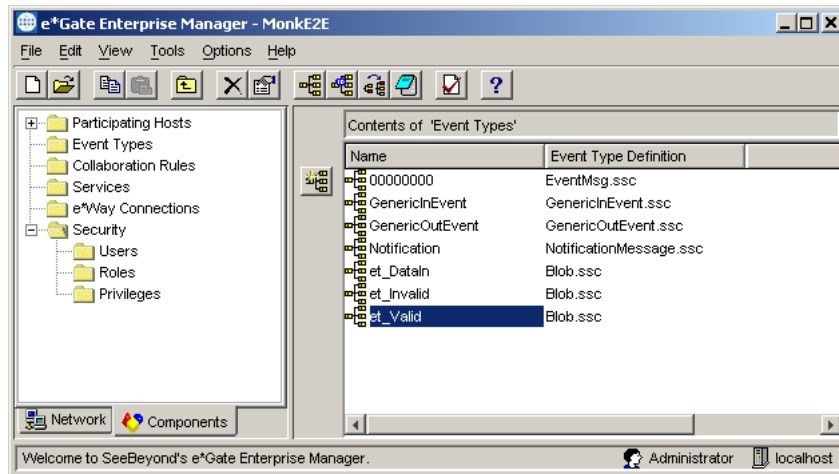
5.5 Creating the Collaboration Rules

Collaboration Rules are used by the Collaborations that route the data. Collaborations that route data without change use Pass Through Collaboration Rules. Data that must be transformed in addition to being routed must use Collaboration Rules. These rules in turn use a Collaboration Service (programming environment) that allows data transformation.

e*Gate has several Collaboration Services such as Java, C, Monk, and others. The choice of a Collaboration Service depends on what you want to do with the data, as well as your available programming resources.

The Monk end-to-end scenario uses three Collaboration Rules:

- a Monk Collaboration Rules Component, **cr_Monk**, which sorts the data into valid and invalid Event Types

- two Pass Through Collaborations Rules, **cr_Invalid** and **cr_Valid**

The figure below shows where the Collaboration Rules fit into the schema.
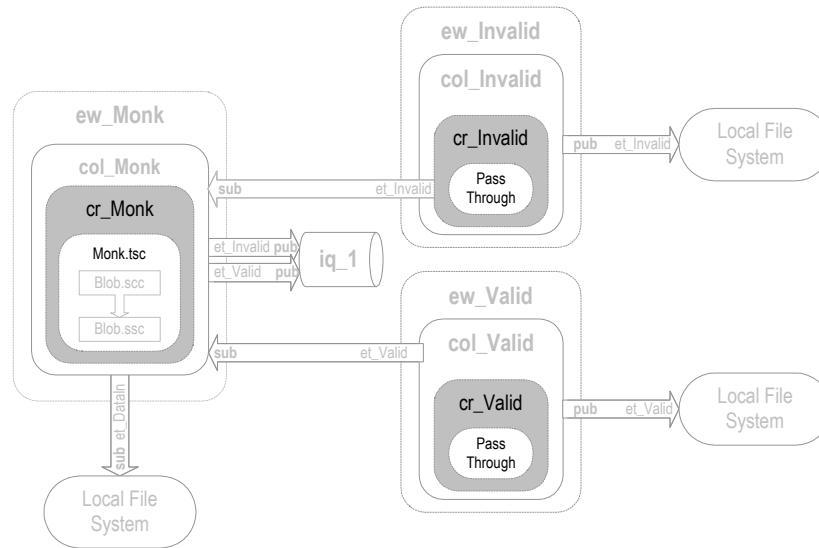
5.5.1 Creating the Pass Through Collaboration Rules

The procedure below describes how to create the **cr_Invalid** and **cr_Valid** Pass Through Collaboration Rules for the Monk scenario.

To create the Pass Through Collaboration Rules

1   In the **Components** tab of the e*Gate Enterprise Manager, click the **Collaboration Rules** folder.

2   Click **Create New Collaboration Rules** .

3   In the **Name** box, type **cr_Invalid** and click **OK**.

    cr_Invalid displays in the Collaboration Rules pane.

4   Double-click **cr_Invalid** in the Collaboration Rules pane.

    The **Collaboration Rules - cr_Invalid Properties** dialog box appears.

5   Click the **Subscriptions** tab.

6 Under **Available Input Event Types**, click **et_Invalid**, and click the right arrow as follows.
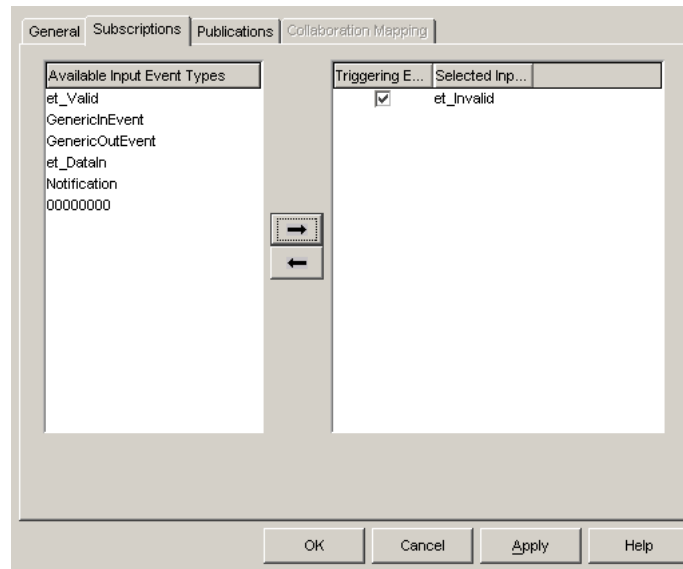
7 Click the **Publications** tab.

8 Under **Available Output Event Types**, click **et_Invalid**, and click the right arrow.

9 Click **OK** to close the **Collaboration Rules - cr_Invalid Properties** dialog box.

10 Repeat steps 2 through 9 to create the **cr_Valid** Collaboration Rules component. Substitute **cr_Valid** for the Collaboration Rule name and substitute **et_Valid** for the Event Type.

5.5.2 Creating the cr_Monk Collaboration Rule Component

This section describes how you create the Collaboration Rules component **cr_Monk**. Creating **cr_Monk** consists of the following steps:

1 Adding the component.

2 Specifying the Collaboration Rules script.

Adding the cr_Monk Collaboration Rule Component

To add the cr_Monk Collaboration Rule component

1 In the **Components** tab of the e*Gate Enterprise Manager, click the **Collaboration Rules** folder.

2 Click **Create New Collaboration Rules** .

3 In the **Name** box, type **cr_Monk** and click **OK**.

4 Double-click **cr_Monk** in the Collaboration Rules pane.

5 In the **Service** list in the **General** tab, click **Monk**.

6 Click the **Subscriptions** tab.

7   Under **Available Input Event Types**, click **et_DataIn**, and click the right arrow.

8   Click the **Publications** tab.

9   Under **Available Output Event Types**, click **et_Invalid**, and click the right arrow.

10   Under **Available Event Types**, click **et_Valid**, and click the right arrow.

11   Click the **General** tab.

12   Under **Collaboration Rules**, click **New**.

13   In the **File name** box in the **New** dialog box, type **Monk.tsc**.

14   Verify that the other settings are **SeeBeyond Collaboration Rules (.tsc)** and **Blob.ssc** as the source ETD and destination ETD, and click **OK**.

The **Collaboration Rules Editor** window displays **Monk.tsc**. You can now add the **cr_Monk** Collaboration Rules script as described in the next section.

Specifying the cr_Monk Collaboration Rules Script

The procedure below describes how you specify the **cr_Monk** script to implement the business logic that validates the data. The data is valid if it is a number and invalid if it is not a number. You use the Monk scripting language to create the **cr_Monk** script.

To specify the cr_Monk Collaboration Rules script

1   In the **Collaboration Rules Editor** window, click **If** on the toolbar.

An **If...Else** rule appears in the Rules pane.

2   Replace **<test>** in the condition portion of the **IF** rule with the following.

```
(number? (get ~input%Blob))
```

This tests whether the data in the input Event is a number.

3   Click the blue line below the **IF** line.

4   On the **Rules** menu, click **Add Function**.

5   In the **FUNCTION** line, replace **<Functions>** with the following.

```
(iq-put "et_Valid" (get ~input%Blob) (list "et_DataIn") 0 0 0)
```

This retrieves the input data and repackages it as an **et_Valid** Event Type and publishes it to an IQ using **iq-put**.

6   Click the blue line below the **ELSE** line.

7   On the **Rules** menu, click **Add Function**.

8   Replace **<Functions>** with the following.

```
(iq-put "et_Invalid" (get ~input%Blob) (list "et_DataIn") 0 0 0)
```

This retrieves the input data and repackages it as an **et_Invalid** Event Type and publishes it to an IQ using **iq-put**. The Rules pane shows the script as follows:

```
▼        IF          (number? (get ~input%Blob))
              FUNCTION        (iq-put "et_Valid" (get ~input%Blob) (list "et_DataIn") 0 0 0)
         ELSE
              FUNCTION        (iq-put "et_Invalid" (get ~input%Blob) (list "et_DataIn") 0 0 0)
```

9 Click **Validate** ✔ on the toolbar.

10 Correct any errors, then click **OK.**

11 Click **Save** 💾 on the toolbar.

12 On the **File** menu, click **Close** the **Collaboration Rules Editor** window.

The **Collaboration Rules - cr_Monk Properties** dialog box displays **monk_scripts\common\Monk.tsc** under **Collaboration Rules**.

13 Click **OK** to close the **Collaboration Rules - cr_Monk Properties** dialog box.

5.6 Adding the e*Ways and IQ

After creating the ETDs and Collaborations, you add the e*Ways and the IQ Manager. This section describes how you add those components.

As shown in the figure below, the MonkE2E schema has the following e*Way and IQ components:

- **ew_Monk** File e*Way
- **ew_Invalid** File e*Way
- **ew_Valid** e*Way
- **iq_1** IQ Manager
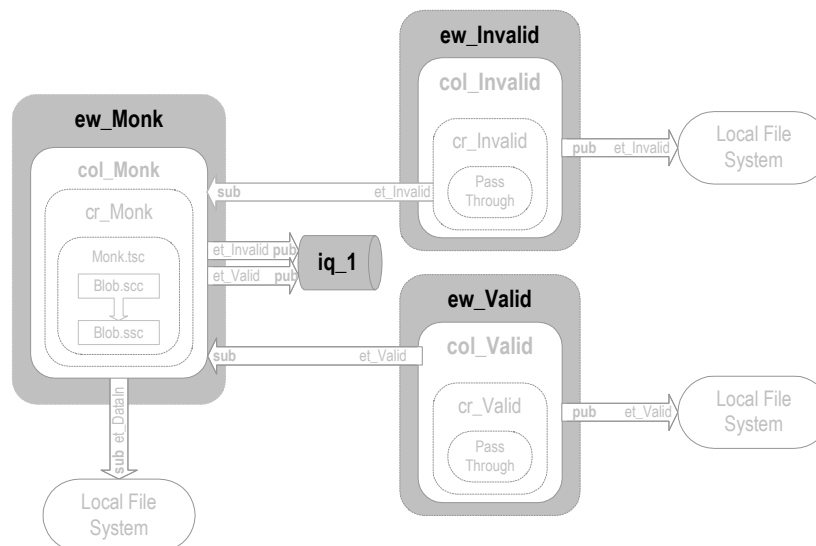
All e*Ways in the MonkE2E schema are File e*Ways. File e*Ways connect to external file systems to read and write files.

5.6.1 Adding the ew_Monk File e*Way

To add the ew_Monk File e*Way

1 In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder, and then click the Control Broker (*hostname*_**cb**).

2 Click **Create a New e*Way** .

3 In the **New e*Way Component** box, type **ew_Monk** and click **OK**.

   This adds the **ew_Monk** e*Way to the Control Broker pane.

4 In the **Components** tab, double-click the **ew_Monk** e*Way.

   The **ew_Monk - Properties** dialog box appears.

5 Under **Executable file**, click **Find**.

6 In the **File Selection** dialog box, double-click **stcewfile.exe**.

   This adds **bin\stcewfile.exe** file as the executable file, making the component a File e*Way.

7 Under **Configuration file**, click **New**.

   The **Edit Settings** dialog box appears as shown below.
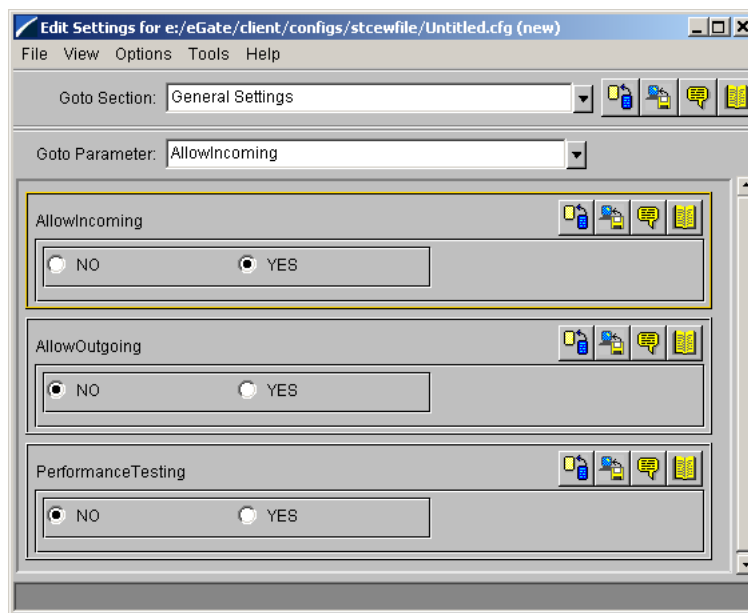
8 In the **Goto Section** list, click **Poller (inbound) settings**.

9 In the **PollDirectory** box, type *drive:\eGatefolder***Client\Data\MonkE2E** and press ENTER.

   where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client\Data\MonkE2E.

This adds the **Data** directory as the directory to be polled as shown in the screen below.

10 On the **File** menu, click **Save**.

11 In the **Save As** dialog box, click **Save** to accept the default file name **ew_Monk.cfg** and save the file.

12 On the **File** menu, click **Close** to close the Edit Settings dialog box.

This adds **configs\stcewfile\ew_Monk.cfg** under **Configuration file** in the **e*Way - ew_Monk Properties** dialog box.

13 Click the **Start Up** tab and click **Start automatically**.

14 Click **OK** to close the **e*Way - ew_Monk Properties** dialog box.

5.6.2 Adding the ew_Invalid e*Way

To add the ew_Invalid e*Way

1 Use steps 1 through 7 from **"Adding the ew_Monk File e*Way" on page 81** to add another File e*Way named **ew_Invalid**.

The **Edit Settings** dialog box appears.

2 Under **General Settings**, click **NO** under **AllowIncoming**, and **YES** under **AllowOutgoing**.

3 In the **Goto Section** list, click **Outbound (send) settings**.

4 In the **OutputDirectory** box, type *drive:\eGatefolder***Client\Data\MonkE2E** and press ENTER.

where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client\Data\MonkE2E.

5 In the **OutputFileName** box, type **Invalid_output%d.dat** and press ENTER.
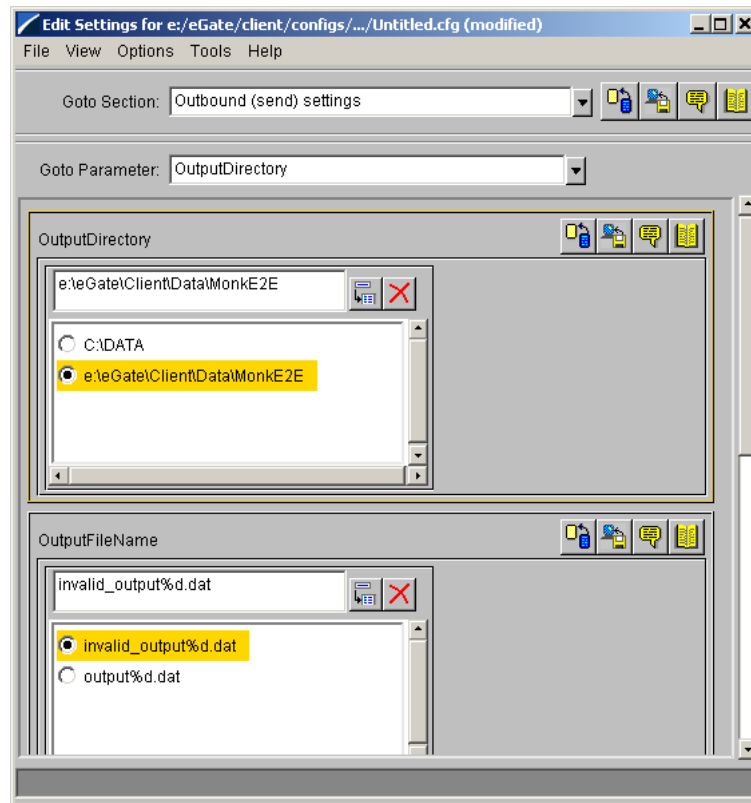

6 On the **File** menu, click **Save**.

7 In the **Save As** dialog box, click **Save** to accept the default file name (**ew_Invalid.cfg**) and save the file.

8 On the **File** menu, click **Close** to close the **Edit Settings** dialog box.

9 Click the **Start Up** tab, and then select the **Start automatically** check box.

10 Click **OK** to close the **e*Way - ew_Invalid Properties** dialog box.

5.6.3 Adding the ew_Valid e*Way

To add the ew_Valid e*Way

1 Use steps 1 through 7 from **"Adding the ew_Monk File e*Way" on page 81** to add another File e*Way named **ew_Valid**.

The **Edit Settings** dialog box appears.

2 Under **General Settings**, click **NO** under **AllowIncoming**, and **YES** under **AllowOutgoing**.

3   In the **Goto Section** list, click **Outbound (send) settings**.

4   In the **OutputDirectory** box, type *drive:\eGatefolder***Client\Data\MonkE2E** and press ENTER.

   where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client\Data\MonkE2E.

5   In the **OutputFileName** box, type **Valid_output%d.dat** and press ENTER.

6   On the **File** menu, click **Save**.

7   In the **Save As** dialog box, click **Save** to accept the default file name (**ew_Valid.cfg**) and save the file.

8   On the **File** menu, click **Close** to close the **Edit Settings** dialog box.

9   Click the **Start Up** tab, and then select the **Start automatically** check box.

10   Click **OK** to close the **e*Way - ew_Valid Properties** dialog box.

5.6.4 Configuring the IQ Manager

To configure the IQ Manager

1   In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder, then double-click the IQ manager (*hostname*_**iqmgr**).

   The **IQ Manager - Properties** dialog box appears.

2   In the **IQ Manager Type** list, click **SeeBeyond JMS**.

3   Click the **Start Up** tab and click **Start automatically**.

4   Click **OK** to close the **IQ Manager - Properties** dialog box.

5   In the e*Gate Enterprise Manager, click **Create a new IQ** 🖼 on the toolbar.

6   In the **Name** box, type **iq_1** for the IQ name and click **OK**.

5.7 Adding the Collaborations

The Collaborations route the data through e*Gate. Typically, you configure the Collaborations in upstream to downstream order: the **col_Monk** Collaboration must be configured before configuring either the **col_Invalid** or **col_Valid** Collaborations.

The figure below shows the Collaborations to the remainder of the parts that make up the complete schema.

5.7.1 Adding the col_Monk Collaboration

The procedure below describes how you add the **col_Monk** Collaboration. The **col_Monk** Collaboration brings the data into e*Gate and routes it depending on the whether the data is valid or invalid.
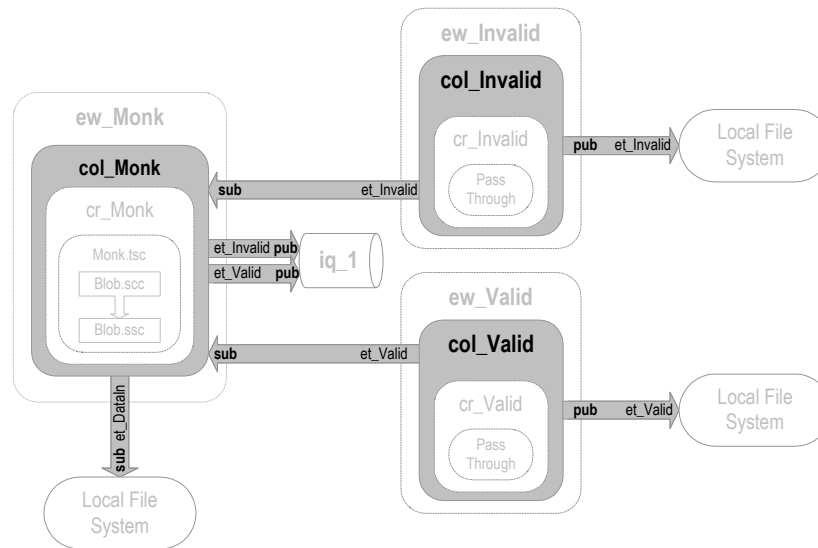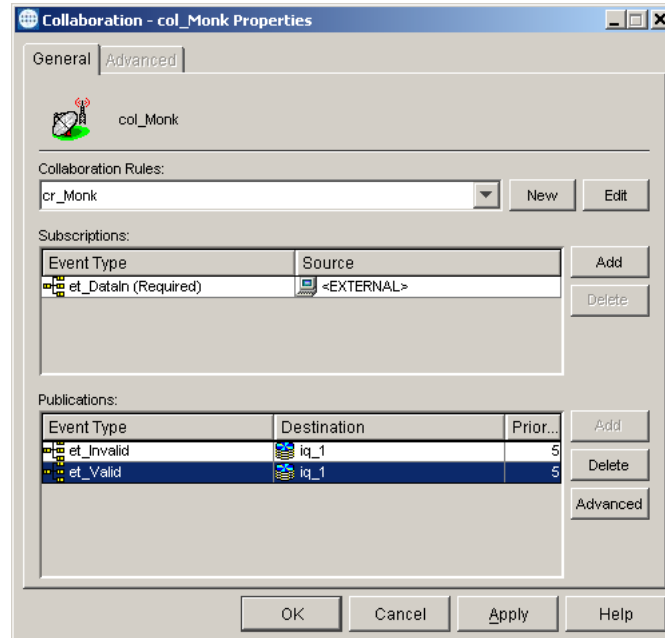
To add the col_Monk Collaboration

1   In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder and the Control Broker (*hostname_*cb).

2   Click the **ew_Monk** e*Way.

3   Click **Create a New Collaboration** .

4   In the **New Collaboration Component** dialog box, type **col_Monk** and click **OK**.

5   Double-click **col_Monk**.

    The **Collaboration - col_Monk Properties** dialog box displays.

6   In the **Collaboration Rules** list, click **cr_Monk**.

7   Under **Subscriptions**, click **Add**.

    This adds a row to the **Subscriptions** area.

8   Click the row in the **Event Type** column and click **et_DataIn**.

9   Click the row in the **Source** column and click **<EXTERNAL>**.

10 If a message appears regarding journal expiration time, click **No**.

11 Under **Publications**, click **Add**.

12 In the **Event Type** column, click **et_Invalid**.

13 In the **Destination** column, click **iq_1**.

14 Under **Publications**, click **Add**.

15 In the **Event Type** column, click **et_Valid**.

16 In the **Destination** column, click **iq_1**.

The **Collaboration - col_Monk Properties** dialog box now displays as follows:

17 Click **OK** to close the **Collaboration - col_Monk Properties** dialog box.

5.7.2 Adding the col_Invalid Collaboration

The procedure below describes the **col_Invalid** Collaboration. The **col_Invalid** Collaboration subscribes to **et_Invalid** Events published by **col_Monk**. It publishes them to **<EXTERNAL>**.

To add the col_Invalid Collaboration

1 In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder and the Control Broker (*hostname_cb*).

2 Click the **ew_Invalid** e*Way.

3 Click **Create a New Collaboration** .

4 In the **New Collaboration Component** dialog box, type **col_Invalid** and click **OK**.

5 Double-click **col_Invalid**.

6 In the **Collaboration Rules** list, click **cr_Invalid**.

7 Under **Subscriptions**, click **Add**.

8 Click the row in the **Event Type** column and click **et_Invalid**.

9 Click the row in the **Source** column and click **col_Monk**.

10 Under **Publications**, click **Add**.

    11 In the **Event Type** column, click **et_Invalid**.

    12 In the **Destination** column, click **<EXTERNAL>**.

    13 Click **OK** to close the **Collaboration - col_Invalid Properties** dialog box.

5.7.3 Adding the col_Valid Collaboration

The procedure below describes how you add the **col_Valid** Collaboration. The **col_Valid** Collaboration subscribes to **et_Valid** Events published by **col_Monk**. It publishes them to **<EXTERNAL>**.

To add the col_Valid Collaboration

    1 In the **Components** tab of the e*Gate Enterprise Manager, expand the **Participating Hosts** folder and the Control Broker (*hostname_***cb**).

    2 Click the **ew_Valid** e*Way.

    3 Click **Create a New Collaboration** .

    4 In the **New Collaboration Component** dialog box, type **col_Valid** and click **OK**.

    5 Double-click **col_Valid**.

    6 In the **Collaboration Rules** list, click **cr_Valid**.

    7 Under **Subscriptions**, click **Add**.

    8 Click the row in the **Event Type** column and click **et_Valid**.

    9 Click the row in the **Source** column and click **col_Monk**.

    10 Under **Publications**, click **Add**.

    11 In the **Event Type** column, click **et_Valid**.

    12 In the **Destination** column, click **<EXTERNAL>**.

    13 Click **OK** to close the **Collaboration - col_Valid Properties** dialog box.

5.8 Reviewing the Complete Schema

Now that you have completed the schema, use the table below to verify that you have created all the components and that all settings for the components are correct.

Component	Logical Name	Settings
Schema	MonkE2E	
Control Broker	*hostname_*cb	
IQ Manager	*hostname_*iqmgr	Start Up = Auto

Component	Logical Name	Settings
Event Type	et_DataIn	ETD = Blob.ssc
	et_Invalid	ETD = Blob.ssc
	et_Valid	ETD = Blob.ssc
ETD	Blob.ssc	
Collaboration Rule	cr_Monk	Service = Monk CRS = Monk.tsc Sub = et_DataIn Pub = et_Invalid, et_Valid
	cr_Invalid	Service = Pass Through Sub = et_Invalid Pub = et_Invalid
	cr_Valid	Service = Pass Through Sub = et_Valid from col_Monk Pub = et_Valid to <EXTERNAL>
Collaboration Rules script	Monk.tsc	Source = Blob.ssc Destination = Blob.ssc
Inbound e*Way	ew_Monk	Executable = stcewfile.exe Config file = ew_Monk.cfg Start Up = Auto Collaboration = col_Monk
Outbound e*Ways	ew_Invalid	Executable = stcewfile.exe Config file = ew_Invalid.cfg Start Up = Auto Collaboration = col_Invalid
	ew_Valid	Executable = stcewfile.exe Config file = ew_Valid.cfg Start Up = Auto Collaboration = col_Valid
IQ	iq_1	Service = STC_Standard
Collaboration	col_Monk	Collab Rule = cr_Monk Sub = et_DataIn from <EXTERNAL> Pub = et_Invalid, et_Valid to iq_1
	col_Invalid	Collab Rule = cr_Invalid Sub = et_Invalid from col_Monk Pub = et_Invalid to <EXTERNAL>
	col_Valid	Collab Rule = cr_Valid Sub = et_Valid from col_Monk Pub = et_Valid to <EXTERNAL>

5.9 Creating the Input Data File

Follow the procedure below to create the sample input data file. The schema uses this file to bring data into e*Gate.

To create the input data file

1 Use a text editor such as Notepad to create a text file with the following information:

```
1
2
3
error
```

The Collaboration Rule used by the **ew_Monk** e*Way tests the data for validity. The data contains both valid (numeric) and invalid data (characters) for the test.

2 In Windows Explorer, open the *drive*:*eGatefolder***Client** folder, where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\eGate\Client.

3 In the **Client** folder, create a folder named **Data**.

4 Open the data folder and create a folder named **MonkE2E**.

5 Save the file as **MonkE2Einput.txt**.

## 5.10 Starting the Schema

1 Use the following command to start the Control Broker from a command line.

```
stccb.exe -rh hostname -rs MonkE2E -ln hostname_cb -un username -
up password
```

2 Start the e*Gate Monitor and open the **MonkE2E** schema.

3 Expand the **Schema - MonkE2E** folder, the **Participating Hosts** folder, and then click the Control Broker (**hostname_cb**).

4 Verify that all the components are running by checking that the Status Name for each component lists **Up** as shown in the screen below.
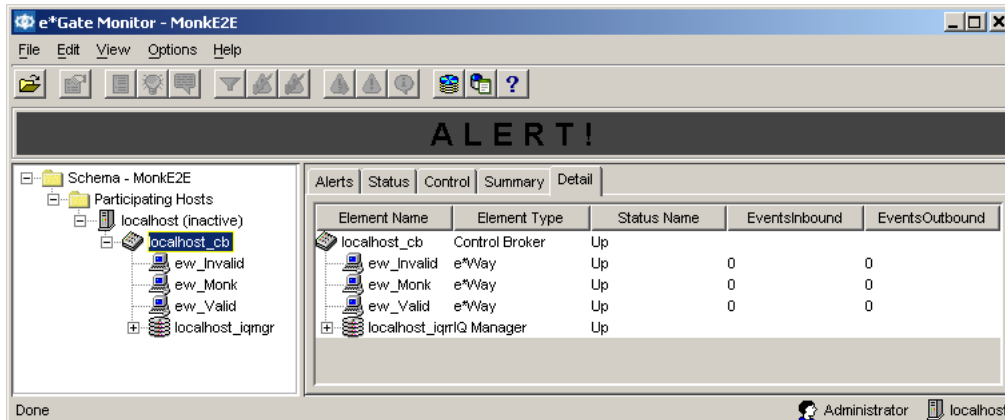

5.11 Testing the Schema

To test the schema

1 In Windows Explorer, open the *drive***:***eGatefolder***\\Client\\Data\\MonkE2E** folder, where *drive* is the drive where e*Gate is installed and where *eGatefolder* is the folder where e*Gate is installed. For example: c:\\eGate\\Client\\Data\\MonkE2E.

2 Change the file extension of **MonkE2Einput.txt** to **.fin**.

3 Click **Yes** when the confirmation message appears.

4 Verify that the extension changes to **.~in** indicating that the **ew_Monk** e*Way retrieved the file.

Almost immediately, the **Invalid_output0.dat** and **Valid_output0.dat** output files appear in the folder, indicating the schema ran successfully.

5 Verify that the output files contain the information shown in the screen below.
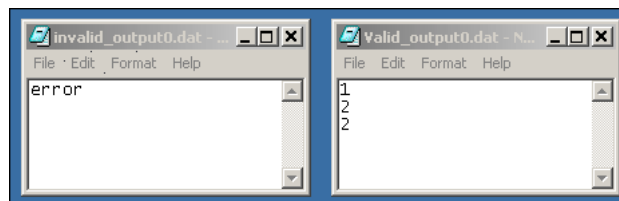
5.12 Troubleshooting

If you do not get the expected results, do the following:

- Double-check your schema. Use the table at **"Reviewing the Complete Schema" on page 87**. Make sure that you created and configured all the components as directed in the instructions.

- Use the e*Gate Monitor to confirm that all of the components are running; manually start any that may not have started correctly.

- If a component starts successfully but halts immediately, the most likely cause is misconfiguration. Check the following:

  ◆ Does each e*Way have an executable file and a configuration file associated with it?

  ◆ Does each Event Type have an ETD file assigned?

  ◆ Do the validating Collaboration Rules have the Monk service assigned, and the proper Collaboration Rules file both created and assigned?

  ◆ Is the IQ Manager running, and did you configure both IQs?

- If all of the components start and stay running, the most likely problem is that the validation logic in the **Inbound** e*Way's Collaboration Rules script is not working properly. Use the Collaboration Rules Editor and check that you have created the file in accordance with the procedure in **"Creating the cr_Monk Collaboration Rule Component" on page 78**.

If you encounter a problem and are unable to solve it with **"Reviewing the Complete Schema" on page 87**, refer to **"Debugging and Log Files" on page 92** to start the trace or error log component in your schema.

Debugging and Log Files

One of the easiest ways to debug your e*Gate configuration is through the use of log files. All executable components—BOBs, e*Ways, IQ Managers, and Control Brokers— have the ability to create log files that contain whatever level of debugging information you select.

6.1 Where To Find Log Files

All log files are stored in the **\eGate\client\logs** directory on the Participating Host running the elements that generate the log entries. Logs are named after the component that creates them; for example, the component **Inbound_eWay** creates a log file called **Inbound_eWay.log**.

6.2 Generating Log Files

To configure a component to generate a log file:

1 In the e*Gate Enterprise Manager window, select the component that you want to configure and display its properties.

2 Select the **Advanced** tab, then click **Log**.

3 Select the desired logging options (see **Figure 1 on page 93**).

Figure 1 Logging Options
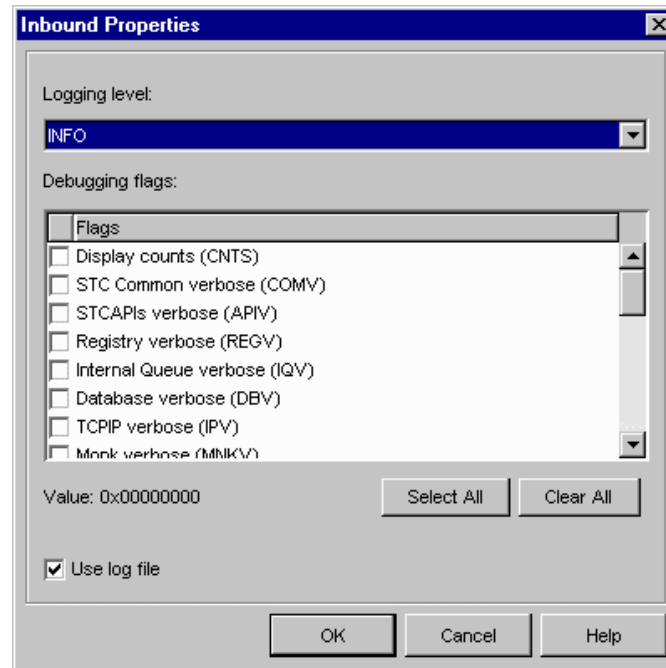
You can view a component's log using any text editor, and you can view the log while the component is still running. However, depending on the editor, you may need to re-read the file to "refresh" your view of the log data. You cannot get log updates "on the fly."

The most common error most first-time e*Gate developers will find in a log file is, "Unable to load module configuration." This message means that you have created an e*Gate component but not assigned both an executable file and a configuration file to it.

For more information about logging and debugging options, see the *e*Gate Integrator System Administration and Operations Guide.*

Index