

SeeBeyond ICAN Suite

e*Way Intelligent Adapter for JDE OneWorld GenJava User's Guide

Release 5.0.5 for Schema Run-time Environment (SRE)



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

SeeBeyond, e*Gate, e*Way, and e*Xchange are the registered trademarks of SeeBeyond Technology Corporation in the United States and/or select foreign countries. The SeeBeyond logo, SeeBeyond Integrated Composite Application Network Suite, eGate, eWay, eInsight, eVision, eXchange, eView, eIndex, eTL, ePortal, eBAM, and e*Insight are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2005 SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

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

Version 20050406010920.

Contents

Chapter 1

Introduction	5
Overview	5
Intended Reader	5
Components	5
Operational Overview	6
Supported Operating Systems	6
System Requirements	6
Host System Requirements	7
GUI Host Requirements	7
External System Requirements	7

Chapter 2

Installation	8
Installing on Windows Systems	8
Pre-installation	8
Installation Procedure	8
Installing on UNIX Systems	9
Pre-installation	9
Installation Procedure	9
Files/Directories Created by the Installation	9
Installing Business Functions	10
Setting Up the JDE OneWorld GenJava Environment	11
Setting Up the JDE OneWorld GenJava Environment on a Non-OneWorld Machine	11

Chapter 3

e*Way Connection Configuration	12
Creating e*Way Connections	12
e*Way Configuration Parameters	13
Connector	13
Type	14

class	14
Property.Tag	14
Connector Settings	14
user name	14
password	14
EnvironmentID	15
JDE Ini File	15

Chapter 4

Implementation	16
Implementation Overview	16
Importing the AddressBook Sample Schema	17
Creating and Configuring Event Types	17
Creating ETDs with the JDE ETD Wizard	17
The JDE OneWorld GenJava ETD	20
Collaboration Rules	21
To create Collaboration Rules	21
Running the Schema	24
JDE OneWorld GenJava Data Types	24

Chapter 5

JDE OneWorld GenJava e*Way Methods	26
com.stc.eways.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD Class	26
getN0100041	27
com.stc.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD.N0100041Wrapper Class	27
MyJDEBusinessFunctionsTestETD.N0100041Wrapper	27
com.stc.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD.N0100041Wrapper.AddressBookMasterMBFWrapper Class	28
cloneData	28
execute	28
getD0100041	29
getReturnValue	29

Index	31
--------------	-----------

Introduction

This document describes how to install and configure the e*Way Intelligent Adapter for JDE OneWorld.

This Chapter Includes:

- [“Overview” on page 5](#)
- [“Operational Overview” on page 6](#)
- [“Supported Operating Systems” on page 6](#)
- [“System Requirements” on page 6](#)
- [“External System Requirements” on page 7](#)

1.1 Overview

The JDE OneWorld GenJava e*Way allows the e*Gate system to exchange data with an OneWorld server by using GenJava generated Java wrapped classes.

1.1.1 Intended Reader

The reader of this guide is presumed:

- to be a developer or system administrator with responsibility for maintaining the e*Gate system
- to have intermediate-level knowledge of Windows-style operations and administration
- to be thoroughly familiar with JDE OneWorld
- to be thoroughly familiar with Windows-style GUI operations
- to be thoroughly familiar with GenJava

1.1.2 Components

The following components comprise the JDE OneWorld GenJava e*Way:

- **e*Way Connections:** The e*Way Connections provide access to the information necessary for connecting to a specified external system.

- **stcjdeowgenjava.jar**: Contains the logic required by the e*Way to interact with the JDE OneWorld GenJava server.

A complete list of installed files appears in [Table 1 on page 9](#).

1.2 Operational Overview

The Java-enabled version of the JDE OneWorld GenJava e*Way uses Java Collaborations to interact with one or more external JDE OneWorld GenJava server. By using the Java Collaboration Service it is possible for e*Gate components—such as e*Way Intelligent Adapters (e*Ways) and Business Object Brokers (BOBs)—to connect to the JDE OneWorld server and execute business rules.

An e*Gate component is defined as *Java-enabled* based on the selection of the Java Collaboration Service in the Collaboration Rule setup. For more information on the Java Collaboration Service, see the *e*Gate Integrator Collaboration Services References Guide*.

1.3 Supported Operating Systems

The JDE OneWorld GenJava e*Way is available on the following operating systems:

- Windows 2000 and Windows Server 2003
- HP-UX 11.0 and 11i (PA-RISC)
- IBM AIX 5.1L and 5.2
- Sun Solaris 8 and 9

Note: *Although the JDE OneWorld GenJava e*Way components can run on the supported platforms listed previously, the Java-enabled ETD Editor and Collaboration Editor require Windows.*

1.4 System Requirements

To use the JDE OneWorld GenJava e*Way, you need the following:

- An e*Gate Participating Host.
- A TCP/IP network connection.

The client components with which the e*Way interfaces have their own requirements; see that system's documentation for more details.

1.4.1 Host System Requirements

The system requirements are different for a GUI host machine—specifically a machine running the ETD Editor and the Java Collaboration Editor GUIs—versus a participating host which is used solely to run the e*Gate schema.

1.4.2 GUI Host Requirements

The JDE Client must be installed on any machine you run the JDE OneWorld GenJava Wizard from.

1.5 External System Requirements

To enable the JDE OneWorld GenJava e*Way to communicate properly with external JDE OneWorld GenJava server, the following are required:

- JDE OneWorld B7333

Installation

This chapter describes how to install the JDE OneWorld GenJava e*Way.

This Chapter Includes:

- [“Installing on Windows Systems” on page 8](#)
- [“Installing on UNIX Systems” on page 9](#)
- [“Files/Directories Created by the Installation” on page 9](#)

2.1 Installing on Windows Systems

2.1.1 Pre-installation

- 1 Exit all Windows programs before running the setup program, including any anti-virus applications.
- 2 You must have Administrator privileges to install this e*Way.

2.1.2 Installation Procedure

To install the JDE OneWorld GenJava e*Way on Windows systems

- 1 Log in as an Administrator on the workstation on which you want to install the e*Way.
- 2 Insert the e*Gate installation CD-ROM into the CD-ROM drive.
- 3 If the CD-ROM drive’s “Autorun” feature is enabled, the setup application should launch automatically; skip ahead to step 4. Otherwise, use Windows Explorer or the Control Panel’s **Add/Remove Applications** feature to launch the file **setup.exe** on the CD-ROM drive.

The InstallShield setup application will launch. Follow the on-screen instructions to install the e*Way.

Be sure to install the e*Way files in the suggested “client” installation directory. The installation utility detects and suggests the appropriate installation directory. **Unless you are directed to do so by STC support personnel, do not change the suggested “installation directory” setting.**

2.2 Installing on UNIX Systems

2.2.1 Pre-installation

You do not require root privileges to install this e*Way. Log in under the user name that you wish to own the e*Way files. Be sure that this user has sufficient privileges to create files in the e*Gate directory tree.

2.2.2 Installation Procedure

To install the JDE OneWorld GenJava e*Way on a UNIX system

- 1 Log in on the workstation containing the CD-ROM drive, and insert the CD-ROM into the drive.
- 2 If necessary, mount the CD-ROM drive.
- 3 At the shell prompt, type
`cd /cdrom`
- 4 Start the installation script by typing
`setup.sh`
- 5 A menu of options will appear. Select the “install e*Way” option. Then follow any additional on-screen directions.

Be sure to install the e*Way files in the suggested “client” installation directory. The installation utility detects and suggests the appropriate installation directory. **Unless you are directed to do so by SeeBeyond support personnel, do not change the suggested “installation directory” setting.**

2.3 Files/Directories Created by the Installation

The JDE OneWorld GenJava e*Way installation process will install the following files within the e*Gate directory tree. Files will be installed within the “egate\client” tree on the Participating Host and committed to the “default” schema on the Registry Host.

Table 1 Files created by the installation

e*Gate Directory	File(s)
	addonconpt.ini
configs\jdeowgenjava\	jdeowgenjava.def
classes\	jdeowgenjava.jar
classes\	stcjdeowgenjavabuilder.jar
	ewjdeowgenjava.ctl

Table 1 Files created by the installation

e*Gate Directory	File(s)
etd\	jdeowgenjava.ctl
etd\	jdeowgenjavawizard.ctl

2.4 Installing Business Functions

Before you are able to use the JDE OneWorld GenJava builder you will need to install the Business_Functions.zip file on your JDE OneWorld server. To do this do the following:

- 1 Run a complete backup of the JDE OneWorld Deployment server.
- 2 On the JDE OneWorld Deployment server, insert the e*Gate Addons installation disc into the CD ROM drive.
- 3 Navigate to **addons\ewjdeowgenjava\Business_Functions.zip**.
- 4 Unzip the **Business_Functions.zip** file into a temporary folder.
- 5 From the unzipped installation folder, run **InstallManager.exe**
- 6 Follow the JDE OneWorld Software Update steps to deploy the business functions.

Table 2 Files created by the Business_Functions.zip file

Temporary Folder	Files
work.CAB	Install CD 73.3.3\planner\Package\SEEBEYOND\
spec.CAB	Install CD 73.3.3\planner\Package\SEEBEYOND\
source.CAB	Install CD 73.3.3\planner\Package\SEEBEYOND\
obj.CAB	Install CD 73.3.3\planner\Package\SEEBEYOND\
lib32.CAB	Install CD 73.3.3\planner\Package\SEEBEYOND\
include.CAB	Install CD 73.3.3\planner\Package\SEEBEYOND\
data.CAB	Install CD 73.3.3\planner
bin32.CAB	Install CD 73.3.3\planner\Package\SEEBEYOND\
Setup.inf	Install CD 73.3.3\
Seebeyond.inf	Install CD 73.3.3\package_inf
Feature.inf	Install CD 73.3.3\
Deployment.inf	Install CD 73.3.3\
Disk1.sdf	Install CD 73.3.3\
jdeb7.mdb	Install CD 73.3.3\
Setup.exe	Install CD 73.3.3\
InstallManager.ico	Install CD 73.3.3\
Setup.hl	Install CD 73.3.3\

Temporary Folder	Files
Setup.ex	Install CD 73.3.3\
FBDCNFG.dl	Install CD 73.3.3\
InstallManager.ini	Install CD 73.3.3\
InstallManager.exe	Install CD 73.3.3\
jdeb7.ldb	Install CD 73.3.3\

For any additional information, please see your JDE OneWorld GenJava documentation.

2.5 Setting Up the JDE OneWorld GenJava Environment

To run the JDE OneWorld GenJava e*Way on a JDE OneWorld GenJava machine, do the following:

PATH

<bin directory for JDK>

For example: c:\jdk1.3.3\bin

Add the following to your classpath:

<Directory where JDE OneWorld is located>\System\classes\Kernel.jar
<Directory where JDE OneWorld is located>\System\classes\Connector.jar

Note: *The JDE OneWorld client must be installed on any machine you plan to run the JDE OneWorld GenJava Wizard from.*

2.5.1 Setting Up the JDE OneWorld GenJava Environment on a Non-OneWorld Machine

If you are running the JDE OneWorld GenJava e*Way on a non-OneWorld machine, you must install and set up the JDE OneWorld client by doing the following:

- 1 Copy the following files from the JDE OneWorld server to a directory on the non-JDE OneWorld machine.
 - A Kernel.jar
 - B Connector.jar
 - C Jas.ini (jdeinterop.ini).
- 2 Add **Kernel.jar** and **Connector.jar** to the CLASSPATH.
- 3 Edit **jas.ini** (jdeinterop.ini) for proper settings. Please see your JDE OneWorld documentation for more information regarding this file.

e*Way Connection Configuration

This chapter describes how to configure the JDE OneWorld GenJava e*Way Connections.

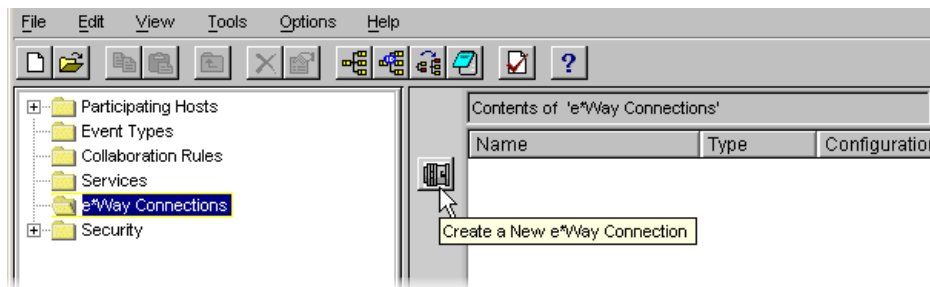
3.1 Creating e*Way Connections

The e*Way Connections are created and configured using the e*Gate Integrator Schema Designer graphical user interface (GUI).

To create and configure the e*Way Connections:

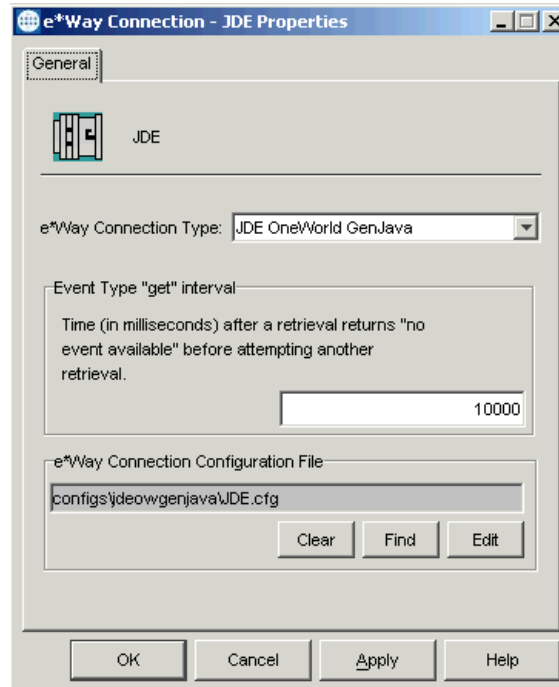
- 1 In the Schema Designer's Component editor, select the e*Way Connections folder.

Figure 1 The e*Way Connections Folder



- 2 On the Palette, click the **New e*Way Connection** icon.
- 3 The **New e*Way Connection Component** dialog box opens. Enter a name for the e*Way Connection and click **OK**.
- 4 Double-click the new e*Way Connection to open the e*Way Connection Properties dialog box. See [Figure 2 on page 13](#).

Figure 2 e*Way Connection Properties Dialog Box



- 5 From the e*Way Connection Type dropdown box, select **JDE OneWorld GenJava**.
- 6 Click **New** to create a new e*Way Connection Configuration File.

The e*Way Connection Configuration File Editor will appear.

The e*Way Connection configuration file parameters are organized into the following sections:

- Connector
- Connector Settings

3.2 e*Way Configuration Parameters

This section explains how to set the configuration parameters for the JDE OneWorld GenJava e*Way.

3.2.1 Connector

The Connection settings define the parameters used to interact with the JDE OneWorld server.

Type

Description

A default value of JDE OneWorld GenJava for connections to JDE OneWorld GenJava.

Required Values

Default value of **JDE OneWorld GenJava**.

class

Description

This parameter specifies the class name of the JDE OneWorld GenJava connector object.

Required Values

A valid class name.

The default is **com.stc.eways.jdeowgenjava.JDEOWGenJavaConnector**.

Property.Tag

Description

Parameter required by the current EBobConnectorFactory. Value identifies a data source.

Required Values

Any valid string.

3.2.2 Connector Settings

The Connector settings define the high level characteristics of the e*Way Connection.

user name

Description

Specifies the user name the e*Way will use to connect to the JDE OneWorld server.

Required Values

Any valid string.

password

Description

Specifies the password used to access the JDE OneWorld server.

Required Values

Any valid string.

EnvironmentID

Description

Specifies a valid EnvironmentID on the JDE OneWorld server.

Required Values

Any valid string.

JDE Ini File

Description

The qualified file name of the initialization file that specifies the JDE connection information.

Required Values

Any valid string.

Implementation

This chapter contains information pertinent to implementing the JDE OneWorld GenJava e*Way in a production environment. A complete sample schema is included on the CD. This chapter gives directions for executing the sample schema and gives examples of how those components were created and configured. It is assumed that the JDE OneWorld GenJava e*Way package has been successfully installed.

4.1 Implementation Overview

During installation, the Host and Control Broker were automatically created and configured. The names of the Host and Control Broker default to the name of the host machine on which the e*Gate Schema Designer GUI is installed.

To complete the implementation of the JDE OneWorld AddressBook Schema do the following

- Make sure that the Control Broker is activated.
- In the e*Gate Schema Designer, define and configure the following:
 - ♦ The JDE OneWorld client, as described in [Setting Up the JDE OneWorld GenJava Environment](#) on page 11
 - ♦ The Inbound and Outbound e*Ways using `stcewfile.exe`
 - ♦ The e*Way Connections, as described in [Creating e*Way Connections](#) on page 12.
 - ♦ Event Type Definitions used to package the data being exchanged with the external system.
 - ♦ Collaboration Rules to process Events.
 - ♦ Configure Collaborations within the e*Way component to apply the required Collaboration Rules.

The following sections describe how to define and associate the above components for a sample schema.

4.1.1 Importing the AddressBook Sample Schema

To import the sample schemas included on the cd, do the following:

- 1 On the **Schema Designer** toolbar, click **File, Import Definitions from File...**
- 2 Review the information provided in the **Import Wizard - Introduction** window. Click **Next**.
- 3 On the **Import Wizard - Step 1** window, select **Schema** and click **Next**.
- 4 On the **Import Wizard - Step 2** window, browse to the **ewjdeowgenjava** sample folder.
- 5 Double click the **ewjdeowgenjava** sample folder and select the **JDEOWSample.zip** file.
- 6 Click **Open**.
- 7 On the **Import Wizard - Step 2** window, click **Next**.
- 8 On the **Import Wizard - Step 3** window, you can rename the host or change the port. This step is optional.
- 9 Click **Next**.
- 10 On the **Import Wizard - Finish** window, click **Finish**.

4.1.2 Creating and Configuring Event Types

An Event Type Definition (ETD) is a graphical representation of the layout of data in an Event. The following are examples of how event types are created and configured.

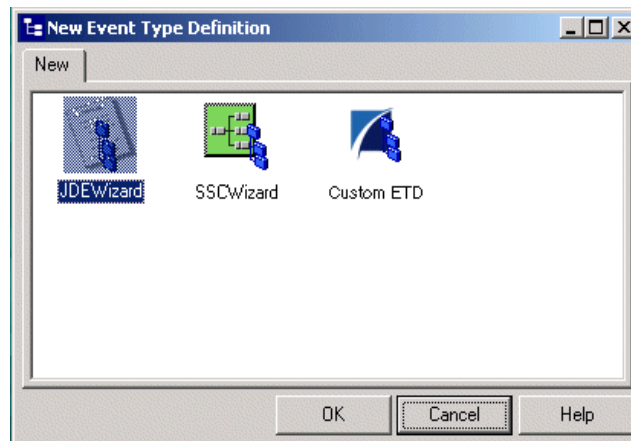
Creating ETDs with the JDE ETD Wizard

To use the JDE ETD wizard

- 1 Highlight the **Event Types** folder on the **Components** tab of the e*Gate Schema Designer.
- 2 On the palette, click the **Create a New Event Type** button.
- 3 Enter the name of the **Event**, then click **OK** (for the purpose of this sample, the first Event Type is defined as **node**).
- 4 Double-click the new Event Type to edit its properties.
- 5 When the **Properties** window opens, click the **New** button. The ETD Editor opens.
- 6 Select **New** from the **File** menu on **Task Manager**.
The **New Event Type Definition** dialog box appears.

- 7 Select the icon for the **JDEWizard**.

Figure 3 New Event Type Definition Dialog Box



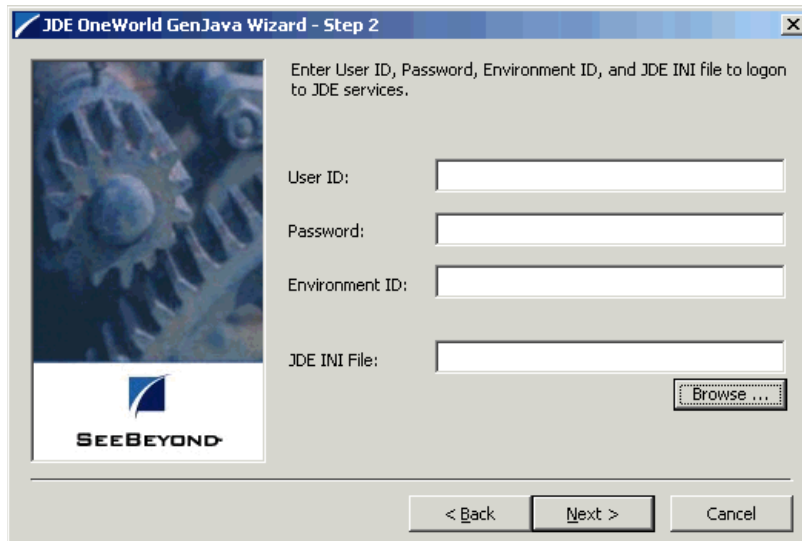
- 8 Enter the Package Name where the ETD builder can place all the generated Java classes associated with the created ETD (For this case, **com.stc.jdeowgenjava.samplepckg**).
- 9 Enter the Root Node Name (for this case, **sampleRootETD**).

Figure 4 JDE OneWorld GenJava Wizard - Step 1



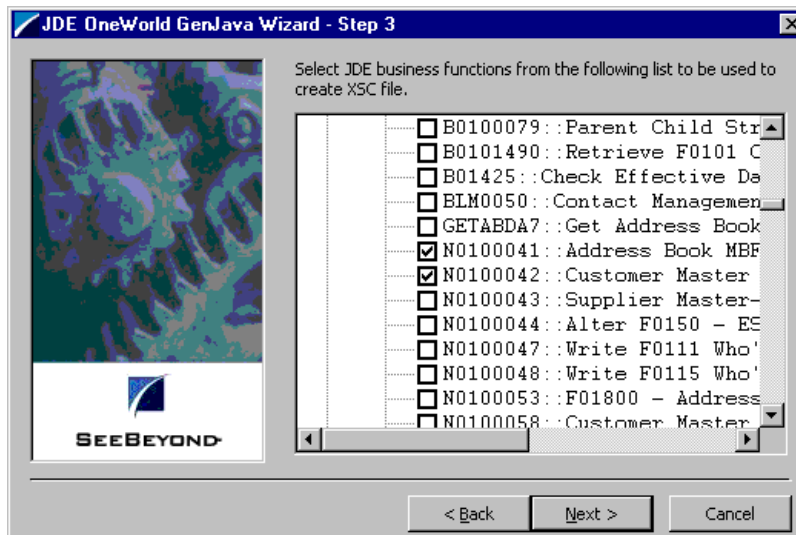
- 10 Click **Next**.
- 11 Enter your **User ID**, **Password**, **EnvironmentID** and the JDE **.ini** file to access JDE services.

Figure 5 JDE OneWorld GenJava Wizard - Step 2



- 12 Click Next.
- 13 Select the JDE functions from the list to be used to create your .xsc file.

Figure 6 JDE OneWorld GenJava Wizard - Step 3



- 14 Click Next.
- 15 After reviewing your selections, click **Finish** to generate your .xsc file.

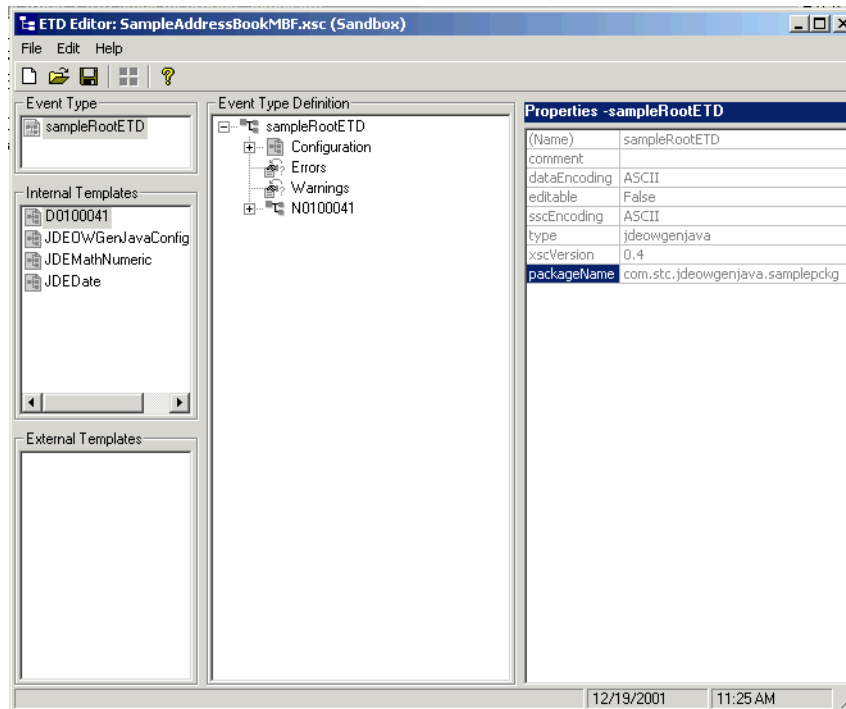
Figure 7 JDE OneWorld GenJava Wizard - Step 4



4.2 The JDE OneWorld GenJava ETD

The JDE OneWorld GenJava ETD does not allow you to make any edits to the data while in the ETD. This ensures the integrity of the data.

Figure 8 ETD Editor



4.2.1 Collaboration Rules

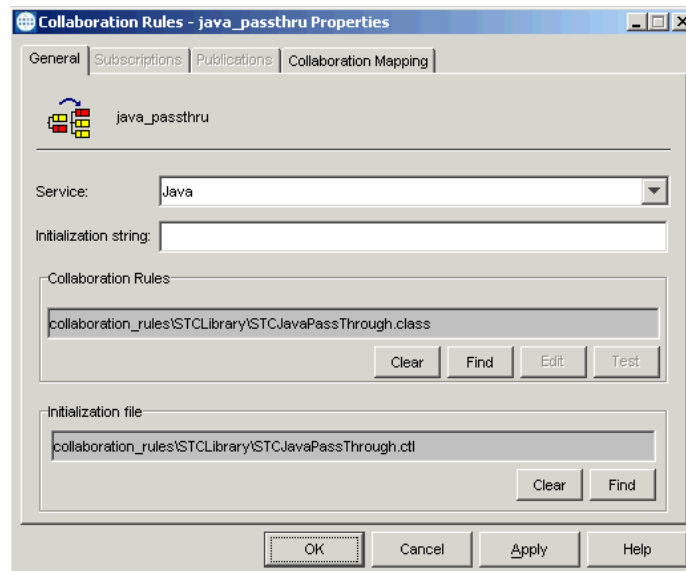
The next step is to create the Collaboration Rules that will extract selected information from the source Event Type defined earlier. The Collaboration Rules will then process the extracted information according to its associated Collaboration Service.

To create Collaboration Rules

Creating the Collaboration Rules

- 1 Select the Navigator's **Components** tab in the e*Gate Schema Designer.
- 2 In the **Navigator**, select the **Collaboration Rules** folder.
- 3 On the palette, click the **Create New Collaboration Rules** button.
- 4 Enter the name of the new Collaboration Rule (for this case **retrieveData**).
- 5 Double-click the new Collaboration Rule to edit its properties. The **Collaboration Rules Properties** dialog box opens.
- 6 The Service defaults to Java. The Collaboration Rules will use the Java Service to manipulate Events or Event data. The **Subscriptions** and **Publications** tabs are disabled and the **Collaboration Mapping** tab is enabled.

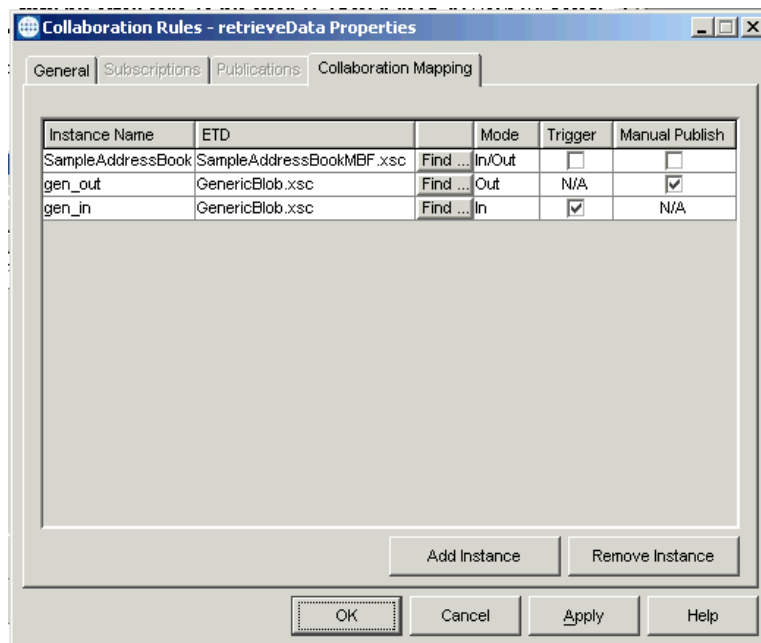
Figure 9 Collaboration Rules Properties Dialog Box



- 7 In the **Initialization string** field, enter any required initialization string that the Collaboration Service may require. This field can be left blank.
- 8 Click on the **Collaboration Mapping** tab.
- 9 Click on the **Add Instance** button to create instances that correspond to the event types. In the **Instance Name** column type **SampleAddressBook** as the instance name.
- 10 Click the **Find** button and navigate to and select **SampleAddressBookMBF.xsc**. This adds **SampleAddressBookMBF.xsc** to the **ETD** column of the instance row.

- 11 In the **Mode** column, click the right side of the field to open a drop down list box. Select **In/Out** as the Mode.
- 12 Make sure the **Trigger** and **Manual Publish** check boxes are cleared.
- 13 Click on the **Add Instance** button again to create another instance.
- 14 In the **Instance Name** column type **gen_out** as the instance name.
- 15 Click the **Find** button and navigate to and select **GenericBlob.xsc**. This adds **GenericBlob.xsc** to the **ETD** column of the instance row.
- 16 In the **Mode** column, click the right side of the field to open a drop down box. Select **Out** from the drop-down menu.
- 17 In the **Manual Publish** column, click the check box to enable.
- 18 Click on the **Add Instance** button again to create another instance.
- 19 In the **Instance Name** column type **gen_in** as the instance name.
- 20 Click the **Find** button and navigate to and select **GenericBlob.xsc**. This adds **GenericBlob.xsc** to the **ETD** column of the instance row.
- 21 In the Mode column, click the right side of the field to open a drop down box. Select **In** from the drop-down menu.
- 22 In the **Trigger** column, click the check box to enable the trigger mechanism.

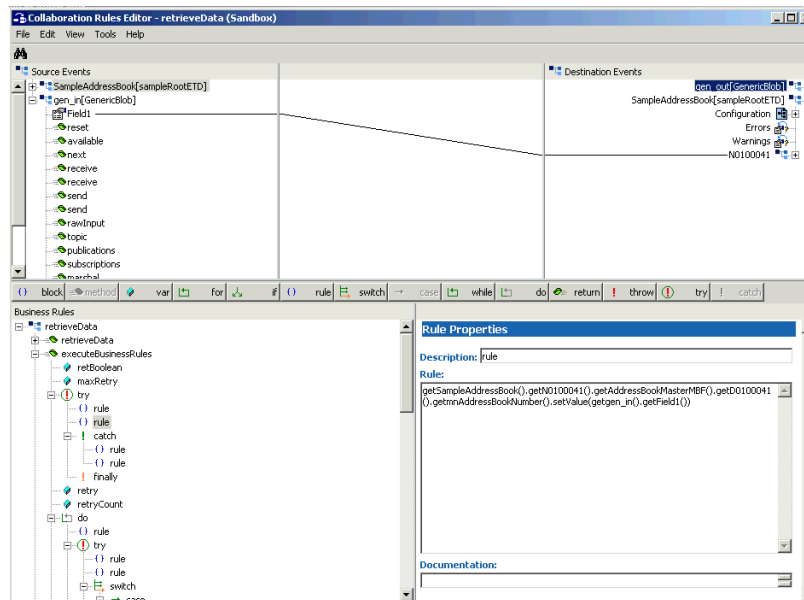
Figure 10 Collaboration Rules Properties, Collaboration Mapping



- 23 Click **Apply**.
- 24 Click **OK**.
- 25 From the Schema Designer toolbar, click the **Collaboration Editor** button.

- 26 From the **File** menu, click **Open**, <name of your schema>, **retrieveData.xpr**. Where <name of your schema> is the name you gave your schema.
- 27 Expand the **Source Events** fields and the **Destination Events** fields to view components. In the **Business Rules** pane, select **retBoolean**.
- 28 Drag-and-drop **gen_in[GenericBlob]** from the Source Events pane to **sampleAddressBook[sampleRootETD]** under Destination Events pane. This should cause a line to display between the two components. A **rule** is added in the Business Rules pane under **retBoolean** and the Java code is displayed in the **Rule Properties** pane in the **Rules** window.

Figure 11 Collaboration Rules Editor, retrieveData



- 29 In the **Initialization string** field, enter any required initialization string that the Collaboration Service may require.
- 30 Drag-and-drop a new rule under **userInitialize** and enter the necessary login parameters to log into the JDE OneWorld server.
- 31 In the **Termination string** field, enter any required termination string that the Collaboration Service may require.
- 32 Drag-and-drop a new rule under **userTerminate** and enter the logoff parameter to logoff of the JDE OneWorld server.
- 33 From the **File** menu click **Compile**. The **Compile** pane opens at the bottom of the Collaboration Rules Editor, displaying whether compile is successful or unsuccessful due to errors. If the compile is successful save, promote the file, and exit the Collaboration Rules Editor.

4.2.2 Running the Schema

To run the schema

- 1 Go to the command line prompt, and enter the following:

```
stccb -rh hostname -rs schemaname -un username -up user password  
-ln hostname_cb
```

Substitute *hostname*, *username* and *user password* as appropriate.

- 2 Exit from the command line prompt, and start the Schema Manager GUI.
- 3 When prompted, specify the *hostname* which contains the Control Broker you started in Step 1 above.
- 4 Select the schema.
- 5 After you verify that the Control Broker is connected (the message in the Control tab of the console will indicate command *succeeded* and status as *up*), highlight the IQ Manager, *hostname_igmgr*, then click on the right button of the mouse, and select **Start**.
- 6 Highlight each of the e*Ways, right-click the mouse, and select **Start**.

4.2.3 JDE OneWorld GenJava Data Types

Each JDE OneWorld GenJava ETD has four specific data types in common. They are:

- Warnings: Given by JDE OneWorld GenJava. Please see your JDE OneWorld GenJava documentation for more information regarding warnings.
- Errors: Thrown by JDE OneWorld GenJava. Please see your JDE OneWorld GenJava documentation for more information regarding errors.
- JDEMathNumeric: Data type set within JDE OneWorld GenJava. See “JDEMathNumeric” on page 25.
- JDEDate: Data type set within JDE OneWorld GenJava (see “[JDEDate](#)” on page 25).

JDEMathNumeric

getValue()	return the value as a String (for example “–12345.6789”)
setValue(String strValue)	set the value from a String (for example “–12345.6789”)
getCurrencyDecimals()	Get the Currency Decimal positions
setCurrencyDecimals(intaValue)	Set the Currency Decimal positions
getCurrencyCode()	Get the Currency Code
setCurrencyCode(StringaValue)	Set the Currency Code
setYear (short)	Set the year of the date
getDecimalPosition()	Get the Decimal Position
isNegative()	Test if the value is negative
reset()	Reset all the internal values

JDEDate

JDEDate()	Construct a JDEDate
getDay()	Get the day of the date
getMonth()	Get the month of the date
getYear()	Get the year of the date
setDay(short)	Set the day of the date
setMonth(short)	Set the month of the date
setYear (short)	Set the year of the date

JDE OneWorld GenJava e*Way Methods

The JDE OneWorld GenJava e*Way sample schema contains Java methods that are used to extend the functionality of the e*Way. These methods are contained in the following classes:

- [com.stc.eways.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD Class](#) on page 26
- [com.stc.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD.N0100041Wrapper Class](#) on page 27
- [com.stc.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD.N0100041Wrapper.AddressBookMasterMBFWrapper Class](#) on page 28

For a list of the methods and objects created by JDE OneWorld GenJava, please refer to your JDE OneWorld GenJava documentation.

5.1 com.stc.eways.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD Class

```

java.lang.Object
|
+--com.stc.jcsre.SimpleETDImpl
    |
    +--com.stc.eways.jdeowgenjava.JDEOWGenJavaETDBase
        |
        +--com.stc.jdeowgenjava.jdebusinessfunctionstest
            .MyJDEBusinessFunctionsTestETD
  
```

All Implemented Interfaces

com.stc.jcsre.ETD, comstc.jcsre.ETDConstants

Direct Known Subclasses

```

public class MyJDEBusinessFunctionsTestETD()
extends JDEOWGenJavaETDBase
  
```

Default constructor, creates a new ETD implementation instance.

Methods of the MyJDEBusinessFunctionsTestETD Class

getN0100041

getN0100041

This method gets the SeeBeyond JDE OneWorld GenJava builder generated JDE Business Function wrapper instance associated with this ETD implementation instance.

```
get[JDE business function name]()
```

Returns

JDE business function wrapper instance associated with this ETD implementation.

Throws

java.sql.SQLException

5.2 com.stc.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD.N0100041Wrapper Class

```
java.lang.Object
```

```
|
```

```
+ --com.stc.jdeowgenjava.jdebusinessfunctionstest  
   .MyJDEBusinessFunctionsTestETD.N0100041Wrapper
```

Direct Known Subclasses

```
public class MyJDEBusinessFunctionsTestETD.N0100041Wrapper  
extends java.lang.Object
```

Methods of the com.stc.jdeowgenjava.jdebusinessfunctionstest. .MyJDEBusinessFunctionsTestETD.N0100041Wrapper Class

```
getAddressBookMasterMBF
```

MyJDEBusinessFunctionsTestETD.N0100041Wrapper

Gets the SeeBeyond JDE OneWorld GenJava builder generated JDE Function wrapper instance associated with this JDE business function wrapper instance. The name of the Function is embedded in the method name.

```
public get[JDE function name]()
```

Returns:

JDE function wrapper instance associated with this business function wrapper instance.

5.3 com.stc.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD.N0100041Wrapper.AddressBookMasterMBFWrapper Class

```

java.lang.Object
|
+ -- com.stc.jdeowgenjava.jdebusinessfunctionstest
      .MyJDEBusinessFunctionsTestETD.N0100041Wrapper
      .AddressBookMasterMBFWrapper

public class
MyJDEBusinessFunctionsTestETD.N0100041Wrapper.AddressBookMasterMBFWra
pper
extends java.lang.Object

```

Methods of com.stc.jdeowgenjava.jdebusinessfunctionstest.MyJDEBusinessFunctionsTestETD.N0100041Wrapper.AddressBookMasterMBFWrapper Class

```

cloneData
execute
getD0100041
getReturnValue

```

cloneData

Clones the data in the data structure associated with this JDE function and returns this deep copy. As the data structure associated with the business function is used to provide the in and output parameters for the JDE function calls, it can be useful to clone the data to preserve it for future use.

```
public D0100041 cloneData()
```

Returns

A deep copy of the data structure - deep means that it will fully copy fields with customer JDE data types such as JDEDate and JDEMathNumeric too.

Throws

IOException - Thrown if there was any problems with cloning the data structure. It is an IOException because streams are used to copy it.

execute

Executes the JDE function wrapper by this class on the JDE OneWorld server. This calls the corresponding method in the JDE GenJava generated classes. This method also populates the field return Value with the return value of the call to JDE.

```
public void execute()
```

Returns

Return values include:

0 = success

1 = warnings

2 = errors

-1 = field is initialized to this value, indicates that execute() has not been called yet successfully. Further details can be retrieved from the Warnings and Errors repeating nodes in the XSC, which are implemented in the JDEOWGenJavaETDBase class, e.g. getWarnings(), getErrors().

Throws

com.stc.common.collabService.CollabConnException - the execution of the JDE function failed, for example if the connection to the JDE server went down, the server went down, the user didn't log in before calling execute or the connection timed out as configured in the jas.ini file. For issues with the data supplied in the data structure NO exception is thrown, but the return Value will indicate whether there are warnings or errors that can be retrieved from the Errors or Warnings nodes in the XSC.

getD0100041

Gets the JDE GenJava generated JDE Data Structure instance associated with this JDE business function wrapper instance. The name of the Data Structure is embedded in the method name as get[JDE Data Structure name](). The same data structure and the same instance is used as the input and output parameter to the JDE function call, meaning that one would populate for example some fields in the data structure required for a query, call execute() and the result will be filled into the same data structure instance. This means that fields will get overwritten with each preparation/call of the JDE functions.

```
public D0100041 getD0100041()
```

Returns

The JDE data structure instance associated with this JDE function wrapper instance.

Throws

None

getReturnValue

Gets the return value populated by the last call to execute() for this JDE function.

```
public int getReturnValue()
```

Returns

Return values include:

0 = success

1 = warnings

2 = errors

-1 = field is initialized to this value. Indicates that execute() has not been called yet successfully. Further details can be retrieved from the Warnings and Errors repeating nodes in the XSC, which are implemented in the JDEOWGenJavaETDBase class, e.g. getWarnings(), getErrors().

Index

C

- class parameter
 - Data Source settings 14
- Collaboration Rules 21
- com.stc.eways.jdeowgenjava.JDEOWGenJavaConnector 14
- configuration parameters
 - class 14
 - password 14
 - user name 14
- configuring e*Way parameters 13
- Connector 13
- Connector Settings 14
- creating e*Way connections 12

E

- e*Way connections
 - configuring 13
 - creating 12
- event types
 - from DTD 17
- External System Requirements 7

G

- GUI Host Requirements 7

H

- host system requirements 7

I

- implementation 16
 - Collaborations Rules 21
 - overview 16
- Importing the AddressBook Sample Schema 17
- Installation 8
- installation
 - UNIX 9
 - Windows 8

J

- JDE ETD wizard 17
- JDE OneWorld GenJava Environment on a Non-OneWorld Machine 11
- JDEMathNumeric 25

P

- password parameter 14

R

- requirements
 - host system 7
 - system 6

S

- sample schema 24
- Setting Up the JDE OneWorld GenJava Environment 11
- supported operating systems 6
- system requirements 6

U

- UNIX 9
- user name parameter 14

W

- Windows 5, 6