

SeeBeyond ICAN Suite

CICS eWay Intelligent Adapter User's Guide

Release 5.0.4



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.

© 2004 by 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 This work is protected as an unpublished work under the copyright laws.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 20040625114545.

Contents

Chapter 1

Introducing the CICS eWay	7
Overview	7
CICS Transaction Server	7
The eWay Intelligent Adapter for CICS	8
IBM CICS Transaction Gateway (CTG)	8
SeeBeyond CICS Listener (STCL)	10
z/OS CICS Security Considerations	11
Security Considerations for SeeBeyond CICS Listener	11
Security Considerations for IBM CICS Transaction Gateway	12

Chapter 2

Installing the CICS eWay	13
Supported Operating System	13
Also Supported	13
System Requirements	14
External System Requirements	14
Requirements for the CICS Server using CICS Transaction Gateway	14
Requirements for the CICS Server Using the SeeBeyond CICS Listener	14
SeeBeyond CICS Listener Requirements for Invoking DB2 Applications	15
Application Server Support	15
Installing the CICS eWay	15
Installing the CICS eWay on an eGate Supported System	15
After Installation	16
SeeBeyond CICS Listener Installation for z/OS	16
Downloading the SeeBeyond CICS Listener Zip File	16
Installing the SeeBeyond CICS Listener from CD for z/OS	17
Installing the SeeBeyond CICS Listener from 3480 Tape	18
Copying the Tape Contents to Disk	18
Creating the STCLCFG File	19
Installing the CICS CEDA Definitions	19
Adding the CICS eWay Load Modules to CICS DFHRPL Concatenation	19
SeeBeyond CICS Listener Configuration Maintenance for z/OS	20
The SeeBeyond CICS Listener Monitor Screen for z/OS	25

CICS Listener Considerations for Invoking DB2 Applications	36
RCT entry for the STCL SeeBeyond CICS Listener Transaction	37
PPT entry to redirect DB2 application to another AOR	37
CICS Transaction Gateway 5.1 Installation and Configuration	38

Chapter 3

Configuring the CICS eWay **39**

Creating and Configuring the CICS eWay	39
Selecting CICS as the External Application	39
Modifying the CICS eWay Properties	40
Using the Properties Sheet	41
CICS eWay Connectivity Map Configuration Properties	42
Connector	42
Class	42
Connection Transport	43
Property.Tag	43
Type	43
CICS Client	43
CICS Program	44
CICS TransId	44
COMMAREA Length	44
ECI call type	44
ECI extend mode	44
Encoding	45
CICS eWay Environment Explorer Configuration Properties	46
SeeBeyond CICS Listener	46
COMMAREA Padding Character	46
Host	47
KeepAlive	47
Listener Timeout	47
NoDelay	47
Polling Rate	48
Port	48
ReceiveBufSize	48
SeeBeyond CICS Listener TransId	48
SendBufSize	48
Start Delay	49
Start Type	49
TP Timeout	49
Transport Timeout	49
CICS Gateway	49
Port	50
SSL KeyRing Class	50
SSL KeyRing Password	50
URL	50
Tracing	51
Dump Offset	51
Filename	51
Level	51

Timing	52
Truncation Size	52
CICS Client	52
CICS Password	52
CICS UserId	53
Alerting and Logging	53

Chapter 4

Using eInsight Business Processes with the CICS eWay	54
The eInsight Engine and Components	54
The CICS eWay With eInsight	55
CICS CTG Concerns	55
Defining More Than One Server When Using CTG	55
Matching the COMMAREA Size	56
The CICS eWay eInsight Sample Project	56
Importing a Sample Project	56
The CICS_BP_Sample Project Overview	57
Creating the CICS_BP_Sample Project	58
Creating a Project	58
Creating a Business Process	58
Creating the Business Process Flow	58
Configuring the Modeling Elements	60
Creating a Connectivity Map	62
Selecting the External Applications	63
Populating the Connectivity Map	63
Binding the eWay Components	64
Creating an Environment	65
Configuring the eWay Properties	66
Configuring the File eWay Properties	66
Configuring the CICS eWay Properties	67
Creating and Activating the Deployment Profile	68
Running the Project	69

Chapter 5

Implementing a CICS eWay Project	71
CICS eWay Components	71
CICS eWay Sample Projects	72
CICS CTG Concerns	72
Defining More Than One Server When Using CTG	72
Matching the COMMAREA Size	73
Importing a Sample Project	73

The CICS_JCE_Sample Project Overview	73
Creating the CICS_JCE_Sample Project	74
Creating a Project	74
Creating a Connectivity Map	74
Selecting the External Applications	74
Populating the Connectivity Map	75
Creating the Collaboration Definitions	76
Using the Collaboration Editor (Java)	77
Creating the CICSjms_Collab Business Rules	77
Creating the JMS2File_Collab Collaboration Business Rules	81
Binding the eWay Components	82
Creating an Environment	84
Configuring the eWays	85
Configuring the File eWays	85
Configuring the CICS eWay	86
Configuring the JMS Clients	87
Creating and Activating the Deployment Profile	88
Running the Project	89
The CICS_COBOL_Sample Project	90
Creating a Project	90
Creating a Connectivity Map	91
Selecting the External Applications	91
Populating the Connectivity Map	91
Creating an Object Type Definition	92
Using the Cobol Copybook Wizard	92
Creating a Collaboration Definition (Java)	93
The CICS_COBOL_Collab Business Rules	93
Creating Collaboration Bindings	101
Creating an Environment	102
Configuring the eWays	103
Configuring the File eWays	103
Configuring the CICS eWay	103
Creating and Activating the Deployment Profile	104
Running the Project	105

Chapter 6

Java Methods	106
The CicsClient Class	106
CICS Javadoc	106
Packed Decimal Java Helper Methods	106
ContainerExists	107
CopyBack	107
CopyTo	108
GiveElem	108
SetElem	109
Index	110

Introducing the CICS eWay

This chapter includes a brief description of IBM's Customer Information Control System™ (CICS™), and an overview of the SeeBeyond CICS eWay Intelligent Adapter.

Chapter Topics

- [Overview](#) on page 7
- [z/OS CICS Security Considerations](#) on page 11

1.1 Overview

1.1.1. CICS Transaction Server

IBM's Customer Information Control System (CICS), is IBM's most widely used proprietary, transaction monitor. CICS provides connectivity and online transaction management for mission-critical applications. It supports real-time distributed processing environments and online transaction processing (OLTP). According to IBM, CICS handles more than thirty billion transactions, processing more than one trillion dollars, per day.

CICS is the premier OLTP (On-Line Transaction Processing) product from IBM. It is used to access many file systems and databases including third party products. For IBM product, it interfaces with DB2, VSAM and IMS/DB. For non-IBM products, it interfaces with IDMS, ADABAS, DATACOM, to name a few. Most applications in CICS are written in COBOL, although it supports other languages such as PL/1.

OLTP systems provide accurate, up-to-date information within seconds, from terminals that give direct access to data held as either files or databases. CICS provides a company with numerous transaction processing and resource management functions, allowing the user to concentrate on developing application programs that meet that organization's specific business needs. CICS controls OLTP application programs in a distributed transaction processing (DTP) environment. CICS handles interactions between the terminal user and the application programs. Programs gain access to the CICS facilities with straightforward, high-level commands.

CICS provides:

- Communication functions to terminals and systems required by application programs

- Control of concurrently running programs serving online users
- Facilities for accessing databases and files
- The ability to communicate with other CICS family members using Transmission Control Protocol/Internet Protocol (TCP/IP)
- Interactive facilities to configure specific systems
- Recovery processing and data protection, should a problem occur

1.1.2. The eWay Intelligent Adapter for CICS

The eWay Intelligent Adapter for CICS is an interface that enables remote bidirectional calls to CICS transactional programs. The CICS eWay includes a build tool, the Cobol Copybook Converter, that creates an Object Type Definition (OTD) from a Cobol Copybook file and generates eGate Object Type Definitions for use within the eGate environment. The Copybook file structures are passed into the CICS environment as the data buffer (Commarea).

A fixed Object Type Definition, the **CICS_eWay.CICSClient OTD**, designed to expose various essential portions of the CICS Java API, provides available methods and properties, as well as access to all message attributes.

The eWay can use either the IBM **CICS Transaction Gateway 5.1** or the **SeeBeyond CICS Listener** as the underlying connection transport for accessing CICS z/OS transactions.

IBM CICS Transaction Gateway (CTG)

CTG provides an API (the External Call Interface or ECI) to call CICS transactions on the mainframe. The ECI allows a non-CICS application program to call a CICS program in a CICS server. SeeBeyond's CICS eWay uses this ECI method to connect to CICS. The CICS eWay connects to CICS with CTG running on a local-host (Figure 1), on a second computer (Figure 2 on page 9), or on the mainframe (Figure 3 on page 9).

Figure 1 eGate and CTG running on the same host

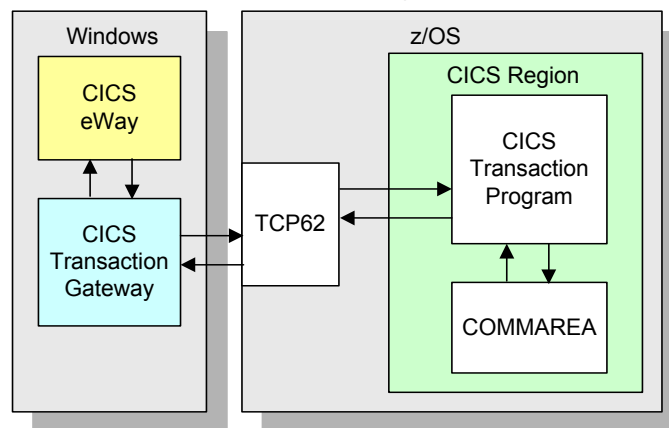


Figure 2 Remote connection with CTG on a UNIX or Windows host

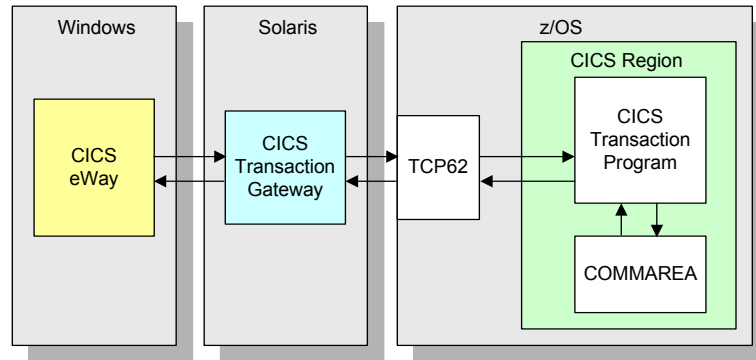
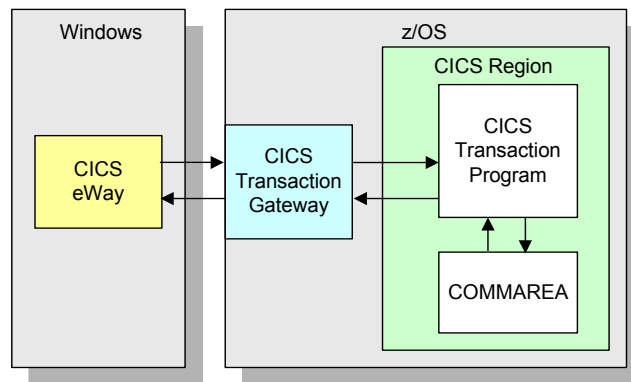


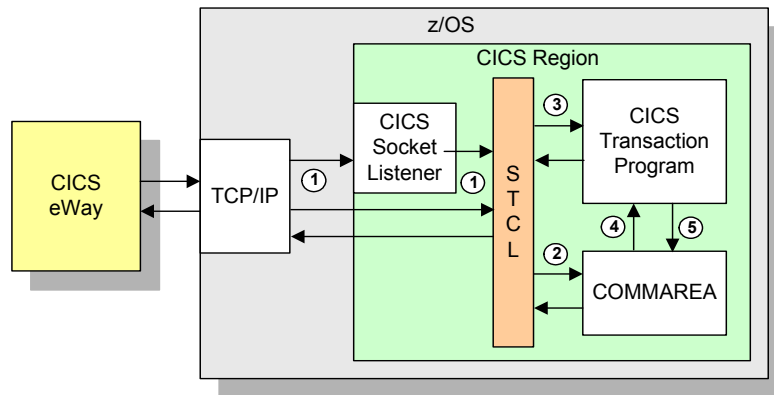
Figure 3 eGate connects with CTG running on the mainframe



SeeBeyond CICS Listener (STCL)

The CICS eWay connects to the IBM CICS Listener running on z/OS via the TCP/IP Sockets. The Listener accepts the incoming request and spawns a new process handling the socket connection off to the newly created process via TCP/IP givesocket()/takesocket() function calls. The spawned process invokes the user written CICS application program through an EXEC CICS LINK.

Figure 4 Using the SeeBeyond CICS Listener for Synchronous Transactions



The CICS eWay (Java version) communicates with SeeBeyond CICS Listener for Synchronous Transactions (See Figure 4) as follows:

- 1 An incoming Connect request is handled by the IBM CICS Socket Listener, which starts the SeeBeyond CICS Listener Transaction STCL and hands off the incoming connection via the IBM TCP/IP Give Socket and Take Socket interface.
- 2 The SeeBeyond CICS Listener allocates a CICS COMMAREA and copies information from the CICS eWay COMMAREA to the actual CICS COMMAREA.
- 3 The SeeBeyond CICS Listener issues an EXEC CICS LINK to requested CICS Transaction Program passing it the newly allocated COMMAREA.
- 4 The requested CICS Transaction obtains data from the COMMAREA, performs typical business rule processing and then returns its results in the COMMAREA and returns control back to the SeeBeyond CICS Listener.
- 5 The SeeBeyond CICS Listener copies information from the CICS COMMAREA back to the CICS eWay COMMAREA.
- 6 The SeeBeyond CICS Listener goes into a listen mode and waits for the next incoming Transaction Program request.

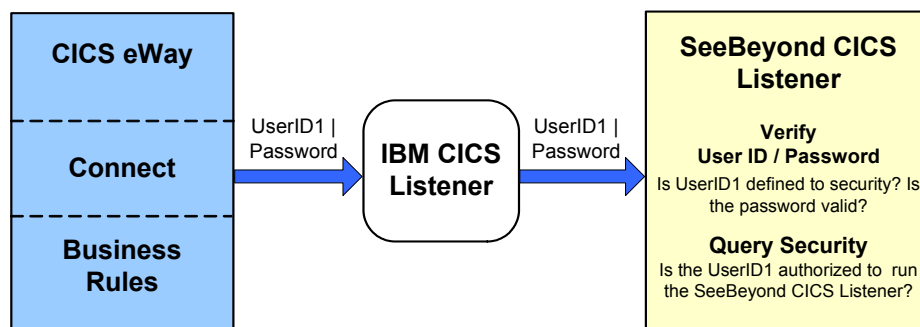
The process continues until the SeeBeyond CICS Listener Timeout is exceeded or a disconnect request is received from the CICS eWay.

1.2 z/OS CICS Security Considerations

Security Considerations for SeeBeyond CICS Listener

The CICS eWay, using the SeeBeyond CICS Listener as the underlying connection transport, utilizes three modes of security with z/OS: Connection Logic, Request Link to Program, and Request Start Transaction. The userID and password are defined in the eWay properties file. The connection manager uses the userID and password in the properties file to start the SeeBeyond CICS Listener on z/OS. During Business Rules processing, requests that flow into the SeeBeyond CICS Listener can use the userID and password from the properties file, or can be overwritten in the Collaborations.

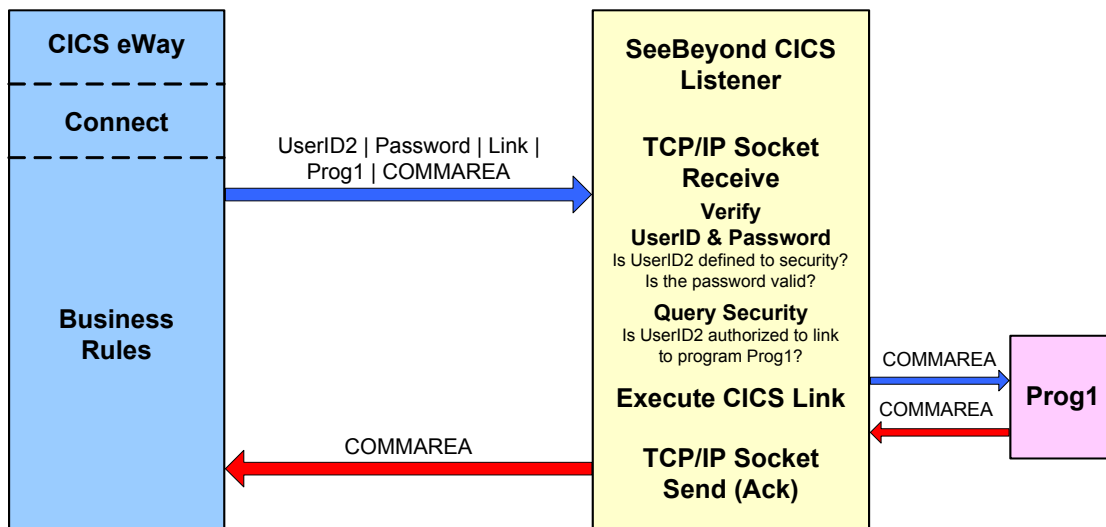
Figure 5 Connection Logic



Connection Logic

For the Connection Logic mode (Figure 5), the userID and password, passed from the CICS eWay through the IBM CICS listener and into the SeeBeyond CICS Listener, must be defined for the z/OS security system (RACF, for example). The userID must be authorized by the z/OS security system to run CICS transaction “xxxx” inside of CICS. The default value for “xxxx” is STCL, and can be changed in the properties of the Connection Manager in the CICS eWay.

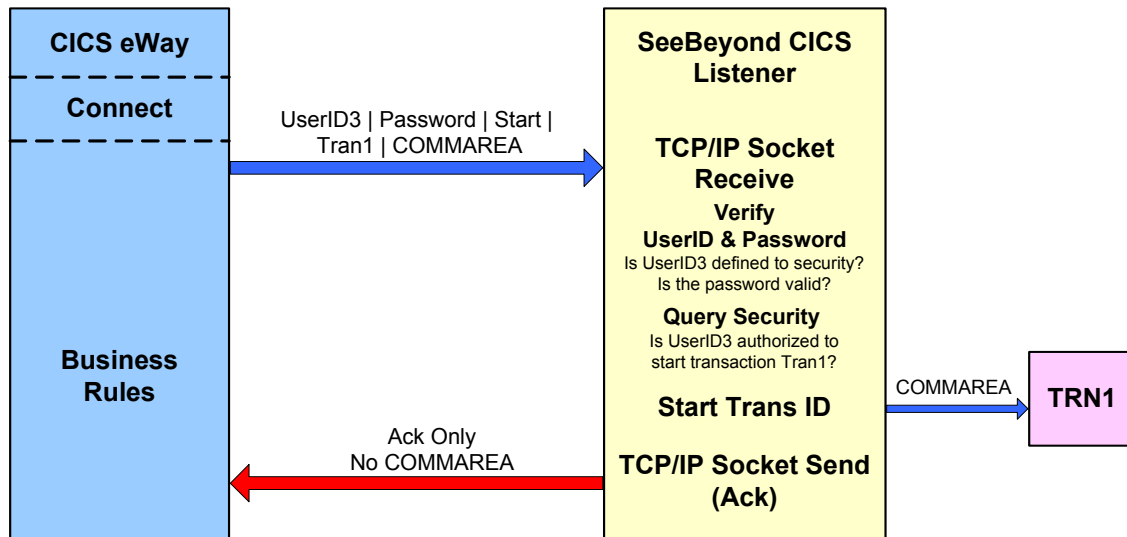
Figure 6 Business Rules Request to Program



Request Link to Program

For the Request Link to Program mode (Figure 6 on page 11), the userID and password passed from the CICS eWay to the SeeBeyond CICS Listener must be defined for the z/OS security system (RACF, for example). The userID must be authorized by the z/OS security system to run CICS program “prog1” inside of CICS. The default value for “prog1” is set in the properties of the CICS eWay, and can be overridden in the Collaboration for each request sent into the SeeBeyond CICS Listener.

Figure 7 Business Rules Request Start Transaction



Request Start Transaction

For the Request Start Transaction mode (Figure 7), the userID and password passed from the CICS eWay to the SeeBeyond CICS Listener must be defined for the z/OS security system (RACF, for example). The userID must be authorized by the z/OS security system to start CICS transaction “TRN1” inside of CICS. The default value for “TRN1” is set in the properties file of the CICS eWay, and can be overridden in the Collaboration for each request sent into the SeeBeyond CICS listener.

Security Considerations for IBM CICS Transaction Gateway

For information on CICS Transaction Gateway 5.1 security validation refer to the following:

- IBM documentation *CICS Transaction Gateway z/OS Administration* or the *CICS Transaction Gateway V5.1 Administration Guide* for your specific operating system, available at:
 - <http://www-306.ibm.com/software/htp/cics/library/cicstgv5.html>
- *Readme.txt* for CTG 5.1 provided on the CTG 5.1 installation CD_ROM
- APAR OW55570 (for RACF)

Installing the CICS eWay

This chapter contains installation information for the CICS eWay.

In addition, this chapter also includes information for installing and configuring the **SeeBeyond CICS Listener** and **IBM CICS Transaction Gateway**. One of these must be installed and configured as the underlying connection transport for accessing z/OS CICS transactions.

Chapter Topics

- [Supported Operating System](#) on page 13
- [System Requirements](#) on page 14
- [Installing the CICS eWay](#) on page 15
- [SeeBeyond CICS Listener Installation for z/OS](#) on page 16
- [SeeBeyond CICS Listener Configuration Maintenance for z/OS](#) on page 20
- [CICS Listener Considerations for Invoking DB2 Applications](#) on page 36
- [CICS Transaction Gateway 5.1 Installation and Configuration](#) on page 38

2.1 Supported Operating System

The CICS eWay is available on the following operating systems:

- Windows 2000, Windows XP, and Windows Server 2003
- HP-UX 11.0, 11i (PA-RISC), and 11i V2 (11.23)
- IBM AIX 5.1L and 5.2
- Sun Solaris 8 and 9

Also Supported

The CICS eWay also supports CICS Transaction Server residing on the following operating systems:

- OS/390 V2R10 or z/OS V1.2 and above (any reference to z/OS in this document includes OS/390 V2R10)

2.2 System Requirements

The system requirements for the CICS eWay are the same as those for eGate Integrator. For information, refer to the *SeeBeyond ICAN Suite Installation Guide*. It is also helpful to review the **Readme.txt** for any additional requirements prior to installation. The **Readme.txt** is located on the installation CD-ROM.

Although the CICS eWay, the Repository, and Logical Hosts run on the platforms listed under Supported Operating Systems, the Enterprise Designer requires the Windows operating system. The Enterprise Manager can run on any platform that supports Internet Explorer 6.0.

2.3 External System Requirements

Software Requirements

- IBM CICS Transaction Server version 1.3, 2.1, or 2.2
- An underlying connection transport using either of the following
 - ♦ **IBM CICS Transaction Gateway 5.1** (see [Requirements for the CICS Server using CICS Transaction Gateway](#) on page 14)
 - ♦ **SeeBeyond CICS Listener** (see [Requirements for the CICS Server Using the SeeBeyond CICS Listener](#) on page 14)

Requirements for the CICS Server using CICS Transaction Gateway

For directions on configuring CICS Transaction Gateway 5.1 to communicate with a CICS server over a TCP/IP using TCP62, see the “Setting Up Client/Server Communications” section of the *CICS Transaction Gateway Administration* manual for your specific platform available at:

<http://www-306.ibm.com/software/htp/cics/library/cicstgv5.html>

Requirements for the CICS Server Using the SeeBeyond CICS Listener

To enable the eWay to communicate correctly with CICS using the SeeBeyond CICS Listener the following are required:

- OS/390 V2R10 or above (see [System Requirements](#) on page 14)
- Resource Access Control Facility (RACF) or an equivalent security product
- IBM CICS Transaction Server version 1.3 or 2.1
- IBM MVS TCP/IP socket runtime libraries, installed and configured for each CICS region in which the SeeBeyond CICS Listener will be run. For more information please refer to IBM's TCP/IP V3R2 for MVS: CICS TCP/IP Socket Interface Guide.
- COBOL for z/OS and Language Environments

SeeBeyond CICS Listener Requirements for Invoking DB2 Applications

There are two different techniques that can be used for invoking DB2 application programs through the SeeBeyond CICS Listener.

- RCT entry for the STCL SeeBeyond CICS Listener Transaction
- PPT entry to redirect DB2 application to another AOR (Application Owning Region)

For detailed information refer to [CICS Listener Considerations for Invoking DB2 Applications](#) on page 36.

2.4 Application Server Support

The CICS eWay in Outbound mode is supported on WebSphere™ and WebLogic™ Application Servers when using Java Collaborations only. See the eGate Integrator User's Guide for additional information regarding the running of this eWay on this Application Server.

2.5 Installing the CICS eWay

During the eGate Integrator installation process, the Enterprise Manager, a web-based application, is used to select and upload eWays (eWay.sar files) from the eGate installation CD-ROM to the Repository.

When the Repository is run on a UNIX operating system, eGate and the eWays are installed using the Enterprise Manager from a computer running Windows, connected to the Repository server.

2.5.1. Installing the CICS eWay on an eGate Supported System

The CICS eWay is installed during the installation of the eGate Integrator. The eGate installation process includes the following operations:

- Installing the eGate Repository
- Uploading products to the Repository
- Downloading components (such as eGate Enterprise Designer and Logical Host)
- Viewing product information home pages

Follow the instructions for installing the eGate Integrator in the *SeeBeyond ICAN Suite Installation Guide*, and include the following steps:

- 1 During the procedures for uploading files to the eGate Repository using the Enterprise Manager, after uploading the **eGate.sar** file, select and upload the following files:

- ♦ **CICSeWay.sar** (to install the CICS eWay)
- ♦ **CobolCopyBook.sar** (to install the Cobol Copybook Converter Wizard)
- ♦ **FileeWay.sar** (to install the File eWay, used in the sample project)
- ♦ **CICSeWayDocs.sar** (to download the CICS eWay user's guide, sample projects, and Javadoc)

2 Continue installation as instructed in the *SeeBeyond ICAN Suite Installation Guide*

2.5.2. After Installation

Once the eWay is installed and configured it must then be incorporated into a project before it can perform its intended functions. See the *eGate Integrator User's Guide* for more information on incorporating the eWay into an eGate project.

2.6 SeeBeyond CICS Listener Installation for z/OS

The following section provides directions for installing the SeeBeyond CICS Listener to an OS/390 or z/OS operating system from the installation CD-ROM or from a 3480 Tape. Directions are also included for installing CICS CEDA definitions, adding the CICS eWay load modules to the CICS DFHRPL concatenation, and using the SeeBeyond CICS Listener monitoring screen to verify that all components are properly installed and working correctly.

2.6.1. Downloading the SeeBeyond CICS Listener Zip File

The SeeBeyond CICS Listener ZIP file, **CICS_Listener.zip** can be downloaded from the ICAN Installation CD-ROMs. To download the **CICS_Listener.zip** to your computer do the following:

- 1 Locate the Products installation CD-ROM containing the **CICSeWay.sar** file.
- 2 Open the CD-ROM to view and locate the **CICSeWay.sar** file.
- 3 Right-click the **CICSeWay.sar** file, and from the shortcut menu, select the "Open With" option and select a zip program such as WinZip™.
- 4 From the opened **CICSeWay.sar** file, click on and drag the **CICS_Listener.zip** file to your Desktop. Close the zip program. The **CICS_Listener.zip** file is now available from your computer's Desktop file.

2.6.2. Installing the SeeBeyond CICS Listener from CD for z/OS

These following describes how to restore the SeeBeyond CICS Listener files from the CICS_Listener.zip file to your MVS system in a usable state. The files are packaged on MVS for transfer using the TSO transmit (XMIT) command to transmit them into a data set. This is done to turn a PDS into FB 80 files which can be sent by FTP. Files are downloaded to a computer and then compressed with PKZIP.

- 1 Download the **CICS_Listener.zip** file to your PC (see [Downloading the SeeBeyond CICS Listener Zip File](#) on page 16), and unzip the files using a zip program such as WinZip.
- 2 Create two MVS datasets to receive the files, as follows:

```
//DD1      DD DSN=USER.XMIT.CICSLOAD,
//          DISP=(NEW,CATLG,DELETE),
//          RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS,
//          SPACE=(3120,(48,5)),
//          UNIT=diskunit

//DD2      DD DSN=USER.XMIT.JCLLIB,
//          DISP=(NEW,CATLG,DELETE),
//          RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS,
//          SPACE=(3120,(30,5)),
//          UNIT=diskunit
```

- 3 Upload (FTP, IND\$FILE) the unzipped files to MVS using a binary file transfer method (no CRLF or ASCII translation).

A Restore the files to PDS by using the **Receive** command on MVS.

B Issue command: TSO RECEIVE INDATASET(uploaded.dataset)

C When prompted by the message:

```
INMR906A Enter restore parameters or 'DELETE' or 'END' +
enter:
```

```
DA(name.of.your.library) UNIT(unit) VOLUME(volume)
```

Note: *The UNIT() and VOLUME() operands are optional but shown in case your installation requires them.*

We suggest using the following names for your received datasets:

```
STC.XMIT.CICS.CICSLOAD
STC.XMIT.CICS.JCLLIB
```

These names match our documentation. Change them as your facility requires.

2.6.3. Installing the SeeBeyond CICS Listener from 3480 Tape

The SeeBeyond CICS Listener installation for OS/390 or z/OS is provided on an installation tape containing the following datasets (Table 1):

Table 1 Tape Contents

File	Dataset Name	Contents
1	TAPE.STC.RESTORE.JCL	JCL sample to load this tape to disk (RESTORE JCL).
2	TAPE.STC.CICS.JCLLIB	Installation jobs and control cards for the CICS eWay.
3	TAPE.STC.CICS.CICSLOAD	Load library that containing the load modules for the CICS eWay.

Copying the Tape Contents to Disk

- 1 Create, customize, and submit the following job to copy the RESTORE JCL to disk:

```

1 //JOBNAME JOB (SYSTEM) , 'SYSPROG' , NOTIFY=&SYSUID , REGION=6M,
2 //          CLASS=A, MSGLEVEL=(1,1) , MSGCLASS=X
3 // *
4 // * COPY SAMPLE JCL TO DISK
5 // *
6 //IEBGENER EXEC PGM=IEBGENER
7 //SYSPRINT DD  SYSOUT=*
8 //SYSUT1   DD  DSN=TAPE.STC.RESTORE.JCL,
9 //          UNIT=tapeunit,
10 //         VOL=SER=STC390,
11 //         LABEL=(1,SL,,,EXPDT=98000) ,
12 //         DISP=OLD
13 //SYSUT2   DD  DSN=customer.dataset,
14 //         SPACE=(TRK,(10,10),RLSE) ,
15 //         UNIT=diskunit,
16 //         VOL=SER=diskvol,
17 //         DCB=(LRECL=80, BLKSIZE=3120, RECFM=FB) ,
18 //         DISP=(NEW,CATLG,DELETE)
19 //SYSIN    DD  DUMMY

```

- 2 Edit the fore mentioned code for your system as follows:
 - A Replace the Jobcard in the first two lines with one that meets your installation standards.
 - B Replace **tapeunit** in line 9 with the unit name of your tape drive: for example, UNIT=TAPE.
 - C Some Tape Management systems require an **EXPDT** value (see line 11) to indicate a foreign tape.
 - D Specify a **customer.dataset** name (see line 13) that meets your installation standards.
 - E Specify a unitname for the disk (replacing **diskunit** in line 15), to meet your installation standards: for example, UNIT=3390.
 - F Specify the volume serial number of the disk (replacing **diskvol** in line 16), where the dataset is to be created.

- 3 Once the job has completed successfully, customize and submit the sample JCL in CUSTOMER.DATASET job to copy the entire contents of the installation tape to disk. Customize displayed in the first 15 lines below

```

1 //JOBNAMEx JOB (SYSTEM), 'SYSPROG', NOTIFY=&SYSUID, REGION=6M,
2 //          MSGLEVEL=(1,1), MSGCLASS=X, CLASS=A
3 // *
4 //TAPECOPY PROC PREFIX=custpref, <== CUSTOMER HIGH LEVEL QUALIFIER
5 //          TAPEUNIT=tapeunit, <== UNIT NAME FOR TAPE DRIVE
6 //          DISKUNIT=3390, <== UNIT NAME FOR DISK DRIVE
7 //          VOLSER=diskvol, <== DISK VOLUME FOR INSTALL DATASETS
8 //          LOADBLK=TRK, <== BLOCKING FACTOR FOR CICS LOAD
9 //          LOADPRI=45, <== PRIMARY ALLOCATION FOR CICS LOAD
10 //         LOADSEC=15, <== SECONDARY ALLOCATION FOR CICS LOAD
11 //         LOADDIR=10, <== DIRECTORY BLOCKS FOR CICS LOAD
12 //         JCLLBLK=TRK, <== BLOCKING FACTOR FOR JCL LIBRARY
13 //         JCLLPRI=1, <== PRIMARY ALLOCATION FOR JCL LIBRARY
14 //         JCLLSEC=1, <== SECONDARY ALLOCATION FOR JCL LIB
15 //         JCLLDIR=10 <== DIRECTORY BLOCKS FOR JCL LIBRARY:

```

- A Replace the Jobcard in the first two lines with one that meets your installation standards.
- B Change the **custpref** value (see line 4) in the TAPECOPY procedure to a high-level-qualifier for your installation datasets.
- C Replace **tapeunit** in line 4 with the unit name of your tape drive: for example, TAPEUNIT=TAPE.
- D Change the **3390** for the DISKUNIT in line 6, to meet your installation standards: for example, DISKUNIT=SYSDA.
- E Specify the volume serial number of the disk (replacing **diskvol** in line 7), where the dataset is to be created.

2.6.4. Creating the STCCLCFG File

Customize the **Jobcard** and **Dataset** names, and submit the **STCCLCFG** job found in **JCLLIB**.

2.6.5. Installing the CICS CEDA Definitions

Customize the file definition in the JCLLIB member CEDALCFG to conform to your file naming conventions, and to match the file name created in the previous step.

Customize and submit job STCLCEDA to create CICS CEDA definitions for the eWay.

Using the CICS RDO interface, install the new definitions in group STCLSN into CICS, and add the group name STCLSN, to your start-up list.

2.6.6. Adding the CICS eWay Load Modules to CICS DFHRPL Concatenation

Add the following data set to the DFHRPL concatenation under CICS:

```
// DD DSN=&PREFIX..STC.CICS.CICSLOAD, DISP=SHR
```

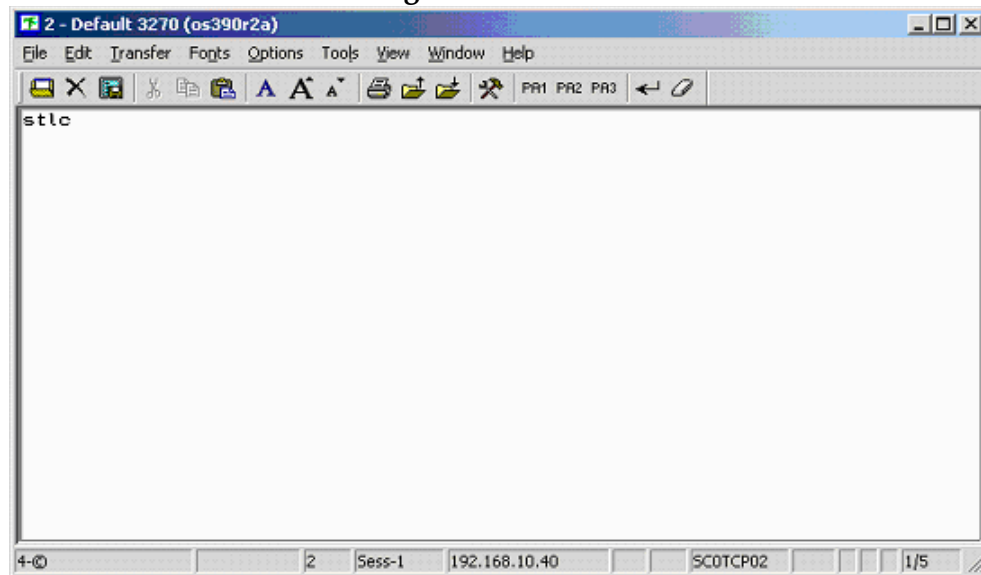
Note: Remember to re-cycle your CICS region to pick up this new library.

2.7 SeeBeyond CICS Listener Configuration Maintenance for z/OS

After the STCCLCFG Configuration File has been created and the CEDA file definition has been installed, use the SeeBeyond CICS Configuration File Maintenance screen to view and update the contents of the STCCLCFG configuration file.

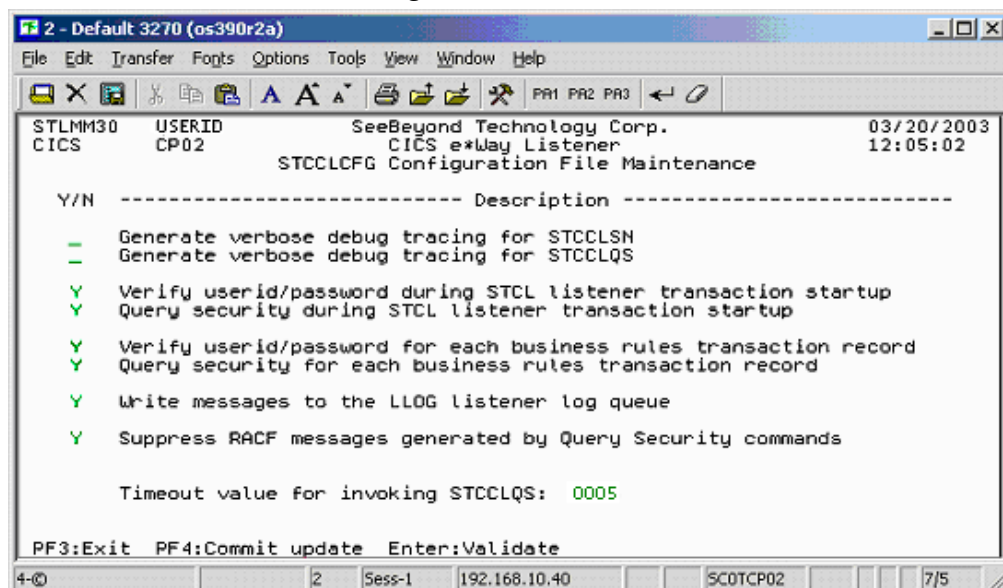
- 1 Logon to the CICS region that the SeeBeyond Listener is running in. Clear the screen, type in **stlc** as shown in Figure 8, and press **Enter**.

Figure 8



- 2 The **Configuration File Maintenance** screen appears as displayed in Figure 9.

Figure 9



3 The fields of the Listener Configuration Maintenance screen are described as follows:

- ♦ **Generate verbose debug tracing for STCCLSN:** This flag controls the logging of debug tracing messages issued by the STCCLSN SeeBeyond CICS Listener Program. These tracing messages are written to the LLOG transient data queue.

If the “Write messages to the LLOG listener log queue” field on this screen is turned off, then debug tracing will NOT be logged.

- ♦ **Y:** yes, log debug tracing messages.
 - ♦ **N or blank:** no, do not log debug tracing messages.
- ♦ **Generate verbose debug tracing for STCCLQS:** This flag controls the logging of debug tracing messages issued by the STCCLQS SeeBeyond CICS Listener Query Security Program. These tracing messages are written to the LLOG transient data queue.

If the “Write messages to the LLOG listener log queue” field on this screen is turned off, then debug tracing will NOT be logged.

- ♦ **Y:** yes, log debug tracing messages.
 - ♦ **N or blank:** no, do not log debug tracing messages.
- ♦ **Verify userid/password during STCL listener transaction startup:** This flag controls userid/password verification at the STCL SeeBeyond Listener Transaction startup. The userid and password are passed from the CICS eWay to the IBM Listener program in the user area of Connection Request Record. The IBM Listener passes this user area to the SeeBeyond CICS Listener when it spawns it as a child listener task.

If this flag is turned on, the SeeBeyond Listener Program startup logic issues a Verify Password command to check whether the userid is defined for the RACF or any equivalent security system, and if the password is valid and current.

- ♦ **Y:** yes, verify userid/password.
 - ♦ **N or blank:** no, do not verify userid/password.
- ♦ **Query Security during STCL listener transaction startup:** This flag controls Query Security processing at the STCL SeeBeyond Listener Transaction startup. The userid and password are passed from the CICS eWay to the IBM Listener program in the user area of Connection Request Record. The IBM Listener passes this user area to the SeeBeyond CICS Listener when it spawns it as a child listener task.

If this flag is turned on, the SeeBeyond Listener Program startup logic invokes the STLQ SeeBeyond Listener Query Security transaction to check if the userid is authorized to run the STCL SeeBeyond Listener transaction.

- ♦ **Y:** yes, perform Query Security processing.
- ♦ **N or blank:** no, do not Query Security processing

- ♦ **Verify userid/password for each business rules transaction record:** This flag controls userid/password verification for each Business Rules transaction sent in from the CICS eWay. The userid and password are passed in from the CICS eWay in the Application Control Record (ACR) that contains the Business Rules transaction data.

If this flag is turned on, the SeeBeyond CICS Listener program Business Rules logic issues a Verify Password command to check whether the userid is defined for the RACF or any equivalent security system, and if the password is valid and current.

- ♦ **Y:** yes, verify userid/password.
- ♦ **N or blank:** no, do not verify userid/password.
- ♦ **Query Security for each business rules transaction record:** This flag controls Query Security processing for each Business Rules transaction sent in from the CICS eWay. The userid and password are passed in from the CICS eWay in the Application Control Record (ACR) that contains the Business Rules transaction data.

If this flag is turned on, the SeeBeyond CICS Listener Program Business Rules logic invokes the STLQ SeeBeyond Listener Query Security transaction to check whether the userid is authorized to run the requested customer application program or transaction that is specified in the ACR.

- ♦ **Y:** yes, perform Query Security processing.
- ♦ **N or blank:** no, do not Query Security processing.
- ♦ **Write messages to the LLOG listener log queue:** This flag controls the logging of messages to the LLOG listener log queue.

If this flag is turned off, NO messages (normal information as well as debug tracing) will be written by either the STCCLSN (SeeBeyond CICS Listener) or STCCLQS (SeeBeyond Listener Query Security) programs.

- ♦ **Y:** yes, write log messages to the LLOG listener log queue.
- ♦ **N or blank:** no, do not write log messages to the LLOG listener log queue.
- ♦ **Suppress RACF messages generated by Query Security commands:** This flag controls RACF informational message logging for Query Security exceptions.

If this flag is turned off, then every Query Security command that results in a negative result for Control, Alter, Update, or Read will cause respective RACF information messages to be written to the system message log. This could create unnecessary, high volume logging on the system message log files.

It is recommended that this flag be turned off for debugging or low volume testing purposes only.

In a production environment, it is recommended that this flag be turned on to

suppress the RACF informational messages generated by Query Security exception conditions.

- ♦ **Y:** yes, suppress RACF messages generated by Query Security processing.
- ♦ **N or blank:** no, do not suppress (in other words, allow) RACF messages to be generated by Query Security processing.
- ♦ **Timeout value for invoking STCCLQS:** This value is used to control the amount of time the STCCLSN SeeBeyond CICS Listener program waits for the STLQ Listener Query Security Transaction to return a response. Valid values are 0000 to 9999. Since the STLQ transaction, under normal CICS processing conditions, executes within one second, it is recommended that the value be set above 1 and below 10. However, if the CICS region that the SeeBeyond CICS Listener is running in experiences occasional performance bottle necks, you may choose to set this value higher to avoid unnecessary STLQ timeout errors during peak load conditions.

This value does not affect the time STLQ takes to respond. It simply sets a timeout threshold above which the STCCLSN stops waiting for a response from STLQ and issues an error message back to the CICS eWay indicating that an STLQ timeout has occurred. For example, if this timeout value is set to 5, and STLQ executes in 1 second, then STCCLSN will “wake up” after 1 second, not the full 5 seconds.

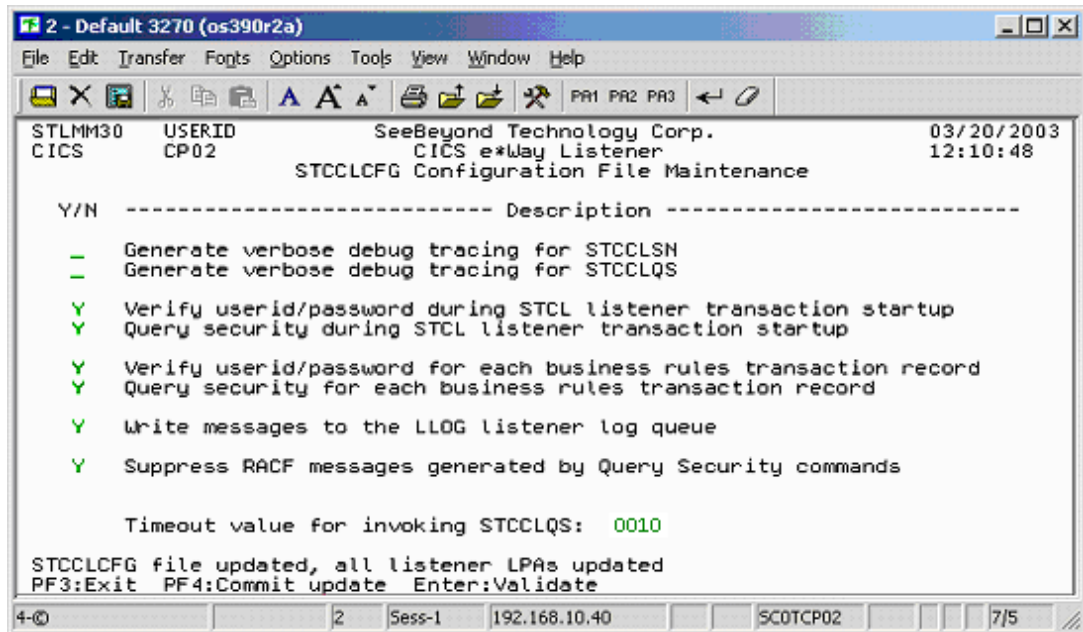
4 The pfkeys for this screen are as follows:

- ♦ **PF3:** exit.
- ♦ **PF4:** commit the updates on the screen to the STCCLCFG file and update all existing Listener Program Areas (LPA's) in CICS storage.
- ♦ **Enter:** validates the onscreen data for errors.

The user may make any necessary changes, then use the **Enter** key to validate any new values that have been entered. IF no error conditions occur for the data on the screen, then the user must press the PF4 key to commit the updates to the STCCLCFG Listener Config VSAM File. The updated screen appears as displayed in [Figure 10 on page 24](#). A message appears at the bottom of the screen that indicating the file and whether any LPA's were updated.

During commit processing, the program will also automatically propagate the new data to all existing SeeBeyond Listener Program Areas (LPA's) in the same CICS region. There is one LPA for each instance of the SeeBeyond CICS Listener in the region. The STCCLSN (SeeBeyond CICS Listener) program uses its copy of the LPA to store monitoring statistics and to retrieve data used to control the execution of the SeeBeyond CICS Listener. Live updates to the LPAs provide the capability for real time control of the configuration flags and timers for all active SeeBeyond CICS Listeners executing at that time.

Figure 10



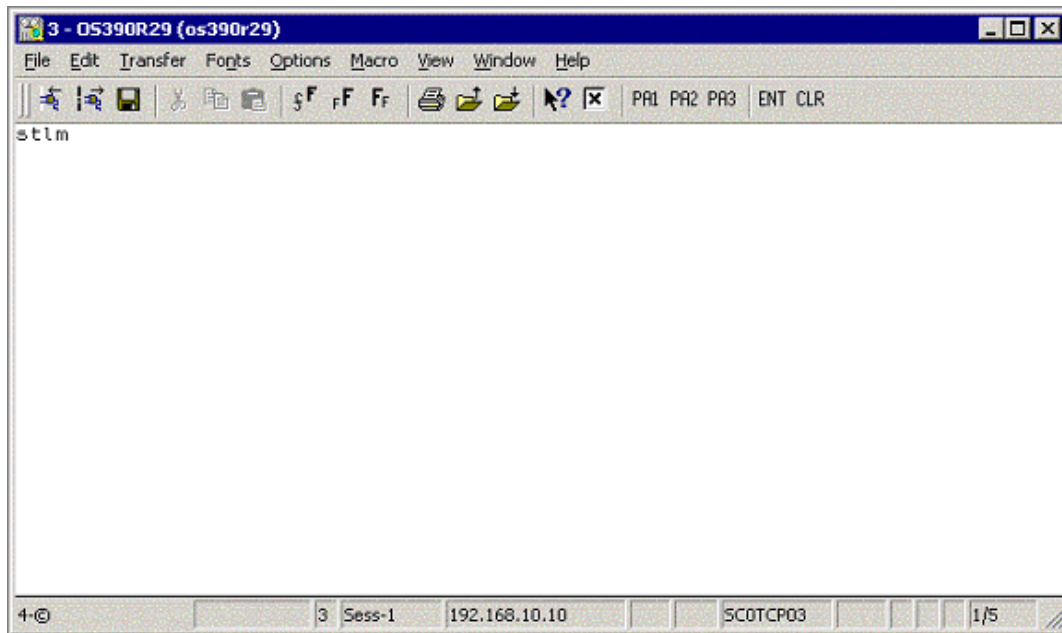
- 5 Continue making any necessary changes, committing them by pressing **PF4**. To exit the screen press **PF3**.

2.7.1. The SeeBeyond CICS Listener Monitor Screen for z/OS

After the components are installed, use the SeeBeyond CICS Listener monitoring screen to verify that all components are properly installed and working correctly.

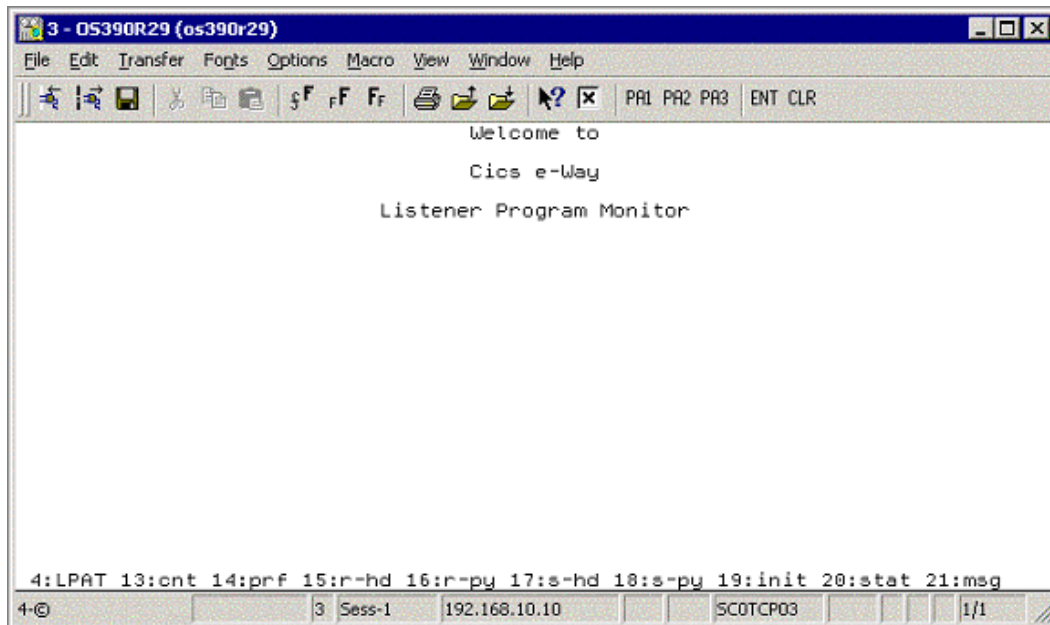
- 1 Logon to the CICS region that the SeeBeyond Listener is running in. Clear the screen, and type in **stlm** as displayed in Figure 11. Press **Enter**.

Figure 11



- 2 The menu screen appears as displayed in Figure 12.

Figure 12

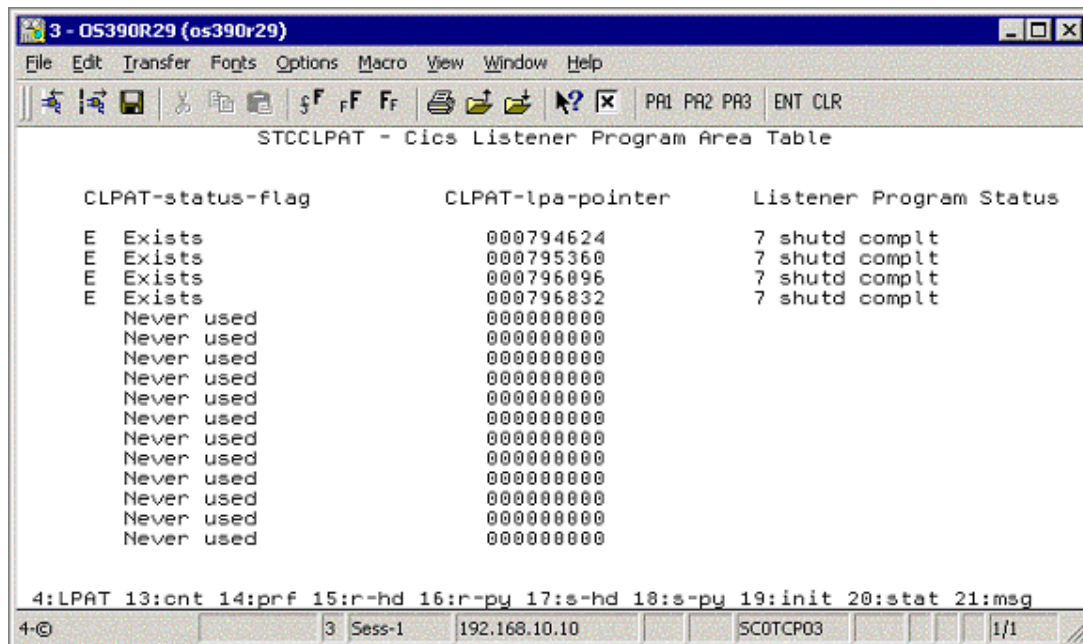


3 The **pfkeys** for this screen, and all of the monitor screens shown in this document, are as follows:

- ♦ **PF4**: displays the LPAT (Listener Program Area Table).
- ♦ **PF13**: displays the LPA counts and polling rates.
- ♦ **PF14**: displays the LPA performance statistics.
- ♦ **PF15**: displays the LPA last request header received from the CICS eWay.
- ♦ **PF16**: displays the LPA last request payload received from the CICS eWay.
- ♦ **PF17**: displays the LPA last response header sent to the CICS eWay.
- ♦ **PF18**: displays the LPA last response payload sent to the CICS eWay.
- ♦ **PF19**: displays the LPA initial record sent from the IBM Listener (EZACIC02).
- ♦ **PF20**: displays the LPA status.
- ♦ **PF21**: displays the LPA last error message sent to the CICS eWay.

Press **PF4**. The **Listener Program Area Table** screen appears as displayed in Figure 13.

Figure 13



4 This screen displays the Listener Program Area Table (LPAT). Each line contains information about a Listener Program Area (LPA). There is one LPA associated with each instance of the SeeBeyond CICS Listener Program that is running or has run in this CICS region. The fields on the screen are:

- ♦ **CLPAT-status-flag**: shows the current status of the LPAT entry as follows:
 - ♦ **E**: Exists.
 - ♦ **N**: Never used.

- ♦ C: Corrupted.
- ♦ **CLPAT-lpa-pointer**: shows the address of the most recent LPA in this entry.
- ♦ **Listener Program Status**: shows the last status of the Listener Program using this LPA.

Press PF13. The **LPA Counts and Polling Rates** screen appears as displayed in Figure 14.

Figure 14

program link	start trans	avg poll rate	non-max poll	highest poll	receive delay cnt	listener delay cnt
373	0	1.0	2	2	0	0
900	0	1.0	2	2	0	0
1	0	1.0	1	1	0	0
1	0	1.0	1	1	0	0

4:LPAT 13:cnt 14:prf 15:r-hd 16:r-py 17:s-hd 18:s-py 19:init 20:stat 21:msg

4-C 3 Sess-1 192.168.10.10 SCOTCP03 1/1

- 5 This screen displays counts and polling rates information for each LPA. Each line shows one LPA (one for each instance of the SeeBeyond Listener Program).

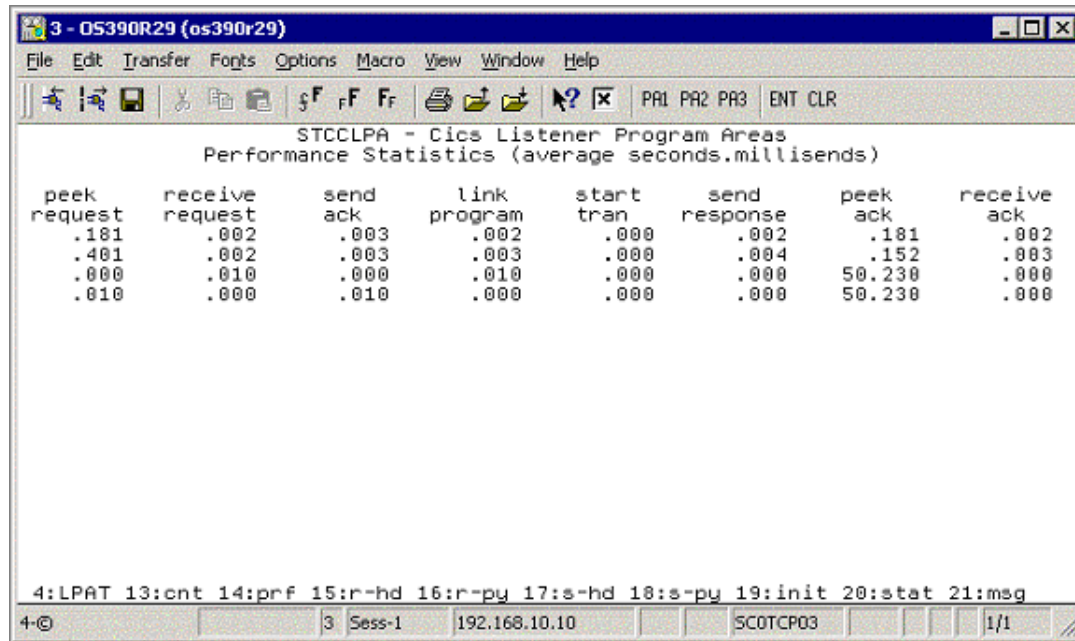
The fields on the screen are as follows:

- ♦ **program link**: the number of program links that have been requested.
- ♦ **start trans**: the number of transaction starts that have been requested.
- ♦ **avg poll rate**: the average number of receives per polling cycle to satisfy a full message receive from the CICS eWay.
- ♦ **non-max cnt**: the 'high water mark' of receives within a polling cycle without reaching the polling rate (polling rate is sent in from the CICS eWay when it initializes connection with the CICS Listener program).
- ♦ **highest poll**: the 'high water mark' of receives within a polling cycle that can include reaching the maximum allowed polling rate.
- ♦ **receive delay cnt**: the number of one second delays that occurred due to the Comm Timeout limit being reached (Comm. Timeout is sent in from the CICS eWay when it initializes connection with the CICS Listener program).

- ♦ **listener delay cnt:** number of one second delays that occur due to the Listener Timeout limit being reached (Listener Timeout is sent in from the CICS eWay when it initializes connection with the CICS Listener program).

Press **PF14**. The **LPA Performance Statistics** screen appears as displayed in Figure 15.

Figure 15



- 6 This screen (Figure 15) displays the performance statistics for each LPA. Each line shows one LPA (one for each instance of the SeeBeyond Listener Program).

All times shown are in sss.mmm format (sss = seconds, mmm=milliseconds). The fields on the screen are as follows:

- ♦ **peek request:** the average time spent peeking for the next incoming application request message.
- ♦ **receive request:** the average time spent receiving the next incoming application request message.
- ♦ **send ack:** the average time spent sending an ack for an incoming application request message.
- ♦ **link program:** the average time spent link to requested application programs.
- ♦ **start tran:** the average time spent starting requested application transactions.
- ♦ **send response:** the average time spent sending the application response (including the updated COMMAREA for program links) back to the eWay.
- ♦ **peek ack:** the average time spent peeking for the incoming ack to the preceding send response.
- ♦ **receive ack:** the average time spent receiving the incoming ack to the preceding send response.

Press PF15. The **Last Application Request Header Received** screen appears as displayed in Figure 16.

Figure 16

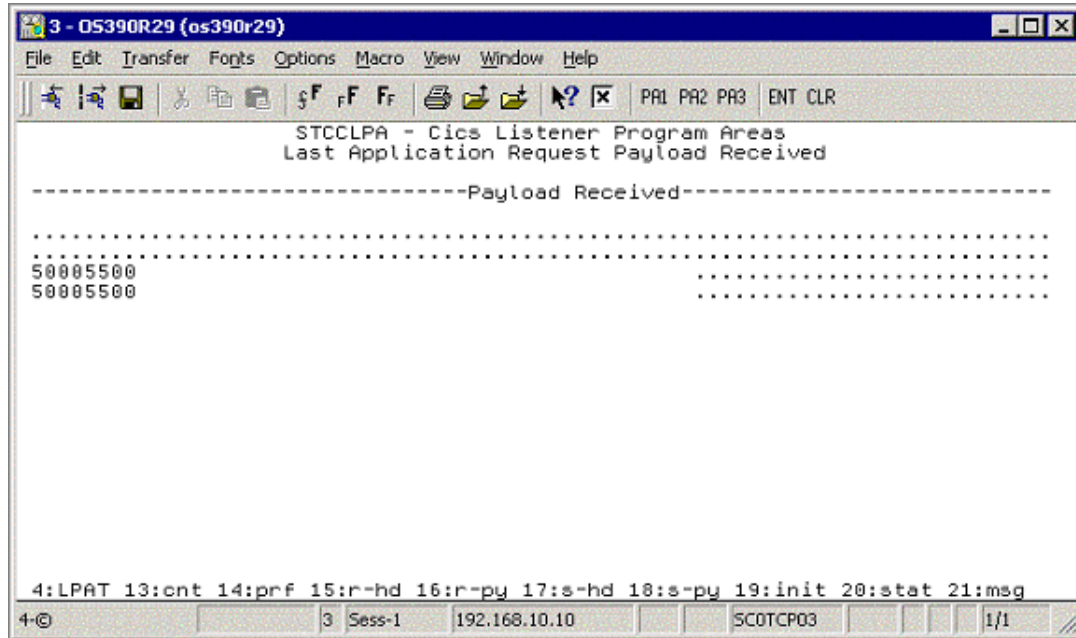
message length	program or tran	appl timeout	request code	response code	pad char	commarea length	payload length
113	QAN3GLR1	0	9000	0000	40	0	0
113	QAN3GLR1	0	9000	0000	40	0	0
163	BOB3GLR1	0	0010	0000	40	50	50
163	BOB3GLR1	0	0010	0000	40	50	50

7 This screen (Figure 16) displays the last application request header received from the CICS eWay for each LPA. Each line shows one LPA (one for each instance of the SeeBeyond Listener Program). The fields on the screen are as follows:

- ♦ **message length:** the entire length of the incoming message including the header and payload.
- ♦ **program or tran:** the requested application program or transaction.
- ♦ **appl timeout:** (reserved for future development).
- ♦ **request code:** which action is being requested.
 - ♦ 0010 = link to application program.
 - ♦ 0020 = start application transaction.
 - ♦ 0111 = ack for a link response.
 - ♦ 0121 = ack for a start response.
 - ♦ 9000 = shutdown the Listener program.
- ♦ **response code:** always set to zero from the CICS eWay.
- ♦ **pad char:** padding character for the COMMAREA if the payload length is less than the COMMAREA length.
- ♦ **commarea length:** the length of the data to pass to the application program (link) or the application transaction (start).
- ♦ **payload length:** the length of the payload portion of the incoming message.

Press **PF16**. The **Last Application Request Payload Received** screen appears as displayed in Figure 17.

Figure 17



- This screen (Figure 17) displays the last application request payload received from the CICS eWay for each LPA. Each line shows one LPA (one for each instance of the SeeBeyond Listener Program). There is one field on the screen:

- ♦ **Payload Received:** the incoming payload from the CICS eWay.

Press **PF17**. The **Last Application Response Header** screen appears as displayed in Figure 18.

Figure 18

message length	program or tran	appl timeout	request code	response code	pad char	commarea length	payload length
113	QAN3GLR1	0	9001	0000	40	0	0
113	QAN3GLR1	0	9001	0000	40	0	0
113	BOB3GLR1	0	9001	0000	40	0	0
113	BOB3GLR1	0	9001	0000	40	0	0

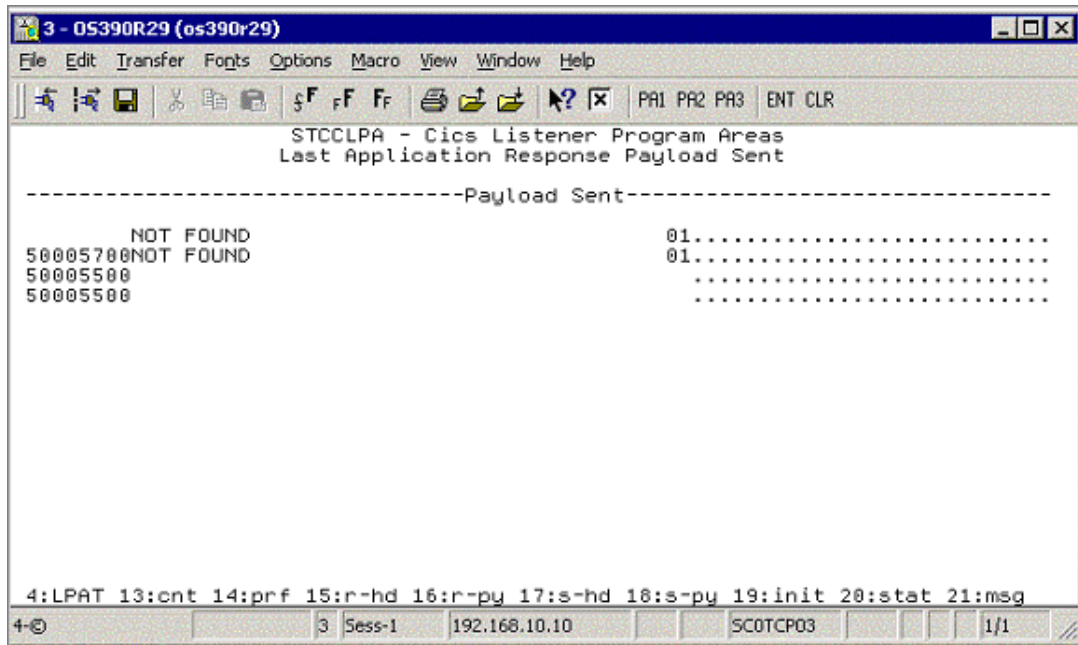
9 The **Last Application Response Header** screen (Figure 18 on page 31) displays the last response header sent to the CICS eWay for each LPA. Each line shows one LPA (one for each instance of the SeeBeyond Listener Program). The fields on the screen are as follows:

- ♦ **message length:** the entire length of the outgoing message including the header and payload.
- ♦ **program or tran:** the application program or transaction that was executed or started.
- ♦ **appl timeout:** (reserved for future development).
- ♦ **request code:** what action is being requested.
 - ♦ 0001 = ack the init request.
 - ♦ 0011 = ack for a link program request.
 - ♦ 0021 = ack for a start trans response.
 - ♦ 0110 = response from a linked application program (including COMMAREA).
 - ♦ 0120 = response from starting an application transaction (no COMMAREA included).
 - ♦ 9001 = ack for a shutdown request.
- ♦ **response code:** response code indicating what occurred while processing the previous incoming request from the eWay.
 - ♦ 0000 = ok.
 - ♦ 0011 = bad startcode during initialization.

- ♦ 0012 = bad password on init record.
- ♦ 0013 = bad comm timeout on init record.
- ♦ 0014 = bad appl timeout on init record.
- ♦ 0015 = bad Listener timeout on init record.
- ♦ 0016 = error reading the init record from transient data queue.
- ♦ 0017 = error retrieving the init record from the start data.
- ♦ 0018 = error while setting blocking mode.
- ♦ 0051 = bad payload length.
- ♦ 0052 = error while linking to a program.
- ♦ 0053 = error while starting a transaction.
- ♦ 0054 = bad request code.
- ♦ 0055 = bad response code.
- ♦ 0056 = partial message received.
- ♦ 0057 = bad program name.
- ♦ 0058 = bad message length.
- ♦ 0059 = bad commarea length.
- ♦ **pad char**: padding char that was sent in from the CICS eWay.
- ♦ **commarea length**: COMMAREA length that was sent in from the CICS eWay.
- ♦ **payload length**: payload length that was sent in from the CICS eWay (the Listener sets this to zero if this is a response to a start transaction request).

Press **PF18**. The **Last Application Response Payload Sent** screen appears as displayed in Figure 19.

Figure 19

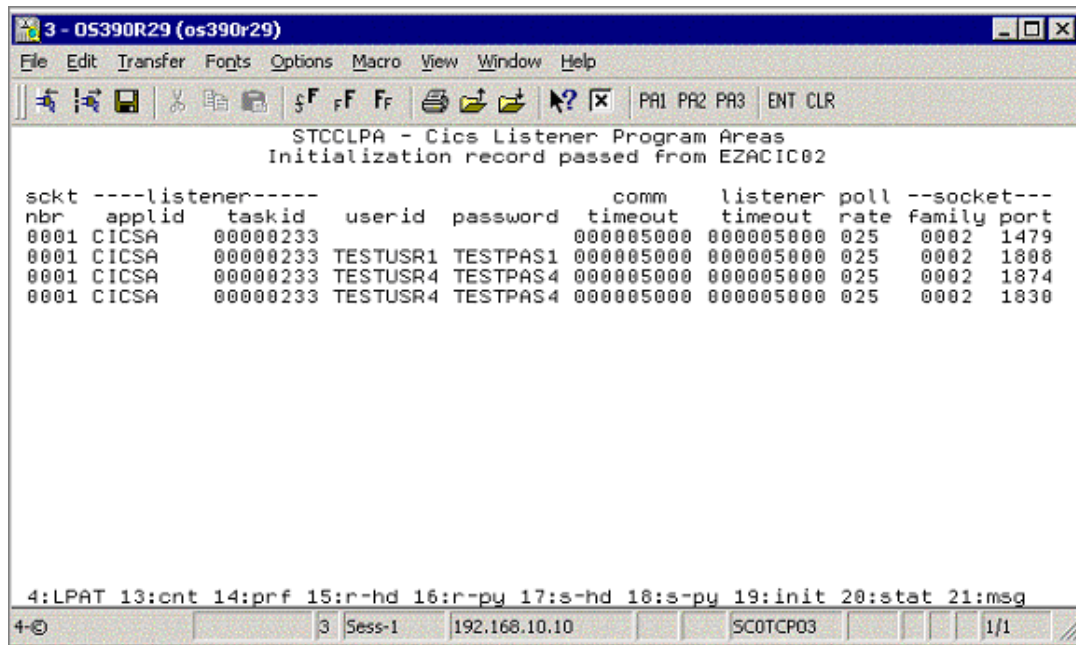


- 10 This screen (Figure 19) displays the last application response payload sent to the CICS eWay for each LPA. Each line shows one LPA (one for each instance of the SeeBeyond Listener Program). There is one field on the screen:

- ♦ Payload sent: the outgoing payload being sent to the CICS eWay.

Press PF19. The **Initialization Record Passed from EZACIC02** screen appears as displayed in [Figure 20 on page 34](#).

Figure 20

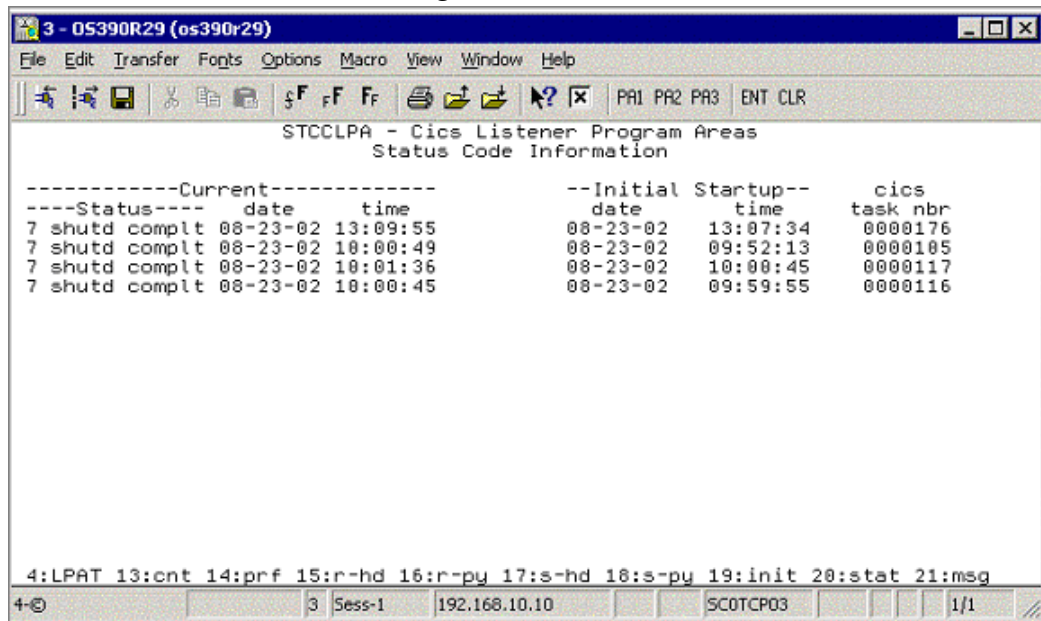


11 This screen (Figure 20) displays the init record that was passed to the SeeBeyond Listener program from the IBM Listener program (ezacic02) for each LPA. Each line shows one LPA (one for each instance of the SeeBeyond Listener program). The fields on the screen are as follows:

- ◆ **sckt nbr:** the socket number that is passed to the SeeBeyond Listener.
- ◆ **Listener applid:** the applid of the CICS region that the IBM Listener is executing in that started this instance of the SeeBeyond Listener.
- ◆ **Listener tasked:** the CICS taskid for the IBM Listener program that started this instance of the SeeBeyond Listener.
- ◆ **userid:** userid passed in from the CICS eWay.
- ◆ **password:** user password passed in from the CICS eWay.
- ◆ **comm. Timeout:** this timeout value is the threshold limit for waiting for all the bytes of an incoming message.
- ◆ **Listener timeout:** this timeout value is the threshold limit for waiting for a new incoming message.
- ◆ **poll rate:** how many times the SeeBeyond Listener will perform a receive loop to satisfy a complete message receive, after which it will wait one second before trying again.
- ◆ **socket family:** the TCP/IP family to which this socket belongs.
- ◆ **socket port:** the port this instance of the SeeBeyond Listener is using.

Press PF20. The **Status Code Information** screen appears as displayed in [Figure 21 on page 35](#).

Figure 21

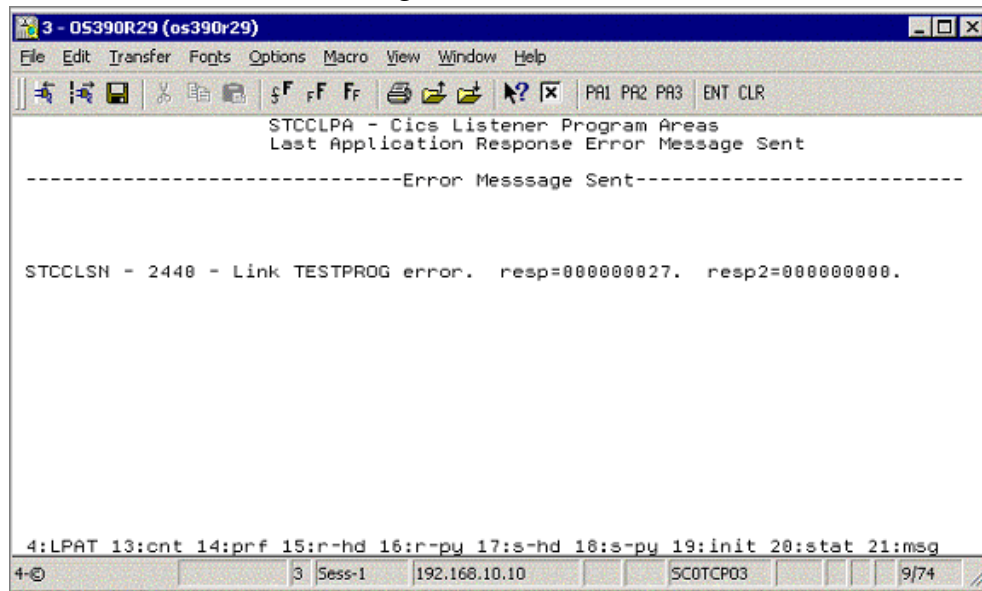


12 This screen (Figure 21) shows the status information for each LPA. Each line shows one LPA (one for each instance of the SeeBeyond Listener program). The fields on the screen are as follows:

- ♦ **Current status:** the current status of an active SeeBeyond Listener program, or the last known status of a previously executing SeeBeyond Listener program.
 - ♦ C = LPA is initialized.
 - ♦ E = about to get ezacic02 data.
 - ♦ G = about to take socket from IBM Listener program.
 - ♦ I = about to set mode to blocking.
 - ♦ K = about to send ack for init request.
 - ♦ M = peeking for length of next incoming request.
 - ♦ O = peeking for entire incoming request.
 - ♦ Q = receiving full incoming request message.
 - ♦ S = sending ack for application request.
 - ♦ U = linking to requested application program.
 - ♦ W = starting a request application transaction.
 - ♦ Y = sending a application response.
 - ♦ 0 = peeking for application response ack.
 - ♦ 2 = receiving an application response ack.
 - ♦ 4 = a shutdown request has been received.
 - ♦ 6 = sending an ack for the shutdown request.

- ♦ 7 = shutdown is complete, this LPA is now available for reuse.
 - ♦ 8 = SeeBeyond Listener program appended.
 - ♦ **Current Status date:** date for this status.
 - ♦ **Current Status time:** time for this status.
 - ♦ **Initial Startup date:** date this instance of the SeeBeyond Listener was started.
 - ♦ **Initial Startup time:** time this instance of the SeeBeyond Listener was started.
 - ♦ **cics task nbr:** the CICS eibtaskn for this instance of the SeeBeyond Listener.
- Press **PF21**. The **Error Message Sent** screen appears as displayed in Figure 22.

Figure 22



- 13 This screen (Figure 22) shows the last error message sent to the CICS eWay for each LPA. Each line shows one LPA (one for each instance of the SeeBeyond Listener program). There is one field on the screen:

Error Message Sent: the last error message sent to the CICS eWay for this instance of the SeeBeyond Listener program.

2.8 CICS Listener Considerations for Invoking DB2 Applications

One of the following two techniques are required when using the CICS eWay to run CICS applications that invoke a DB2 table:

- RCT entry for the STCL SeeBeyond CICS Listener Transaction
- PPT entry to redirect DB2 application to another AOR

RCT entry for the STCL SeeBeyond CICS Listener Transaction

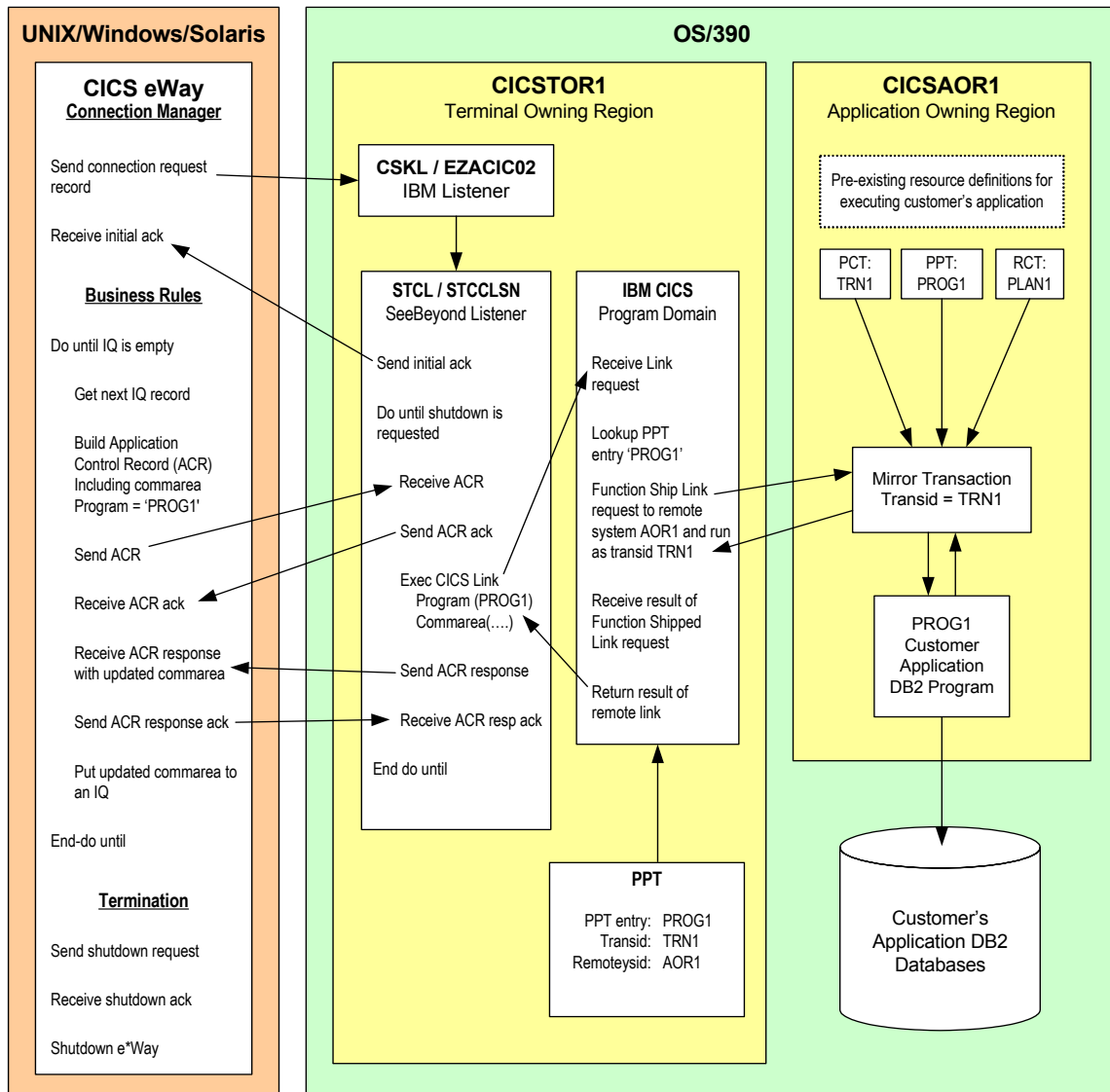
To run DB2 application programs under the STCL SeeBeyond CICS Listener transid, an RCT entry for the STCL transaction must be created using an RCT macro definition similar to the one shown below or by using an equivalent RDO entry.

```
DSNCRCT TYPE=ENTRY, TXID=STCL, THRDM=0, THRDA=0, THRDS=0,
        TWAIT=POOL, PLAN=HD45LE06, AUTH=(SIGNID, *, *)
```

PPT entry to redirect DB2 application to another AOR

To redirect DB2 application programs to another AOR, the PPT entry for the requested DB2 application program in the TOR (where the STCL SeeBeyond CICS Listener transaction is running) must be defined to run the DB2 application program in a different region (usually an AOR) and under a specific PCT in that AOR. Refer to Figure 23 for a detailed flow diagram showing the use of the PPT and PCT.

Figure 23 SeeBeyond CICS Listener invoking DB2 programs



2.9 CICS Transaction Gateway 5.1 Installation and Configuration

For installation and configuration information regarding IBM's CICS Transaction Gateway 5.1, consult the latest CTG 5.1 documentation at IBM's CICS Library Website:

<http://www-3.ibm.com/software/ts/cics/library/cicstgv5.html>

This documentation should also be consulted for information on configuring CTG for use with RACF and the latest APARs.

IBM CICS Transaction Gateway properties are set using the CTG Configuration Tool. The Configuration Tool is located under the CICS Transaction Gateway program menu.

Configuring the CICS eWay

This chapter describes how to create and configure the CICS eWay properties.

Chapter Topics

- [Creating and Configuring the CICS eWay](#) on page 39
- [Using the Properties Sheet](#) on page 41
- [CICS eWay Connectivity Map Configuration Properties](#) on page 42
- [CICS eWay Environment Explorer Configuration Properties](#) on page 46

3.1 Creating and Configuring the CICS eWay

All eWays contain a set of parameters with properties unique to that eWay type. After the eWays are established and a CICS External System is created in the project's Environment, the eWay parameters can be modified for your specific system. The CICS eWay parameters are modified from these locations:

- From the **Connectivity Map**. These parameters most commonly apply to a specific component eWay, and may vary from other eWays (of the same type) in the project.
- From the **Environment Explorer tree**. These parameters are commonly global, applying to all eWays (of the same type) in the project. The saved properties are shared by all eWays in the CICS External System window.
- CICS eWay properties may also be set from the **Collaboration Editor (Java)** or **eInsight Business Rules Designer**, in which case they will override the corresponding properties in the eWay's configuration file. Any properties that are not overridden retain their saved settings. Thieves

3.1.1 Selecting CICS as the External Application

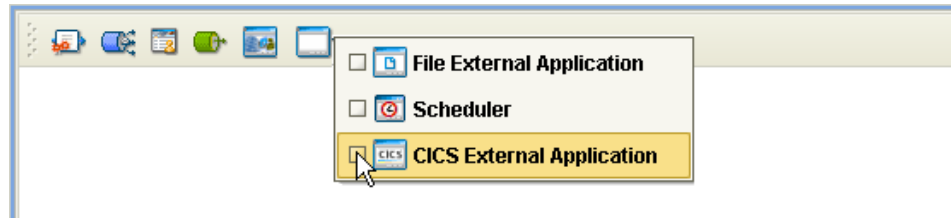
To create a CICS eWay you must first create a CICS External Application in your Connectivity Map. CICS eWays are located between a CICS External Application and a Service. Services are containers for Collaborations, Business Processes, eTL processes, and so forth.

To create the CICS External Application

- 1 From the Connectivity Map toolbar, click the **External Applications** icon.

- 2 Select the **CICS External Application** from the menu (see [Figure 24 on page 40](#)). The selected CICS External Application icon appears on the Connectivity Map toolbar.

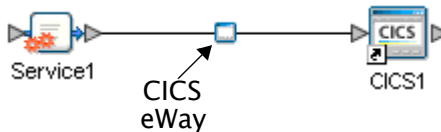
Figure 24 External Applications Selection Menu



- 3 Drag the new **CICS External Application** from the toolbar onto the Connectivity Map canvas. This represents an external CICS system.

From the Connectivity Map, you can associate (bind) the External Application with the Service to establish an eWay (see [Figure 25](#)).

Figure 25 eWay Location.



When CICS is selected as the External Application, it automatically applies the default CICS eWay properties, provided by the OTD, to the eWay that connects it to the Service. These properties can then be or modified for your specific system using the **Properties Sheet**.

3.1.2 Modifying the CICS eWay Properties

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

Modifying the CICS eWay (Connectivity Map) Properties

- 1 From the Connectivity Map, double click the eWay icon, located in the link between the associated External Application and the Service.
- 2 The eWay **Properties Sheet** opens with the CICS eWay Connectivity Map properties. Make any necessary modifications and click **OK** to save the settings.

Modifying the CICS eWay (Environment Explorer) Properties

- 1 From the Environment Explorer tree, right-click the CICS External System. Select **Properties** from the shortcut menu. The **Properties Sheet** opens with the CICS eWay Environment properties.
- 2 Make any necessary modifications to the Environment properties of the CICS eWays, and click **OK** to save the settings.

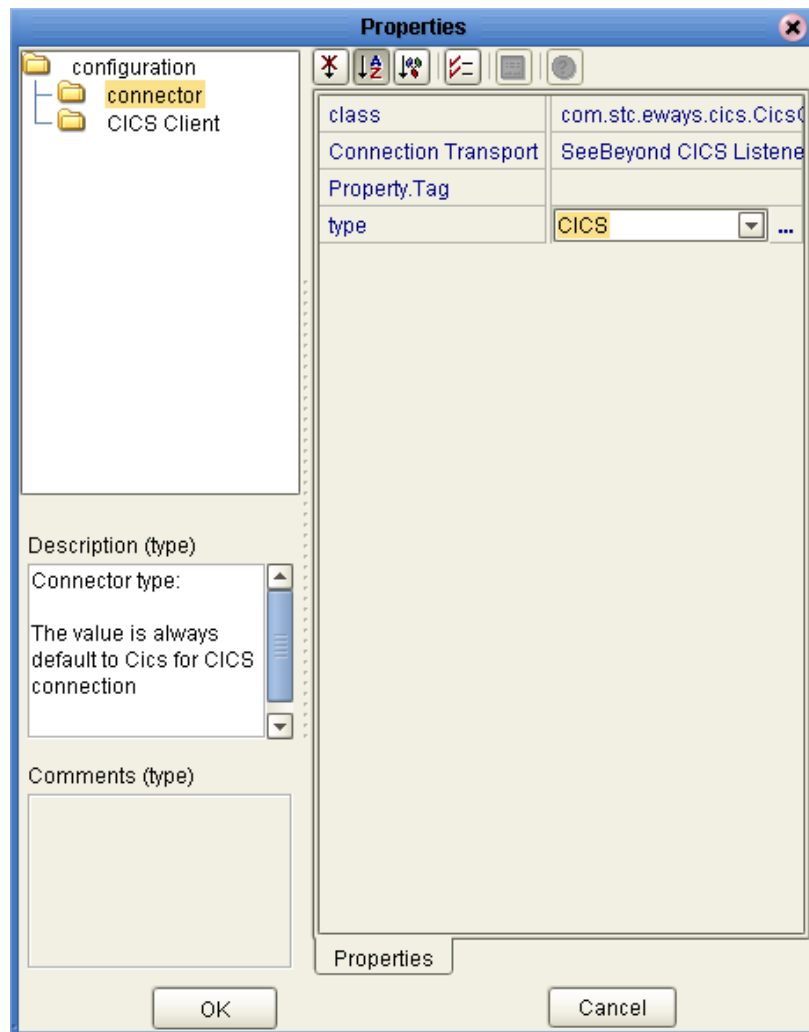
3.1.3. Using the Properties Sheet

Modifications to the eWay configuration properties are made from the CICS eWay Properties Sheet.

To modify the default eWay configuration properties

- 1 Open the Properties Sheet to the CICS eWay default properties. An eWay has two different sets of parameters: those specific to that particular eWay (accessed from the Connectivity Map), and those that are common to all eWays of this type (accessed from the Environment Explorer tree).
- 2 From the upper-right pane of the Properties Sheet, select a subdirectory of the configuration directory. The parameters contained in that subdirectory are now displayed in the Properties pane of the Properties Sheet. For example, clicking on the **connector** subdirectory displays the editable parameters in the right pane, as shown in Figure 26.

Figure 26 Properties Sheet -- CICS Properties



- 3 Click on any property field to make it editable. For example, click on the **class** parameter to edit the class value. If a parameter's value is true/false or multiple choice, the field reveals a submenu of property options.

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

- 4 A description of each parameter is displayed in the **Description** pane when that parameter is selected, providing an explanation of any required settings or options.
- 5 The **Comments** pane provides an area for recording notes and information regarding the currently selected parameter. This is saved for future referral.
- 6 After modifying the configuration properties, click **OK** to close the Properties Sheet and save the changes.

3.2 CICS eWay Connectivity Map Configuration Properties

The CICS eWay configuration parameters, accessed from the Connectivity Map, are organized into the following sections:

- **Connector** on page 42
- **CICS Client** on page 43

3.2.1. Connector

This section contains a set of top level parameters:

- **Class** on page 42
- **Connection Transport** on page 43
- **Property.Tag** on page 43
- **Type** on page 43

Class

Description

Specifies the class name of the CICS Client connector object.

Required Value

The valid (class) package name for the CICS Client connector object. The default is `com.stc.eways.cics.CicsClientConnector`.

Connection Transport

Description

Specifies the underlying connection transport used by the CICS eWay to send requests and get responses from a CICS region.

Required Value

Enter the name of the selected underlying connection transport as follows:

- **SeeBeyond CICS Listener**
- **Transaction Gateway** (specifies the IBM CICS Transaction Gateway 5.1)

SeeBeyond CICS Listener is the configured default.

Property.Tag

Description

Specifies the data source identity. This parameter is required by the current EBobConnectorFactory.

Required Value

The valid data source package name.

Type

Description

Specifies the connector type.

Required Value

CICS. The value always defaults to **CICS** for CICS connections.

3.2.2. CICS Client

This section contains the following parameters for CICS Client setup:

- **CICS Program** on page 44
- **CICS TransId** on page 44
- **COMMAREA Length** on page 44
- **ECI call type** on page 44
- **ECI extend mode** on page 44
- **Encoding** on page 45

CICS Program

Description

Specifies the CICS program to be run on the server. Maximum length is eight characters.

Required Value

A valid CICS program name, eight characters or less.

CICS TransId

Description

Specifies the CICS TransId to be run on the server. Maximum length is four characters.

Required Value

A valid CICS TransId, four characters or less.

COMMAREA Length

Description

Specifies the length (in bytes) of the communication area (COMMAREA) passed to the ECI.

Required Value

An integer in the range of 1 to 32659. The configured default is 1000.

Note: When using the CICS Transaction Gateway transport, data sent to CICS must be padded with spaces, if necessary, to match the full size of the commarea.

ECI call type

Description

Specifies whether the ECI call type is Synchronous. Synchronous calls wait for the transaction to complete, then return the contents of the COMMAREA. Only Synchronous calls are supported

Required Value

Synchronous is the configured default.

ECI extend mode

Description

Specifies whether a logical unit of work is terminated at the end of a call.

Required Value

Yes or **No**. Yes indicates that the work unit is terminated at the end of a call. No is the configured default.

Encoding

Description

Specifies the canonical name for the encoding set.

Required Value

The canonical name for any encoding set supported by Sun's Java Runtime Environment 1.1.8 (contained in rt.jar and i18n.jar). Examples are ASCII and Cp500 (EBCDIC). When running the CICS eWay on z/OS platforms, encoding should be set to "ISO-8859-1".

3.3 CICS eWay Environment Explorer Configuration Properties

The CICS eWay configuration parameters, accessed from the Environment Explorer tree, are organized into the following sections:

- [SeeBeyond CICS Listener](#) on page 46
- [CICS Gateway](#) on page 49
- [Tracing](#) on page 51
- [CICS Client](#) on page 52

3.3.1. SeeBeyond CICS Listener

These parameters are specific to the SeeBeyond CICS Listener. This section contains a set of top level parameters:

- [COMMAREA Padding Character](#) on page 46
- [Host](#) on page 47
- [KeepAlive](#) on page 47
- [Listener Timeout](#) on page 47
- [NoDelay](#) on page 47
- [Polling Rate](#) on page 48
- [Port](#) on page 48
- [ReceiveBufSize](#) on page 48
- [SeeBeyond CICS Listener TransId](#) on page 48
- [SendBufSize](#) on page 48
- [Start Delay](#) on page 49
- [Start Type](#) on page 49
- [TP Timeout](#) on page 49
- [Transport Timeout](#) on page 49

COMMAREA Padding Character

Description

Specifies the EBCDIC code for the character used by the SBYND listener to pad the COMMAREA at the CICS server when the actual length of the payload in the COMMAREA is shorter than the length given by CommAreaLength. The default value is hexadecimal 40 - EBCDIC space.

Required Value

- A character value coded in Hexadecimal. For example: 40 for Blanks, 00 for Low Values, FF for High Values, and so forth.

Host

Description

Specifies the name of the mainframe host with which to connect. This is always CICS.

Required Value

CICS. The value always defaults to CICS for CICS connections. The default is CICS.

KeepAlive

Description

Specifies whether to enable socket keep-alive checking. A setting of TRUE enables an implementation specific time period when a probe is sent to the peer host. The purpose of this option is to detect if the peer host has crashed.

One of three responses is expected:

1. The peer responds with the expected ACK. The application is not notified (since everything is OK). TCP will send another probe following another 2 hours of inactivity.
2. The peer responds with an RST, which tells the local TCP that the peer host has crashed and rebooted. The socket is closed.
3. There is no response from the peer. The socket is closed.

Required Value

Enter TRUE or FALSE. TRUE is the default.

Listener Timeout

Description

Specifies the estimated amount of time (in milliseconds) for the SeeBeyond CICS Listener to wait for the next incoming transaction program request from the CICS eWay.

Required Value

An integer between 1 and 864000 representing milliseconds (for example, 120000 milliseconds equals 2 minutes). The default value is 5000.

NoDelay

Description

Specifies whether the system can delay connections or requests. Generally, **NoDelay/True** is necessary for high-volume and/or critical transactions. In cases of low-volume and/or noncritical transactions, you can use **NoDelay/False**. Required Values

Required Value

Enter **TRUE** or **FALSE**. TRUE is the default.

Polling Rate

Description

Specifies the polling rate. This is the number of times the SeeBeyond CICS Listener will query the current TCP connection for incoming traffic before issuing an EXEC CICS DELAY for one second.

Required Value

An integer between 1 and 255 representing. The default value is 5.

Port

Description

Specifies the TCP/IP port where the SeeBeyond CICS Listener is listening. This is the port to which the CICS eWay will connect.

Required Value

The TCP/IP port to which SeeBeyond CICS Listener is listening. The default is 3001.

ReceiveBufSize

Description

Specifies the Receive Buffer Size (in bytes) for the underlying socket, this is a hint.

Required Value

An integer between 1 and 864000 representing bytes (for example, 10240 bytes equals 10 kilobytes). The default value is 4096.

SeeBeyond CICS Listener TransId

Description

Specifies the TransId of the SeeBeyond CICS Listener on the mainframe host. This is the CICS Transaction that the SeeBeyond CICS Listener is installed under.

Required Value

The valid TransId of the SeeBeyond Cics Listener

SendBufSize

Description

Specifies the Send Buffer Size for the underlying socket.

Required Value

An integer between 1 and 864000 representing bytes (for example, 10240 bytes equals 10 kilobytes). The default value is 4096.

Start Delay

Description

Specifies the hours, minutes and seconds (interval of time) to delay starting the transaction program (TP) on the CICS server for the **IC** Start Type. This field is optional but must specify all 6 digits if used.

Required Value

A 6 digit integer. All 6 digits must be given if this is specified (for example, 000000).

Start Type

Description

Specifies the startup type. This can be either **IC** for CICS interval control or **TD** for CICS transient data. This is the CICS Startup type for the program being executed.

Required Value

Select **IC** or **TD**.

TP Timeout

Description

Specifies the amount of time the CICS eWay will wait for the SeeBeyond CICS Listener to return results for a current transaction program request.

Required Value

An integer between 1 and 864000 representing milliseconds (for example, 120000 milliseconds equals 2 minutes). The default value is 50000.

Transport Timeout

Description

Specifies the timeout used by both the local and host side for receive on the socket.

Required Value

An integer between 1 and 864000 representing milliseconds (for example, 120000 milliseconds equals 2 minutes). The default value is 5000.

3.3.2. CICS Gateway

These parameters are specific to the CICS Transaction Gateway (CTG). This section contains the following parameters for CICS Java Gateway setup:

- **Port** on page 50
- **SSL KeyRing Class** on page 50
- **SSL KeyRing Password** on page 50
- **URL** on page 50

Port

Description

Specifies the TCP/IP port with which to connect, that is, the port where CTG is running.

Required Value

An integer ranging from 1 to 864000. The default value is 2006.

SSL KeyRing Class

Description

Specifies the full classname of the SSL KeyRing class.

Required Value

The valid full classname of the SSL KeyRing class.

SSL KeyRing Password

Description

Specifies the PASSWORD for the encrypted KeyRing class.

Required Value

The valid password for the SSL KeyRing class.

URL

Description

Specifies the remote or local Gateway with which to connect.

Required Value

The valid remote or local Gateway node name or IP address.

3.3.3. Tracing

This section contains the following parameters for CICS Client setup:

- **Dump Offset** on page 51
- **Filename** on page 51
- **Level** on page 51
- **Timing** on page 52
- **Truncation Size** on page 52

Dump Offset

Description

CTG specific. Specifies the offset from which the display of any data blocks will start.

Required Values

A number ranging from **0** to **864000**, indicating the offset. If the offset is greater than the total length of data to be displayed, an offset of **0** is used.

The dump offset can also be set using the system property **gateway.T.setDumpOffset=xxx**, where **xxx** is a number.

Filename

Description

CTG specific. Specifies a file location for writing the trace output. This is an alternative to the default output on stderr. Long filenames must be surrounded by quotation marks; for example: "trace output file.log".

Required Values

A valid output file name.

Filename can also be set using the system property **gateway.T.setTFile=xxx**, where **xxx** is a filename.)

Level

Description

CTG specific. Specifies the level of trace information available.

Required Values

An integer in the range of 0 to 3, indicating the following:

- **0: None.** No CICS Java client application tracing.
- **1: Standard.** Only the first 128 bytes of any data block (for example the COMMAREA, or network flows) are displayed by default. This trace level is equivalent to the Gateway trace set by the ctgstart -trace option. (This can also be set using the system property **gateway.T.trace=on**)

- **2: Full Debug.** Traces out the whole of any data blocks by default. The trace contains more information about CICS Transaction Gateway than the standard trace level. This trace level is equivalent to the Gateway debug trace set by the `ctgstart -x` option. (This can also set using the system property `gateway.T=on`)
- **3: Exception Stacks.** Traces most Java exceptions, including exception which are expected during normal operation of the CICS Transaction Gateway. No other tracing is written. This trace level is equivalent to the Gateway stack trace set by the `ctgstart -stack` option. (This can also set using the system property `gateway.T.stack=on`)

Timing

Description

Specifies whether or not to display time-stamps in the trace.

Required Value

On or **Off**. **On** indicates that the time-stamp is displayed in the trace. The default setting is **On**.

Timing can also set using the system property `gateway.T.timing=on`.

Truncation Size

Description

CTG specific. Specifies the maximum size of any data blocks written in the trace.

Required Value

A number ranging from **0** to **864000**, indicating the maximum data block size. A value of **0** indicates that no data blocks will be written in the trace. No value (leaving the property blank) indicates that no truncation size is specified.

The truncation size can also be set using the system property `gateway.T.setTruncationSize=xxx`, where `xxx` is a number.

3.3.4. CICS Client

This section contains the following parameters for CICS Client setup:

- [CICS Password](#) on page 52
- [CICS UserId](#) on page 53

CICS Password

Description

Specifies the password for the CICS user. Maximum length is eight characters.

Required Value

- A valid password for the user ID, eight characters or less.

CICS UserId

Description

Specifies the ID of the CICS user. Maximum length is eight characters.

Required Value

A valid CICS user ID, eight characters or less.

3.4 Alerting and Logging

eGate provides an alerting and logging feature. This allows monitoring of messages, and captures any adverse messages in order of severity based on configured severity level and higher.

For information on enabling or modifying the level of logging for the various logging categories, see the *eGate Integrator System Administration Guide*.

The alerts/status notifications for the CICS eWay are currently limited to Started, Running, Stopping, and Stopped.

Using eInsight Business Processes with the CICS eWay

This chapter describes how to use the CICS eWay with the ICAN Suite's eInsight Business Process Manager and the Web Services interface.

Note: You must have the *eInsight.sar* file installed to use the Web Services interface.

Chapter Topics

- [The eInsight Engine and Components](#) on page 54
- [The CICS eWay With eInsight](#) on page 55
- [The CICS eWay eInsight Sample Project](#) on page 56

4.1 The eInsight Engine and Components

eGate components can be deployed as Activities in eInsight Business Processes. Once a component is associated with an Activity, eInsight invokes it using a Web Services interface. eGate components that can interface with eInsight in this way include the following:

- Object Type Definitions (OTDs)
- eWays
- Collaborations

Using the Enterprise Designer and eInsight, you can add an Activity to a Business Process, then associate that Activity with an eGate component, for example, an eWay. Then, when eInsight runs the Business Process, it automatically invokes that component via its Web Services interface.

See the *eInsight Business Process Manager User's Guide* for details.

4.2 The CICS eWay With eInsight

An eInsight Business Process Activity can be associated with the CICS eWay during the system design phase. To make this association, select the desired operators under the eWay in the Enterprise Explorer and drag it onto the eInsight Business Process Designer canvas.

The CICS eWay has the following operators available:

- execute
- getServerList

The operation is automatically changed to an Activity with an icon identifying the component that is the basis for the Activity. At run time, eInsight invokes each step in the order defined in the Business Process. Using eInsight's Web Services interface, the Activity in turn invokes the CICS eWay.

Dynamic Configuration

The CICS eWay web service message structure exposes properties such as **server**, **transId**, **password**, **userId**, and so forth, that correspond to eWay configuration parameters. Populating these properties dynamically at runtime from the Business Rules Designer, allows the user to override the corresponding configuration parameters.

4.3 CICS CTG Concerns

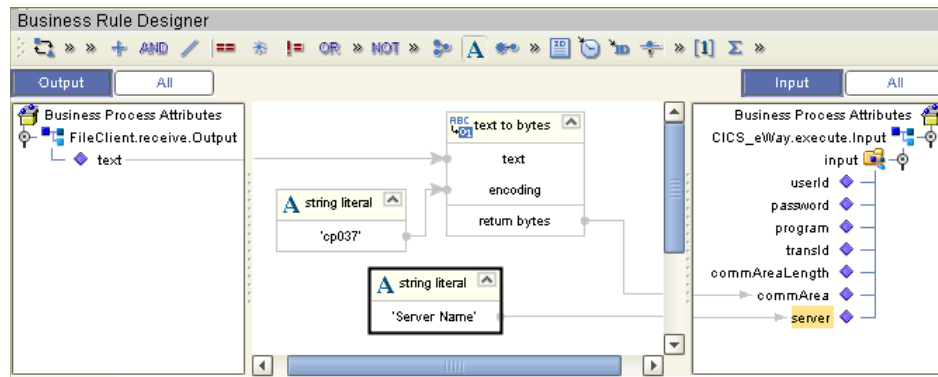
Defining More Than One Server When Using CTG

If CTG is used as the connection transport and more than one server is defined in a CICS Transaction Gateway installation, the target server name must be assigned as a literal to the server node in the Business Process.

To assign the target server, do the following:

- 1 From the eInsight Business Process Designer canvas, double-click the Business Process before the **CICS_eWay.execute** activity, to open the Business Rules Designer.
- 2 From the Business Rules Designer toolbar, select the String Literal and drag it to the Business Rules Designer canvas. Enter the target server name as the Literal Value.
- 3 Map the output node of the String Literal to **server** under **CICS_eWay.execute.Input > Input** in the Business Rule Designer's Input pane (see [Figure 27 on page 56](#)).

Figure 27 eInsight Business Process Designer - Defining the Target Server



- 4 Save changes to the Repository.

Matching the COMMAREA Size

When using the CICS Transaction Gateway transport, data sent to CICS must be padded with spaces, if necessary, to match the full size of the commarea.

4.4 The CICS eWay eInsight Sample Project

The following pages provide directions for creating a simple project that demonstrates how eInsight Business Processes are used with the CICS eWay. The same project can be downloaded from the Installation CD-ROM in a near-complete state.

For more information on creating ICAN projects see the *eInsight Business Process Manager User's Guide* and the *eGate Integrator User's Guide*.

Expanded BPEL Properties

The 5.0.3 version of the CICS eWay displays additional CICS BPEL properties in the message structure, visible from the Business Rules Designer. The packaged sample that is included on the installation CD-ROM is designed to work with eGate, versions 5.0.1 and above, and will not display these additional properties when loaded. To see these additional parameters, create the sample manually, as directed in this chapter.

4.5 Importing a Sample Project

Sample eWay projects are included as part of the installation CD-ROM package. To import a sample eWay project to the Enterprise Designer do the following:

- 1 The sample files are uploaded with the eWay's documentation .sar file and downloaded from the Enterprise Manager's Documentation tab. The **CICS_eWay_Sample.zip** file contains the various sample project zip files. Extract the samples from the Enterprise Manager to a local file.

- 2 Save all unsaved work before importing a project.
- 3 From the Enterprise Designer’s Project Explorer pane, right-click the Repository and select **Import** from the shortcut menu. The **Import Manager** appears.
- 4 Browse to the directory that contains the sample project zip file. Select the sample file (for this sample, **CICS_BP_Sample.zip**) and click **Import**. After the sample project is successfully imported, click **Close**.
- 5 Before the imported sample project can be run you must do the following:
 - ♦ Create an **Environment** (see [Creating an Environment](#) on page 65)
 - ♦ Configure the eWay properties for your specific system (see [Configuring the eWay Properties](#) on page 66)
 - ♦ Create a **Deployment Profile** (see [Creating and Activating the Deployment Profile](#) on page 68)

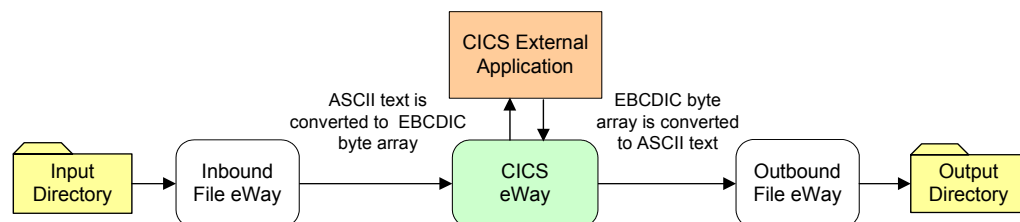
The following pages provide step by step directions for manually creating the CICS_BP_Sample project.

4.6 The CICS_BP_Sample Project Overview

The CICS eWay project, **CICS_BP_Sample**, demonstrates the following:

- The Inbound File eWay subscribes to an external directory and receives input data (ASCII text).
- The ASCII text data is converted to EBCDIC (byte array) using cp037 encoding, and published by the CICS eWay, to the CICS external system.
- Input EBCDIC data from the CICS external system, is received by the CICS eWay and converted back to String ASCII data using cp037 encoding. This data is published to the outbound File eWay.
- The outbound File eWay publishes the data to an external directory.

Figure 28 CICS_BP_Sample Project



4.7 Creating the CICS_BP_Sample Project

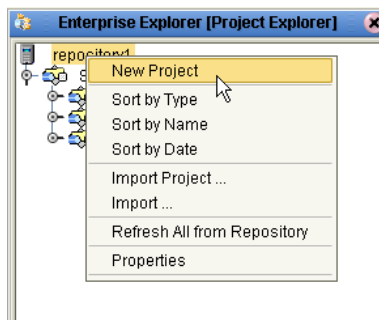
The following pages provide step by step directions for creating the CICS_BP_Sample project.

4.7.1. Creating a Project

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

- 1 Start the Enterprise Designer.
- 2 From the Project Explorer tree, right-click the Repository and select **New Project** (see Figure 29). A new project (**Project1**) appears on the Project Explorer tree.

Figure 29 Enterprise Explorer - New Project



- 3 Click twice on **Project1** and rename the project (for this sample, **CICS_BP_Sample**).

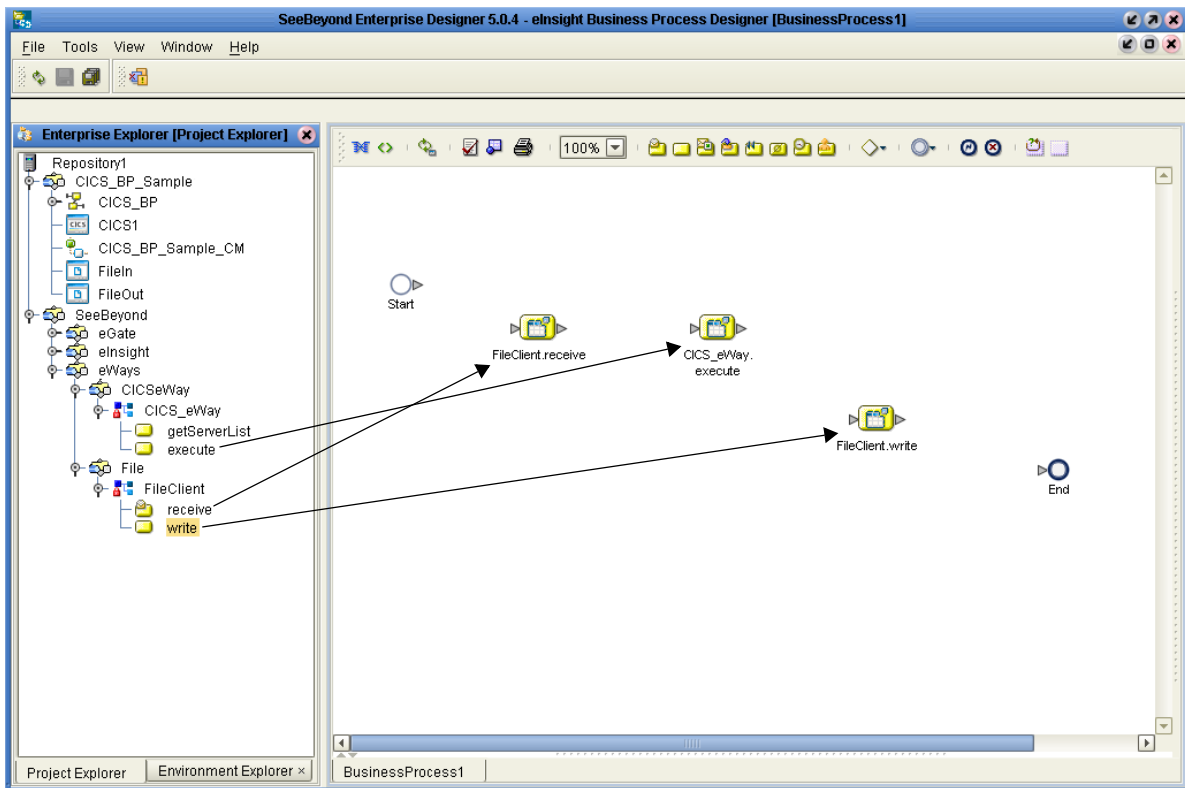
4.7.2 Creating a Business Process

To create the CICS eInsight Business Process, do the following:

Creating the Business Process Flow

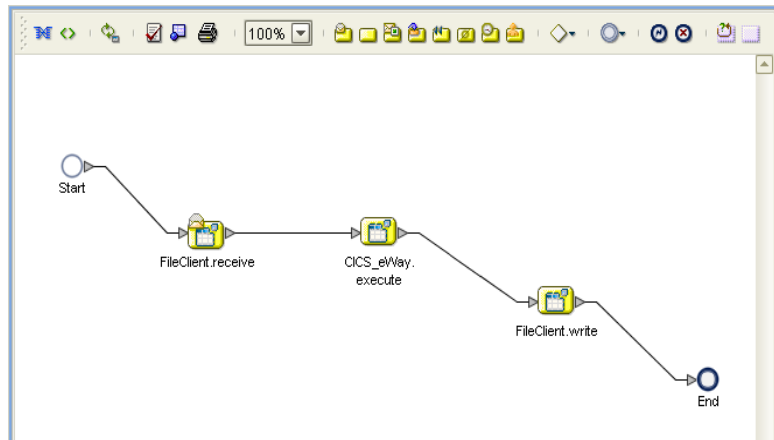
- 1 Right-click an existing project (for this sample **CICS_BP_Sample**) in the Enterprise Designer's Project Explorer, and select **New > Business Process** from the shortcut menu. The eInsight Business Process Designer appears and **BusinessProcess1** is added to the Project Explorer tree. Rename **BusinessProcess1** to **CICS_BP**.
- 2 From the Project Explorer tree, expand the SeeBeyond > eWays > CICS_eWay > CICS_eWay, and File > FileClient nodes.
- 3 Populate the eInsight Business Process Designer's modeling canvas with the following activities from the Project Explorer tree, as displayed in [Figure 30 on page 59](#):
 - ♦ **receive**, under SeeBeyond > eWays > File > FileClient
 - ♦ **execute**, under SeeBeyond > eWays > CICS_eWay > CICS_eWay
 - ♦ **write**, under SeeBeyond > eWays > File > FileClient

Figure 30 eInsight Business Process Designer - Populate the Canvas



- 4 Link the modeling elements by clicking on the element connector and dragging the cursor to the next element connector, making the following links as displayed in Figure 31.
 - Start -> FileClient.receive
 - FileClient.receive -> CICS_eWay.execute
 - CICS_eWay.execute -> FileClient.write
 - FileClient.write -> End

Figure 31 Business Process Designer - Link the Modeling Elements



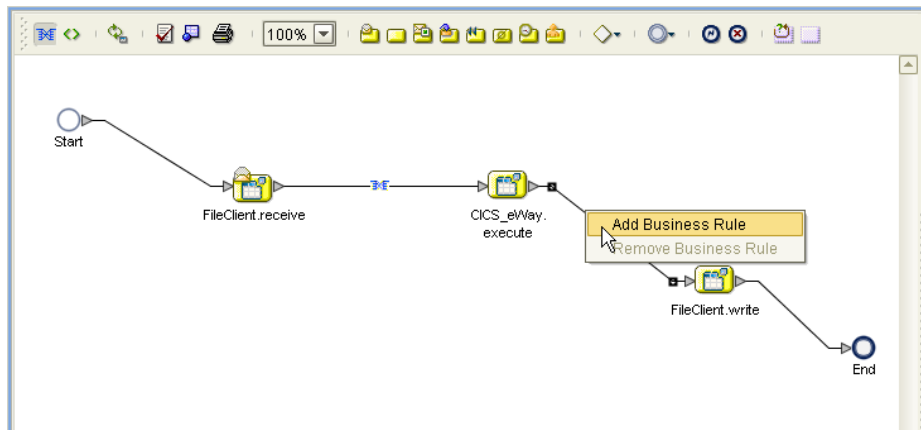
Configuring the Modeling Elements

Business Rules, created between the Business Process Activities, allow you to configure the relationships between the input and output Attributes of the Activities using the Business Process Designer's Business Rule Designer.

Adding Business Rules

- 1 Right-click the link between the **FileClient.receive** and **CICS_eWay.execute** Activities and select **Add Business Rule** from the shortcut menu.
- 2 Repeat step 1 for the **CICS_eWay.execute** to **FileClient.write** link (see Figure 32).

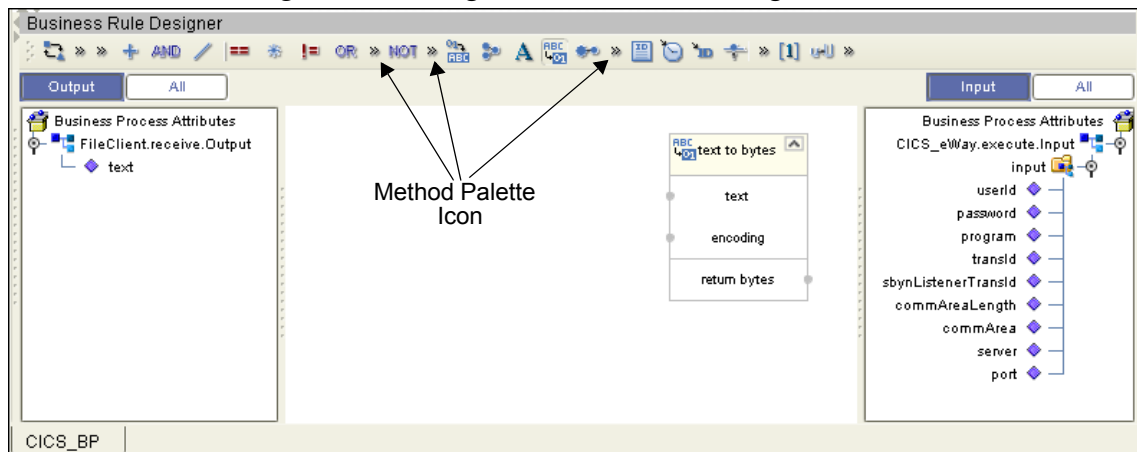
Figure 32 eInsight Business Process Designer - Adding Business Rules



Using the Business Rule Designer

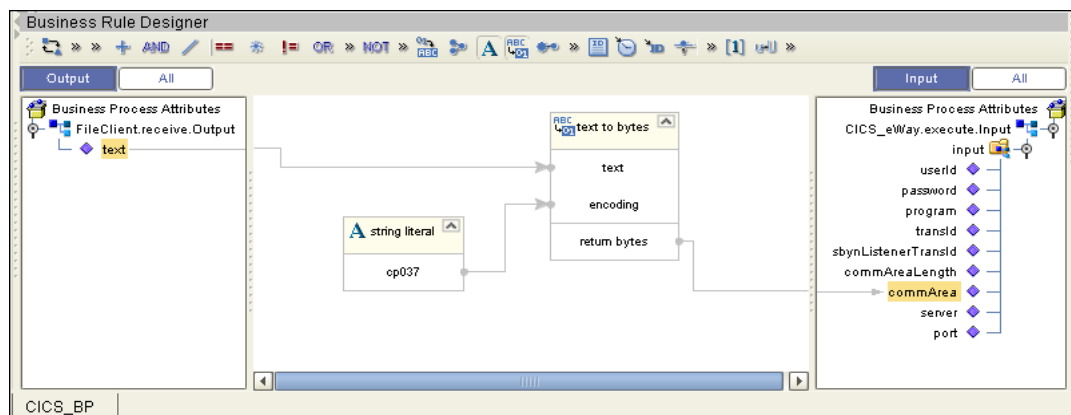
- 1 From the eInsight Business Process Designer toolbar, click the **Map Business Process Attributes** button. The Business Rule Designer appears at the bottom of the eInsight Business Process Designer.
- 2 Click on the Business Rule icon in the link between **FileClient.receive** and **CICS_eWay.execute** to display the Business Rule's Input and Output Attributes in the Business Rule Designer. These Attributes can now be modified.
- 3 From the Business Rule Designer toolbar, click the Method Palette icon (see Figure 33). The Method Palette appears. From the **String** tab, select **bytes to text** and **text to bytes**, and click **Close**. The **bytes to text** and **text to bytes** icons are added to the toolbar.
- 4 Drag the **text to bytes** icon to the Business Rule Designer canvas. The **text to bytes** method box appears on the Business Rule Designer canvas (see [Figure 33 on page 61](#)).

Figure 33 eInsight Business Rule Designer



- 5 Map **text**, under **FileClient.receive.Output** to the **text** input node of the **text to bytes** method box by clicking on **text** under **FileClient.receive.Output** in Output pane of the Business Rule Designer, and dragging the cursor to the **text** input node of the **text to bytes** method box. A line is displayed between the two.
- 6 Map **return bytes** in the **text to bytes** method box, to **commArea** under **CICS_eWay.execute.Input** > **Input** in the Input pane of the Business Rule Designer.
- 7 Drag the **string-literal** icon from the Business Rule Designer toolbar to the Business Rule Designer canvas. The **Input** dialog box appears. Enter **cp037** as the literal value. The string-literal method box is displayed in the Business Rule Designer.
- 8 Map the string-literal value, **cp037**, to the encoding input node of the **text to bytes** method box (see Figure 34).

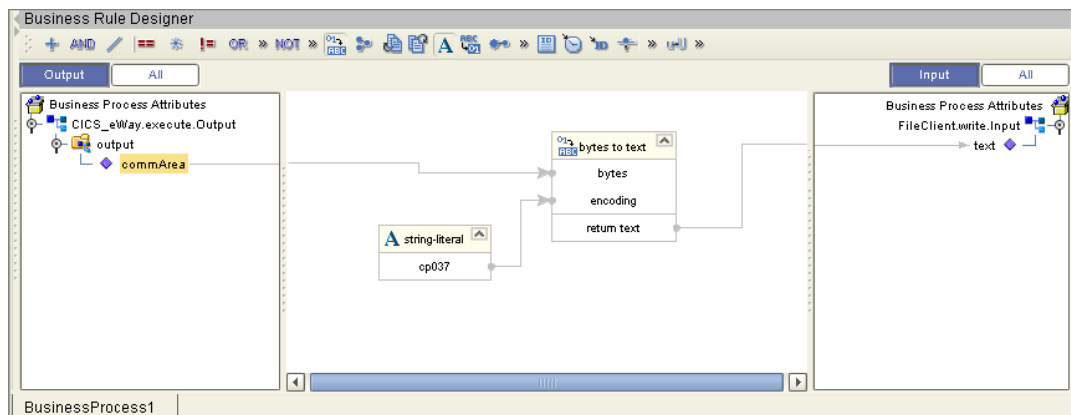
Figure 34 eInsight Business Rule Designer



- 9 From the Enterprise Explorer, right-click the Repository and select **Save Changes into Repository** from the shortcut menu, to save your current changes.
- 10 To configure the second Business Rule, click on the Business Rule icon in the link between the **CICS_eWay.execute** and **FileClient.write** Activities. This Business Rule is now displayed (to be modified) in the Business Rule Designer.

- 11 From the Business Rule Designer toolbar, drag and drop the **bytes to text** icon to the Business Rule Designer canvas. The **bytes to text** method box appears.
- 12 Map **commArea**, under **CICS_eWay.execute.Output > output** in the Output pane, to the **bytes** input node of the **bytes to text** method box.
- 13 Map the **return text** output node of the **bytes to text** method box, to **text** under **FileClient.write.Input** in the Input pane.
- 14 Drag the **string-literal** icon from the Business Rule Designer toolbar to the Business Rule Designer. The **Input** dialog box appears. Enter **cp037** as the literal value. The string-literal method box is displayed in the Business Rule Designer.
- 15 Map the string-literal value, **cp037**, to the **encoding** input node of the **bytes to text** method box (see Figure 35).

Figure 35 eInsight Business Rule Designer



- 16 From the Business Process Designer toolbar, click the **Synchronize Graphical Model and Business Process** icon to synchronize the graphical interface to the Business Process code.
- 17 Save your changes to the Repository.

4.7.3 Creating a Connectivity Map

The Connectivity Map provides a canvas for assembling and configuring a project's components.

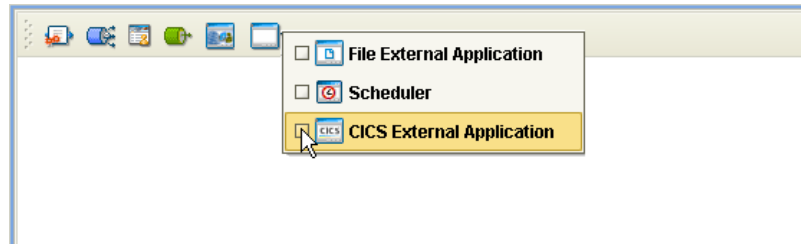
- 1 From the Project Explorer tree, right-click the new **CICS_BP_Sample** project and select **New > Connectivity Map** from the shortcut menu.
- 2 The New Connectivity Map appears and a node for the Connectivity Map is added under the project on the Project Explorer tree labeled **CMap1**. Rename the Connectivity Map **CICS_BP_Sample_CM**.

The icons in the toolbar represent the available components used to populate the Connectivity Map canvas.

Selecting the External Applications

In the Connectivity Map, the eWays are associated with External Systems. For example, to establish a connection to CICS, you must first select CICS as an External System to use in your Connectivity Map (see Figure 36).

Figure 36 Connectivity Map - External Applications



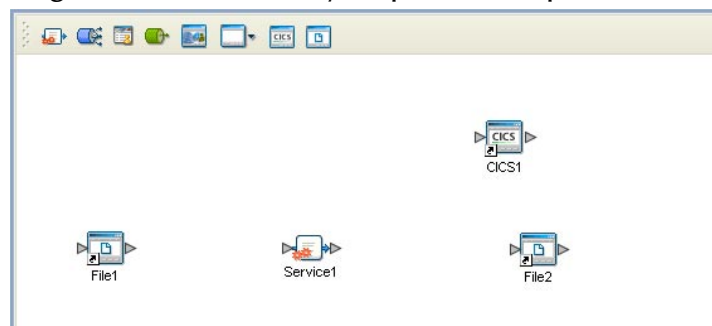
- 1 Click the **External Application** icon on the Connectivity Map toolbar,
- 2 Select the external systems necessary to create your project (for this sample, **CICS** and **File**). Icons representing the selected external systems are added to the Connectivity Map toolbar.

Populating the Connectivity Map

Add the project components to the Connectivity Map by dragging the icons from the toolbar to the canvas.

- 1 For this sample, drag the following components onto the Connectivity Map canvas as displayed in Figure 37:
 - ♦ **File External System (2)**
 - ♦ **Service** (A service is a container for Collaborations, Business Processes, eTL processes, and so forth)
 - ♦ **CICS External System**

Figure 37 Connectivity Map with Components



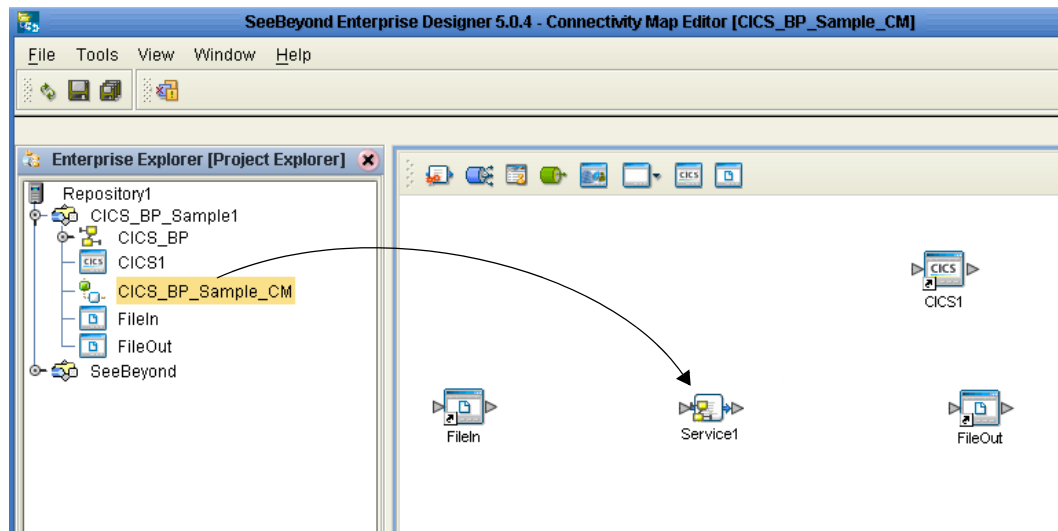
- 2 Rename the **File1** External Application to **FileIn** by right-clicking the object, selecting **Rename** from the shortcut menu, and typing in the new name.
- 3 Rename the **File2** External Application to **FileOut**.
- 4 Save your current changes to the Repository.

4.7.4. Binding the eWay Components

After the Business Processes have been completed, the components are associated and Bindings are created in the Connectivity Map.

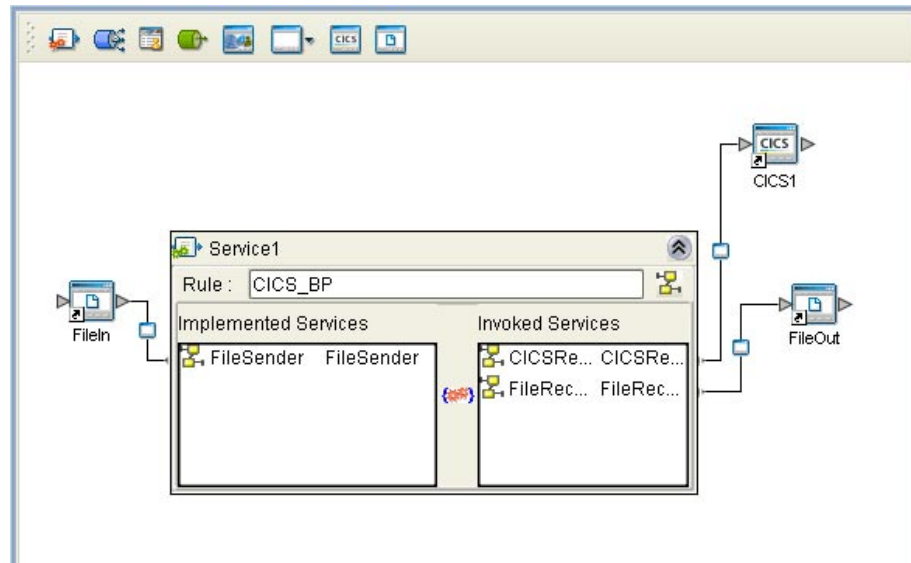
- 1 From the Project Explorer, double-click **CICS_BP_Sample_CM**. The Enterprise Designer canvas now displays the Connectivity Map.
- 2 Drag and drop the **CICS_BP** Business Process from the Project Explorer onto the Service (**Service1**). If the Business Process was successfully associated, the Service's icon changes to a Business Process icon (see Figure 38).

Figure 38 Connectivity Map - Binding the eWay Components



- 3 Double-click **Service1**. The **Service1 Binding dialog box** appears using the **CICS_BP** Rule.
- 4 From the Service1 Binding dialog box, drag **FileSender** (under Implemented Services) to the **FileIn** (File) External Application.
- 5 From the Service1 Binding dialog box, drag **CICS_Receiver** (under Invoked Services) to the **CICS1** External Application.
- 6 From the Service1 Binding dialog box, drag **FileReceiver** to the **FileOut** External Application (see [Figure 39 on page 65](#)). Minimize the Service1 Binding dialog box by clicking the chevrons in the upper-right corner.

Figure 39 Connectivity Map - Associating (Binding) the Project's Components



- 7 Minimize the Service1 Binding dialog box by clicking the chevrons in the upper-right corner.
- 8 Save your current changes to the Repository.

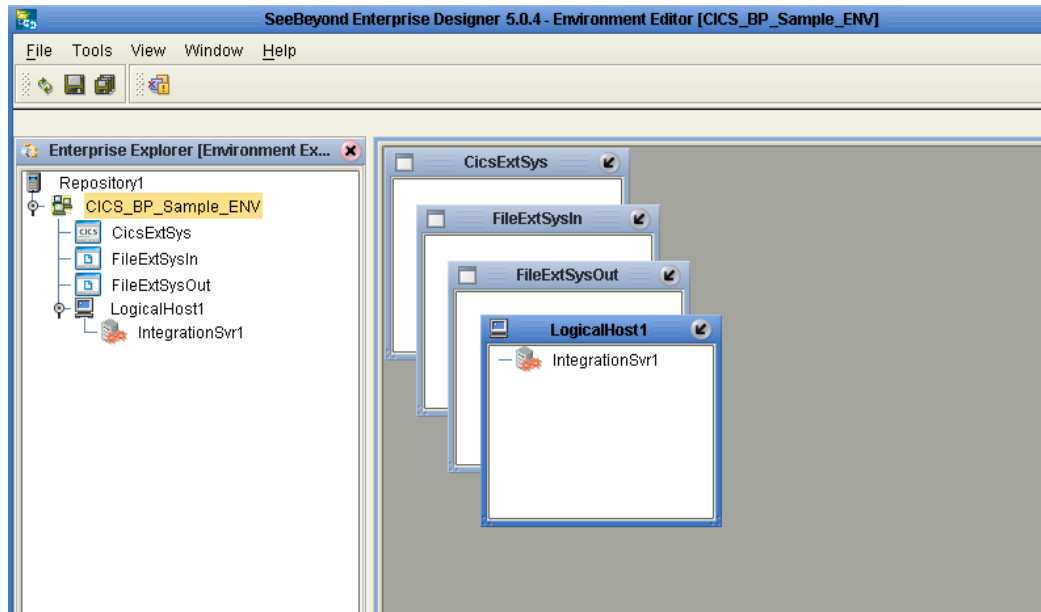
4.7.5. Creating an Environment

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

- 1 From the Enterprise Designer's Enterprise Explorer, click the **Environment Explorer** tab.
- 2 Right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.
- 3 Rename the new Environment to **CICS_BP_Sample_ENV**.
- 4 Right-click **CICS_BP_Sample_ENV** and select **New CICS External System**. Name the External System **CicsExtSys**. Click **OK**. **CicsExtSys** is added to the Environment Editor.
- 5 Right-click **CICS_BP_Sample_ENV** and select **New File External System**. Name the External System **FileExtSysIn** and select **Inbound File eWay** as the **External System Type**. Click **OK**. **FileExtSysIn** is added to the Environment Editor.
- 6 Right-click **CICS_BP_Sample_ENV** and select **New File External System**. Name this External System **FileExtSysOut** and select **Outbound File eWay** as the **External System Type**. **FileExtSysOut** is added to the Environment Editor.
- 7 Right-click **CICS_BP_Sample_ENV** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment Editor tree.

- 8 From the Environment Explorer tree, right-click **LogicalHost1** and select **New SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under LogicalHost1.
- 9 Save changes to the repository. The Environment Explorer and Environment Editor now appear as displayed in Figure 40.

Figure 40 Environment Editor



4.7.6. Configuring the eWay Properties

The CICS_BP_Sample project uses three eWays, each represented in the Connectivity Map as a node between an External Application and a Business Process. eWays facilitate communication and movement of data between the external applications and the eGate system.

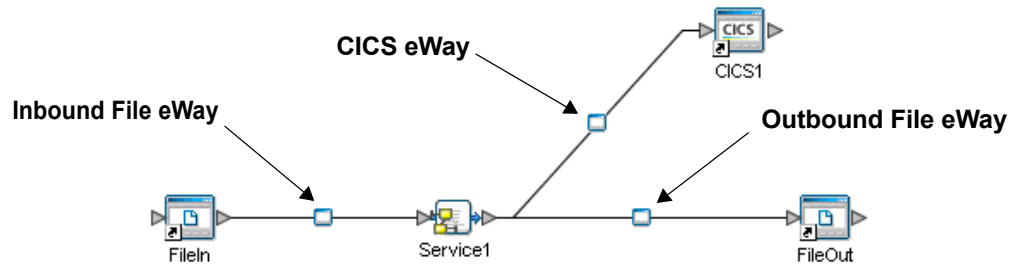
The File eWay properties are configured from the Connectivity Map. The CICS eWay configuration parameters are set from both the Project Explorer's Connectivity Map and the Environment Explorer tree.

To configure the eWays do the following:

Configuring the File eWay Properties

- 1 Double-click the inbound **File eWay** (see [Figure 41 on page 67](#)), select **Inbound File eWay** in the Templates dialog box and click **OK**.

Figure 41 eWays



- 2 The **Properties Sheet** opens to the inbound File eWay properties. Modify the properties for your system, including the settings for the inbound File eWay in Table 2, and click **OK**.

Table 2 Inbound File eWay Settings

Inbound eWay Connection Parameters	
Directory	C:/temp
Input file name	Input*.txt

- 3 In the same way, modify the outbound File eWay properties for your system, including the settings in Table 3, and click **OK**.

Table 3 Outbound File eWay Settings

Outbound eWay Connection Parameters	
Directory	C:/temp
Output file name	output%.dat

Configuring the CICS eWay Properties

The CICS eWay properties must be set in both the Project Explorer’s Connectivity Map and Environment Explorer. For more information on the CICS eWay properties and the Properties Sheet, see [Creating and Configuring the CICS eWay](#) on page 39 or see the *eGate Integrator User’s Guide*.

Modifying the CICS eWay Connectivity Map Properties

- 1 From the **Connectivity Map**, double-click the **CICS eWay**. The **Parameters** dialog box opens to the CICS eWay project configuration properties.
- 2 Modify the CICS eWay Connectivity Map properties for your system, including the settings in [Table 4 on page 68](#), and click **OK**.

Table 4 CICS eWay Connectivity Map Properties

CICS eWay (Connectivity Map) Properties	
Connector Set as directed, otherwise use the default settings	
Class	com.stc.eways.cics.CicsClientConnector
Connection Transport	SeeBeyond CICS Listener
Type	CICS
CICS Client Set as directed, otherwise use the default settings.	
COMMAREA Length	1000
CICS Program	<i>CICS program to be run on server</i>

Modifying the CICS eWay Environment Explorer Properties

- 1 From the **Environment Explorer** tree, right-click the CICS External System (**CicsExtSys** in this sample), and select **Properties** from the shortcut menu. The Properties Sheet appears.
- 2 Modify the CICS eWay environment properties for your system, including the settings in Table 5, and click **OK**.

Table 5 CICS Environment Explorer eWay Properties

CICS eWay Environment Properties	
SeeBeyond CICS Listener Settings Set as directed, otherwise use the default settings	
Host	<i>Host Name</i>
Port	<i>TCP/IP Port Number</i>
CICS Client Settings Set as directed, otherwise use the default settings.	
CICS Password	<i>Password of CICS user</i>
CICS UserId	<i>This is the Id of the CICS user</i>

Note: CICS eWay properties may also be set from the Collaboration Editor (Java) or eInsight Business Rules Designer, in which case they will override the corresponding properties in the eWay's configuration file. Any properties that are not overridden retain their saved settings.

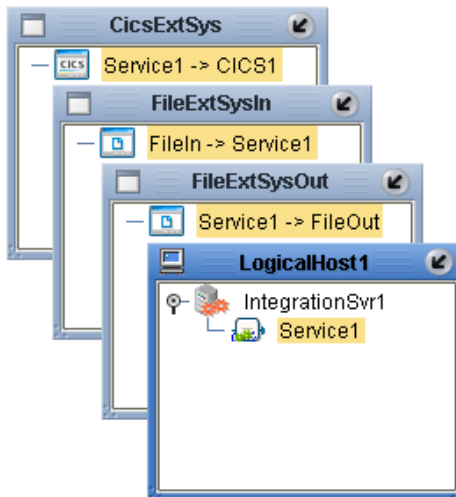
4.7.7 Creating and Activating the Deployment Profile

Deployment Profiles are specific instances of a project in a particular Environment. A Deployment Profile contains information about the assignment of Services and message destinations to integration and message servers (JMS IQ Managers). It also contains version information for all versionable objects in the project. Deployment profiles are created using the Deployment Editor.

To create the CICS_BP_Sample Deployment Profile do the following:

- 1 From the Enterprise Explorer's Project Explorer, right-click the project (CICS_BP_Sample) and select **New > Deployment Profile** from the shortcut menu.
- 2 Enter a name for the Deployment Profile (for this sample CICS_BP_Sample_DP). Make sure that the selected Environment is CICS_BP_Sample_ENV. Click OK. The Deployment Profile Editor appears.
- 3 From the left pane of the Deployment Editor, drag the **FileIn -> Service1** (external application) object to the **FileExtSysIn** window.
- 4 Drag the **Service1 -> FileOut** (external application) object to the **FileExtSysOut** window.
- 5 Drag the **Service1 -> CICS1** (external application) to the **CicsExtSys** window.
- 6 Drag the **Service1** object to **IntegrationSvr1** in the **Logicalhost1** window (see Figure 42).

Figure 42 Deployment Profile



- 7 Click **Activate**. When activation succeeds, save the changes to the Repository.

4.7.8. Running the Project

The following directions assume that the Enterprise Designer was downloaded to C:\ican50. If this is not the case, replace that location in the following directions with the appropriate location.

- 1 From the Enterprise Manager Downloads tab, download **Logical Host - for win32**.
- 2 Extract the file to the **ican50\logicalhost1\logicalhost** directory. You must specify the **logicalhost1** directory for it to be created.
- 3 Navigate to **C:\ican50\logicalhost1\logicalhost\bootstrap\config** directory and open the **logical-host.properties** file using Notepad™.
- 4 Enter the following information in the appropriate fields:
 - ♦ Logical Host root directory: **ican50\logicalhost1\logicalhost**

- ♦ Repository URL: **http://localhost;port number/repository name**
- ♦ Repository user name and password: *Your user name and password*
- ♦ Logical Host Environment name: **CICS_BP_Sample_Env**
- ♦ Logical Host name: **logicalhost1**

Save your changes to **logical-host.properties** and close the file.

- 5 Run the **bootstrap.bat** file in the **eGate50\logicalhost1\logicalhost\bootstrap\bin** directory.
- 6 Copy the sample input data file to the input directory.

Implementing a CICS eWay Project

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

Chapter Topics

- [CICS eWay Components](#) on page 71
- [CICS eWay Sample Projects](#) on page 72
 - ♦ [The CICS_JCE_Sample Project Overview](#) on page 73
 - ♦ [The CICS_COBOL_Sample Project](#) on page 90

5.1 CICS eWay Components

This chapter presents sample CICS eWay projects created using the same procedures as the sample end-to-end project provided in the *eGate Tutorial*. CICS eWay components that are unique to this eWay include the following:

CICS eWay Properties File

The Properties file for the CICS eWay contains the parameters that are used to connecting with a specific external system. These parameters are set using the **Parameters** dialog box. For more information about the CICS eWay Configuration File and the **Parameters** dialog box see [Creating and Configuring the CICS eWay](#) on page 39.

CICS_eWay.CICSClient OTD

The CICS_eWay.CICSClient OTD is provided with the eWay and contains methods and attributes used to create the Business Rules that invoke the CICS program.

Cobol Copybook Wizard

The Cobol Copybook Wizard allows the eGate Integrator to convert Cobol Copybook files into eGate Object Type Definition (OTD) files.

5.2 CICS eWay Sample Projects

Two sample projects are defined in the following sections.

- **CICS_JCE_Sample:** a simple project that demonstrates the CICS eWay receiving data from a file, passing the data to the CICS External System, and writing the output from the CICS System to another file.
- **CICS_COBOL_Sample:** demonstrates a simple CICS project that uses the Cobol Copybook Converter.

5.3 CICS CTG Concerns

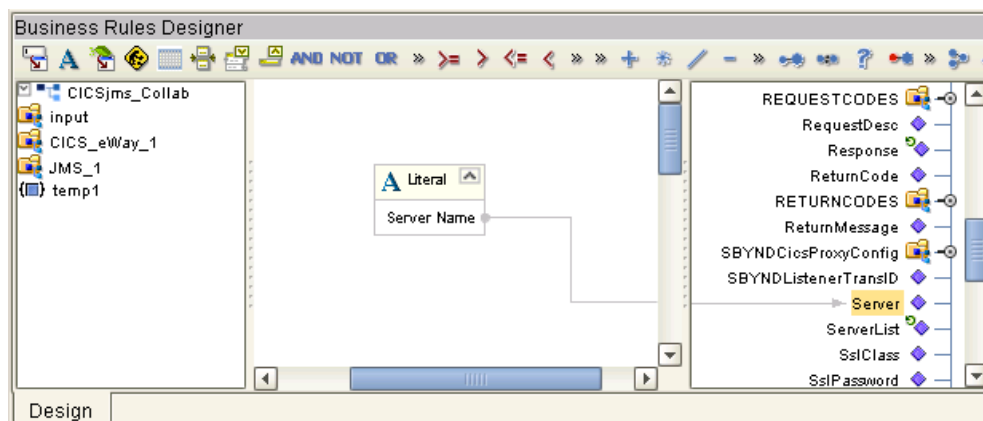
Defining More Than One Server When Using CTG

If CTG is used as the connection transport and more than one server is defined in a CICS Transaction Gateway installation, the target server name must be assigned as a literal to the server node, before to `CICS eWay.execute`, in the Java Collaboration.

To assign the target server, do the following:

- 1 From the (Java) Collaboration Editor's Business Rules pane, click on the Business Rule above the `CICS eWay.execute` rule.
- 2 From the Business Rules toolbar, click the **rule** icon. an **Empty Rule** appears on the Business Rules tree.
- 3 From the Business Rules Designer toolbar, click the Create Literal icon. From the Create Literal dialog box, select **String** as the Type, and enter the name of the target server as the Value. Click **OK**.
- 4 Map the output node of the Literal to **server**, under `CICS_eWay_1` in the right pane of the Business Rules Designer (see Figure 43).

Figure 43 Business Rules Designer - Defining the Target Server



- 5 Continue creating your Collaboration.

Matching the COMMAREA Size

When using the CICS Transaction Gateway transport, data sent to CICS must be padded with spaces, if necessary, to match the full size of the commarea.

5.4 Importing a Sample Project

Sample eWay projects are included as part of the installation CD-ROM package. To import a sample eWay project to the Enterprise Designer do the following:

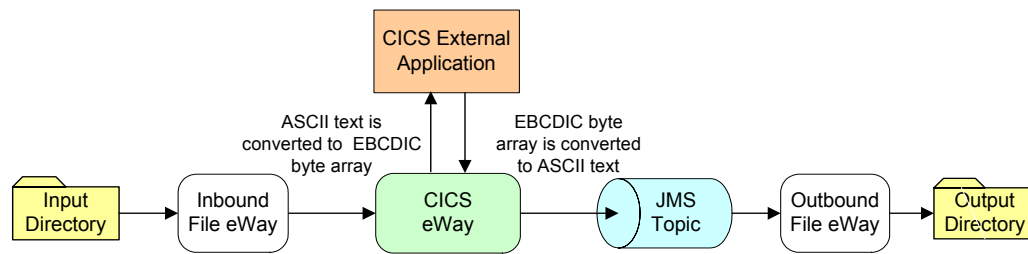
- 1 The sample files are uploaded with the eWay's documentation .sar file and downloaded from the Enterprise Manager's Documentation tab. The **CICS_eWay_Sample.zip** file contains the various sample project zip files. Extract the samples from the Enterprise Manager to a local file.
- 2 Save all unsaved work before importing a project.
- 3 From the Enterprise Designer's Project Explorer pane, right-click the Repository and select **Import** from the shortcut menu. The **Import Manager** appears.
- 4 Browse to the directory that contains the sample project zip file. Select the sample file (for this sample, **CICSjms_JCE_Sample.zip**) and click **Import**. After the sample project is successfully imported, click **Close**.
- 5 Before an imported sample project can be run you must do the following:
 - ♦ Create an **Environment** (see [Creating an Environment](#) on page 84)
 - ♦ Configure the eWays for your system (see [Configuring the eWays](#) on page 85)
 - ♦ Create a **Deployment Profile** (see [Creating and Activating the Deployment Profile](#) on page 88)

5.5 The CICS_JCE_Sample Project Overview

The CICS eWay project, **CICS_JCE_Sample**, demonstrates the following:

- Input data (ASCII text) is read from an external directory by the Inbound File eWay.
- The ASCII text data is converted to EBCDIC (byte array) using cp037 encoding, and published by the CICS eWay to the CICS external system.
- The CICS eWay subscribes to the CICS external system, receives EBCDIC input data, and converts it to String ASCII data using cp037 encoding. The data is then published to the JMS Topic.
- The outbound File eWay subscribes to the JMS Topic and publishes the data to an external directory (see [Figure 44 on page 74](#)).

Figure 44 CICS_JCE_Sample Project



5.6 Creating the CICS_JCE_Sample Project

The following pages provide step by step directions for manually creating the sample eWay components.

5.6.1. Creating a Project

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

- 1 Start the Enterprise Designer.
- 2 From the Enterprise Explorer's Project Explorer tab, right-click your Repository on the Project Explorer tree and select **New Project**. A new project (Project1) appears on the Project Explorer tree.
- 3 Click twice on **Project1** and rename the project (for this sample, **CICS_JCE_Sample**).

5.6.2 Creating a Connectivity Map

The Connectivity Map provides a canvas for assembling and configuring a project's components.

- 1 From the Project Explorer tree, right-click the new **CICS_JCE_Sample** project and select **New > Connectivity Map** from the shortcut menu.
- 2 The New Connectivity Map appears and a node for the Connectivity Map is added under the project on the Project Explorer tree labeled **CMap1**. Rename the Connectivity Map to **CICSjms_JCE_Sample_CM**.

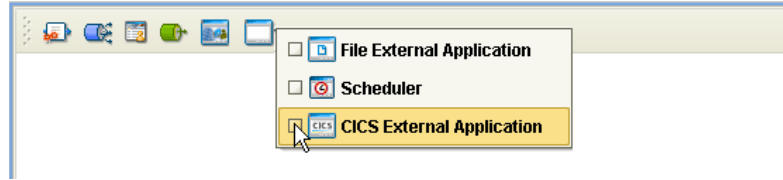
The icons in the toolbar represent the available components used to populate the Connectivity Map canvas.

Selecting the External Applications

The icons in the toolbar represent the available components used to populate the Connectivity Map canvas.

In a Connectivity Map, the eWays are associated with External Systems. For example, to establish a connection to CICS, you must first select CICS as an External System to use in your Connectivity Map (see Figure 45).

Figure 45 Connectivity Map - External Applications



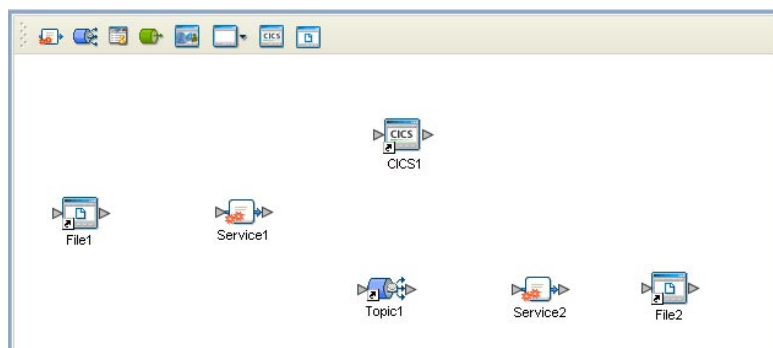
- 1 Click the **External Application** icon on the Connectivity Map toolbar,
- 2 Select the external systems required by your project (for this sample, the **CICS** and **File** External Applications). Icons representing the selected external systems are added to the Connectivity Map toolbar.

Populating the Connectivity Map

Add the project components to the Connectivity Map by dragging the icons from the toolbar to the canvas.

- 1 For this sample, drag the following components onto the Connectivity Map canvas as displayed in Figure 46:
 - ♦ **File External System** (2)
 - ♦ **Service** (2) A service is a container for Collaborations, Business Processes, eTL processes, and so forth.
 - ♦ **Topic**
 - ♦ **CICS External System**

Figure 46 Connectivity Map with Components



- 2 Rename the objects by right-clicking the object, selecting **Rename** from the shortcut menu, and typing in the new name. Change the names as follows:
 - ♦ **File1** to **FileIn**
 - ♦ **Service1** to **CICSjmsCollab**
 - ♦ **Service2** to **JMS2FileCollab**

◆ **File2 to FileOut**

- From the Enterprise Designer toolbar, click the **Save All** icon to save your current changes to the Repository.

5.6.3. Creating the Collaboration Definitions

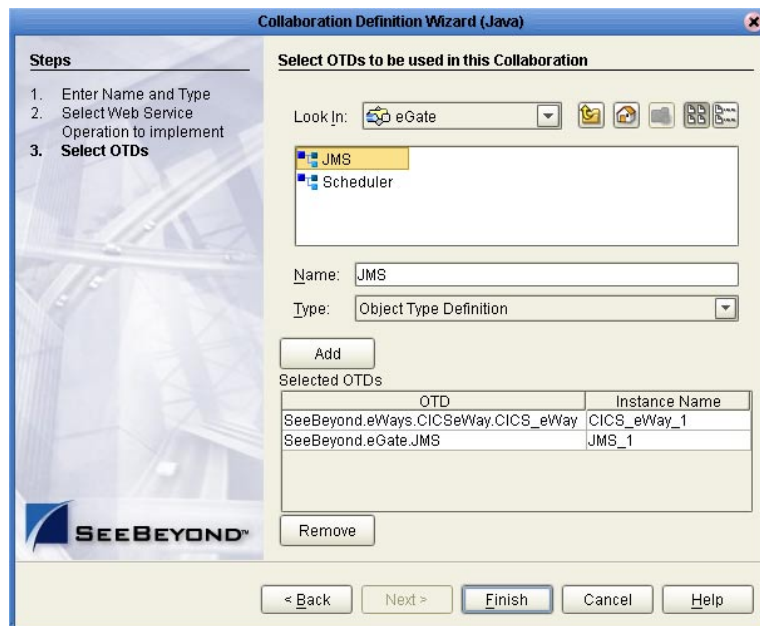
The next step in the sample is to create two Collaborations using the Collaboration Definition Wizard (Java). Once the Collaboration Definitions have been created, the Business Rules of the Collaborations are written using the Collaboration Editor.

Creating the CICSjms_Collab Collaboration

The CICSjms_Collab Collaboration defines transactions from the inbound file application to the CICS eWay and the CICS application, and back to the JMS Topic.

- From the Project Explorer, right-click the sample project and select **New > Collaboration Definition (Java)** from the shortcut menu. The **Collaboration Definition Wizard (Java)** appears.
- Enter a Collaboration Definition name (for this sample CICSjms_Collab) and click **Next**.
- For Step 2 or the wizard, from the Web Services Interfaces selection window, double-click **SeeBeyond > eWays > File > FileClient > receive**. The File Name field now displays **receive**. Click **Next**.
- For Step 3 of the wizard, from the Select OTDs selection window, double-click **SeeBeyond > eWays > CICSeWay > CICS_eWay**. The **CICS_eWay** OTD is added to the Selected OTDs field.
- Click the **Up One Level** button to return to the Repository. Double-click **SeeBeyond > eGate > JMS**. The **Selected OTDs** field now lists the **JMS** OTD (see Figure 47).

Figure 47 Collaboration Definition Wizard (Java) - Select Web Service Interface



- 6 Click **Finish**. The Collaboration Editor with the new CICSjms_Collab Collaboration appears in the right pane of the Enterprise Designer.

Creating the JMS2File_Collab Collaboration (Java)

The JMS2File_Collab Collaboration defines transactions made from the JMS Topic to the Outbound File eWay.

- 1 From the Project Explorer, right-click the sample project and select **New > Collaboration Definition (Java)** from the shortcut menu. The **Collaboration Definition Wizard (Java)** appears.
- 2 Enter a Collaboration Definition name (for this sample **JMS2File_Collab**) and click **Next**.
- 3 For Step 2 of the wizard, from the Web Services Interfaces selection window, double-click **SeeBeyond > eGate > JMS > receive**. The File Name field now displays **receive**. Click **Next**.
- 4 For Step 3, **Select OTDs**, from the Select OTDs selection window, double-click **SeeBeyond > eWays > File > FileClient**. The Selected OTDs field now lists the **FileClient** OTD.
- 5 Click **Finish**. The Collaboration Editor with the new JMS2File_Collab Collaboration appears.

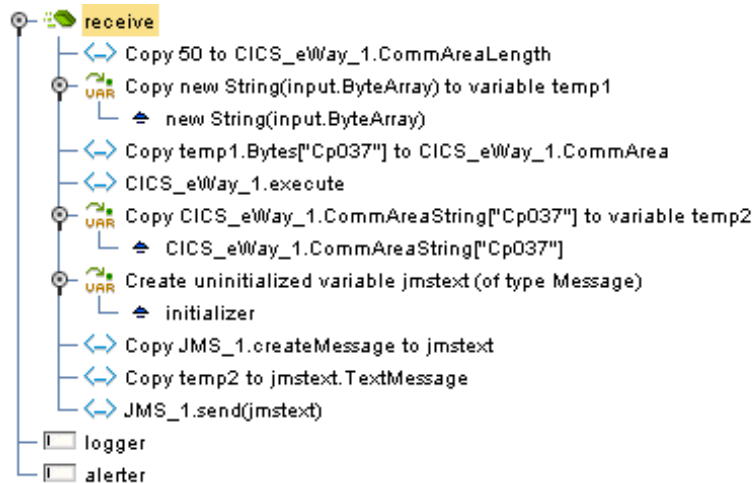
5.6.4. Using the Collaboration Editor (Java)

The next step in the sample is to create the Business Rules of the Collaboration using the Collaboration Editor.

Creating the CICSjms_Collab Business Rules

Be careful to open all nodes specified in the directions to connect to the correct item. The CICSjms_Collab Collaboration contains the Business Rule displayed in Figure 48.

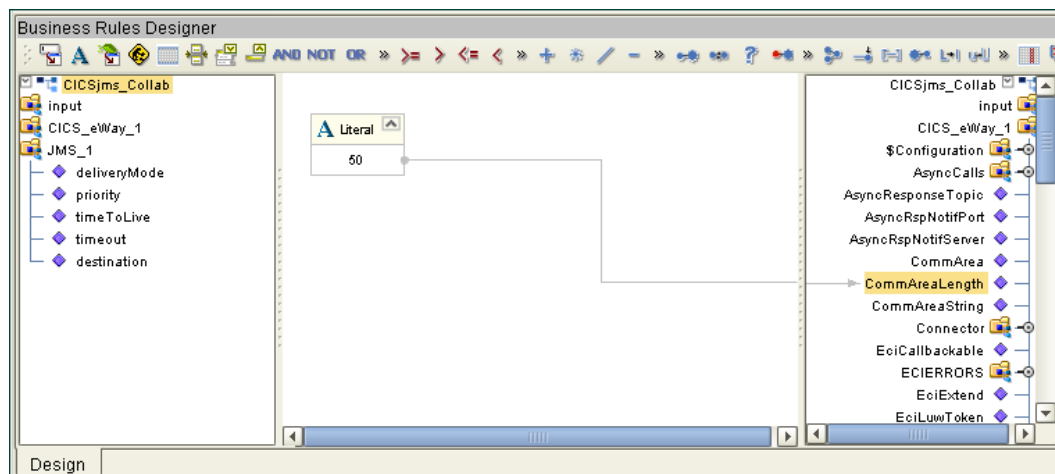
Figure 48 CICSjms_Collab Business Rules



To create the CICSjms_Collab Collaboration Business Rules do the following:

- 1 From the Project Explorer tree, double-click **CICSjms_Collab** to open the Collaboration Editor (Java) to the CICSjms_Collab Collaboration.
- 2 To create comments for the Business Rules, from the Business Rules toolbar, click the **comment** icon. The **Enter a Comment** dialog box appears. Enter the comment and click **OK**. The comment is placed on the Business Rules tree under the last selected item. Once the Comment is created, it can be moved by clicking the comment and dragging it up or down the Business Rules tree to a new location.
- 3 Double-click the **input**, **FileClient_1**, and **CICCS_eWay_1** in the Transformation Designer panes to expand the nodes.
- 4 To create the **Copy 50 to CICS_eWay_1.CommAreaLength** rule do the following:
 - A From the Transformation Designer toolbar, click the **Create Literal** icon. The **Create Literal** method box appears. Select **int** as the type, **50** as the value, and click **OK**. The Literal method box appears in the Transformation Designer.
 - B Map the **50** output node of the Literal method box to **CommAreaLength** under **CICS_eWay_1** in the right pane of the Collaboration Editor. To do this, click on the **50** output node in the Literal method box and drag the cursor to **CommAreaLength** under **CICS_eWay_1** in the right pane of the Transformation Designer (see Figure 49).

Figure 49 CICSjms_Collab - Copy 50 to CICS_eWay_1.CommAreaLength Rule

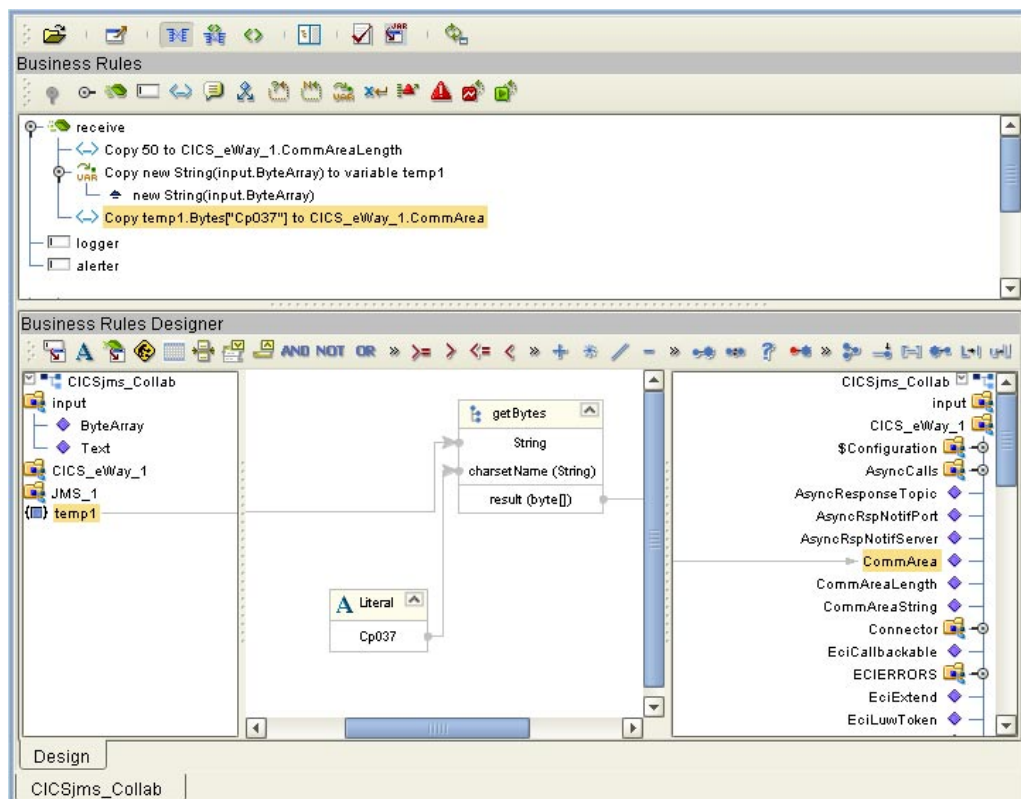


- 5 To create the (variable) **Copy new String(input.ByteArray) to variable temp1** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Collaboration Editor toolbar, click the Advance mode icon. The Java Source Editor appears as the bottom pane of the Collaboration Editor.
 - C From the Java Source Editor, replace *//New rule* with the following code:

```
String temp1 = new String( input.getByteArray() );
```
 - D From the Java Source Editor toolbar, click the **Commit Changes** icon. A new object named temp1 is added to the left pane of the Business Rules Designer.

- 6 To create the **Copy temp1.Bytes["Cp037"] to CICS_eWay_1.CommArea** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the left pane of the Business Rules Designer, right click **temp1** and select **Select a method to call** from the shortcut menu.
 - C From the Method selection box, select **getBytes(java.lang.String charsetName)**. The **getBytes** method box appears on the Business Rules Designer canvas.
 - D From the Business Rules Designer toolbar, click on the **Create Literal** icon. The **Create Literal** dialog box appears.
 - E From the Create Literal dialog box, select **String** as Type, enter **cp037** as the Value, and click **OK**. The **Literal** method box appears on the Business Rules Designer canvas.
 - F Map the **cp037** output node of the Literal method box to the **charsetName (String)** input node of the getBytes method box. To do this, click on **cp037** in the Literal method box and drag the cursor to **charsetName (String)** in the getBytes method box, to create a link as displayed in Figure 50.
 - G Map the **result (byte[])** output node of the getBytes method box to **CommArea** under **CICS_eWay_1** in the right pane of the Business Rules Designer. The Collaboration Editor now appears as displayed in Figure 50.

Figure 50 CICSjms_Collab - Copy temp1.Bytes["Cp037"] to CICS_eWay_1.CommArea

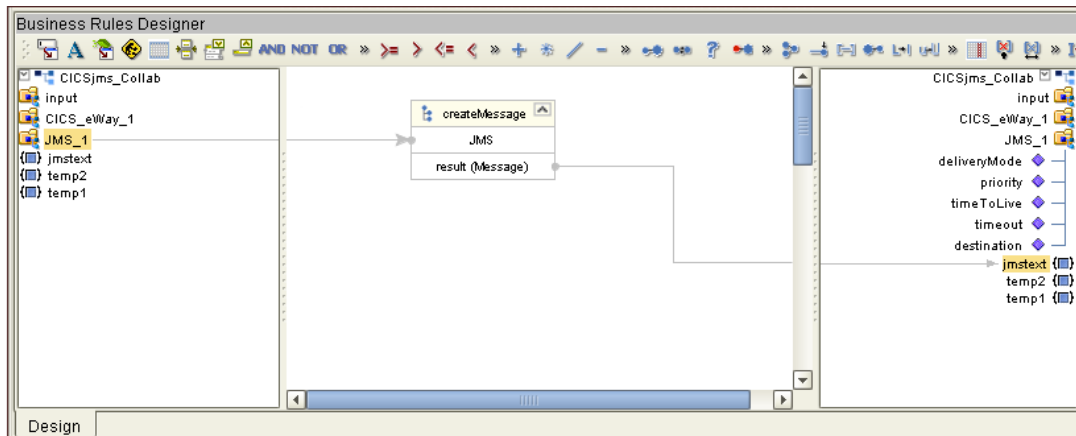


- 7 To create the **CICS_eWay_1.execute** rule do the following:

- A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the left pane of the Business Rules Designer, right click **CICS_eWay_1** and select **Select a method to call** from the shortcut menu.
 - C From the Method selection box, select **execute()**. The **execute** method box appears on the Business Rules Designer canvas.
- 8 To create the (variable) **Copy CICS_eWay_1.CommAreaString["Cp037"] to variable temp2** rule do the following:
- A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Collaboration Editor toolbar, click the Advance mode icon. The Java Source Editor appears as the bottom pane of the Collaboration Editor.
 - C From the Java Source Editor, replace *//New rule* with the following code:

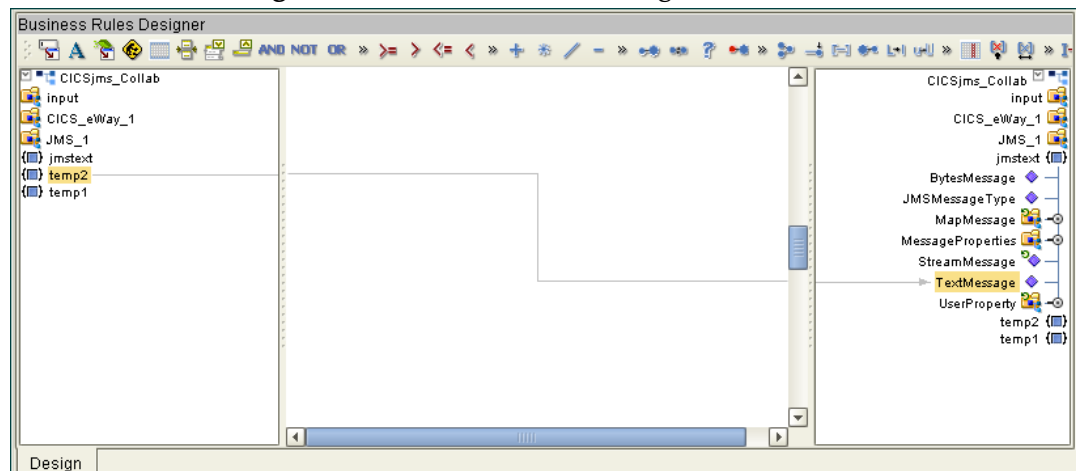
```
String temp2 = CICS_eWay_1.getCommAreaString( "cp037" );
```
 - D From the Java Source Editor toolbar, click the **Commit Changes** icon. A new object named **temp2** is added to the left pane of the Business Rules Designer.
- 9 To create the (variable) **Create uninitialized variable jmstext (of type Message);** rule do the following:
- A From the Business Rules toolbar, click on the **local variable** icon. The Create a Variable dialog box appears.
 - B From the Create a Variable dialog box, select **class**, and click the *ellipsis (...)* button. The **Find Class** dialog box appears. Select **Message** as the Class Name, **com.stc.connectors.jms** as the Package, and click **OK**.
 - C From the Create a Variable dialog box, enter **jmstext** as the Variable Name. Click **OK**.
 - D From the Collaboration Editor toolbar, click the **Refresh Collaboration** icon. A new object named **jmstext** is added to the left pane of the Business Rules Designer.
- 10 To create the **Copy JMS_1.createMessage to jmstext** rule do the following:
- A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the left pane of the Business Rules Designer, right click **JMS_1** and select **Select a method to call** from the shortcut menu.
 - C From the Method selection box, select **createMessage()**. The **createMessage** method box appears on the Business Rules Designer canvas.
 - D Map the **result (Message)** output node of the **createMessage** method box to **jmstext** in the right pane of the Business Rules Designer (see [Figure 51 on page 81](#)).

Figure 51 Business Rules Designer



- 11 To create the **Copy temp2 to jmstext.TextMessage** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B Map **temp2** in the left pane of the Business Rules Designer, to TextMessage under jmstext in the right pane of the Business Rules Designer (see Figure 52).

Figure 52 Business Rules Designer



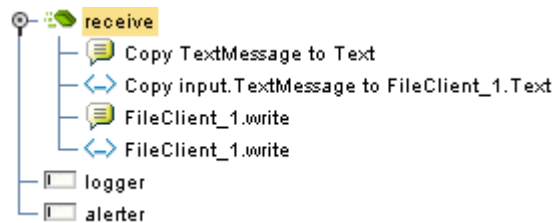
- 12 From the editor's toolbar, click **Validate** to check the Collaboration for errors.
- 13 Save your current changes to the Repository.

Note: See the *eGate Integrator User's Guide* for more information on editing Collaborations.

Creating the JMS2File_Collab Collaboration Business Rules

Be careful to open all nodes specified in the directions to connect to the correct item. The CICSjms_Collab Collaboration contains the Business Rule displayed in [Figure 53 on page 82](#)

Figure 53 JMS2File_Collab Business Rules



To create the **JMS2File_Collab** Collaboration Business Rules do the following:

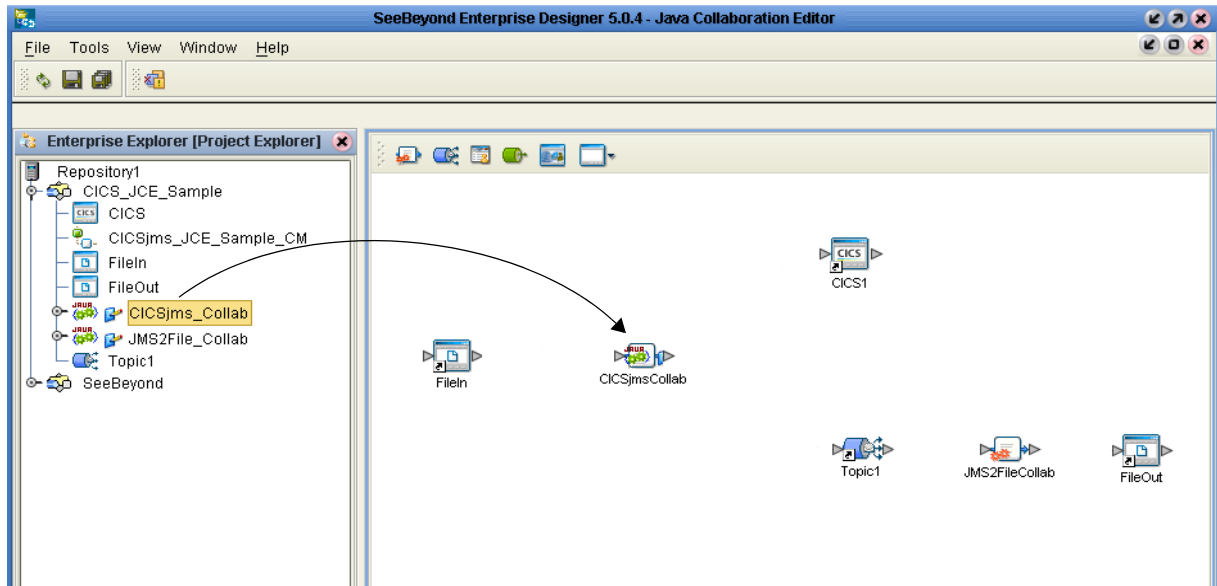
- 1 From the Project Explorer tree, double-click **JMS2File_Collab** to open the Collaboration Editor to the **JMS2File_Collab** Collaboration.
- 2 To create comments for the Business Rules, from the Business Rules toolbar, click the **comment** icon. The **Enter a Comment** dialog box appears. Enter the comment and click **OK**. The comment is placed on the Business Rules tree under the last selected item. Once the Comment is created, it can be moved by clicking the comment and dragging it up or down the Business Rules tree to a new location.
- 3 To create the **Copy input.TextMessage to FileClient_1.Text** rule do the following:
 - A Select the **New rule** in the Business Rules pane.
 - B Map **TextMessage** under **input** in the left pane of the Transformation Designer to **Text** under **FileClient_1** in the right pane.
- 4 To create the **FileClient_1.write** rule do the following:
 - A From the Business Rules toolbar click the **rule** icon to add a new rule.
 - B From the left pane of the Transformation Designer, right-click **FileClient_1** and click **Select a method to call** from the shortcut menu. The method selection box appears.
 - C Select the **write** method. The **write** method box appears in the Transformation Designer canvas.
- 5 From the editor's toolbar, click **Validate** to check the Collaboration for errors.
- 6 Save your current changes to the Repository.

5.6.5. Binding the eWay Components

After the Collaborations have been written, the components are associated and Bindings are created in the Connectivity Map.

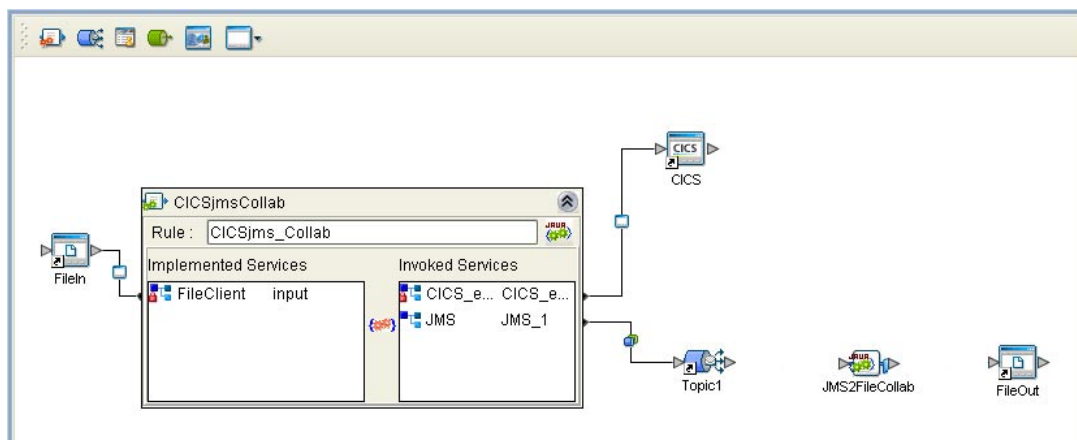
- 1 From the Project Explorer, double-click the Connectivity Map **CICSjms_JCE_Sample_CM**. The Enterprise Designer canvas now displays the **CICSjms_JCE_Sample_CM** Connectivity Map.
- 2 Drag and drop the **CICSjms_Collab** Collaboration from the Project Explorer to the first Service (**CICSjmsCollab**). If the Collaboration was successfully associated, the Service's "gears" icon changes from red to green (see [Figure 54 on page 83](#)).

Figure 54 Connectivity Map - Binding the Components



- 3 Drag and drop **JMS2File_Collab** from the Project Explorer to the second Service (**JMS2FileCollab**).
- 4 From the Connectivity Map canvas, double-click **CICSjmsCollab**. The **CICSjmsCollab** Binding dialog box appears with the **CICSjms_Collab** Rule.
- 5 From the **CICSjmsCollab** Binding dialog box, map **FileClient Input** (under Implemented Services) to the inbound **FileIn** External Application.
- 6 From the **CICSjmsCollab** Binding dialog box, map **CICS_eWay** (under Invoked Services) to the **CICS1** External Application.
- 7 From the **CICSjmsCollab** Binding dialog box, drag **JMS_1** (under Invoked Services) to **Topic1** (see Figure 55).

Figure 55 Connectivity Map - Associating (Binding) the Project's Components



- 8 Minimize the **CICSjmsCollab** Binding dialog box by clicking the chevrons in the upper-right corner.

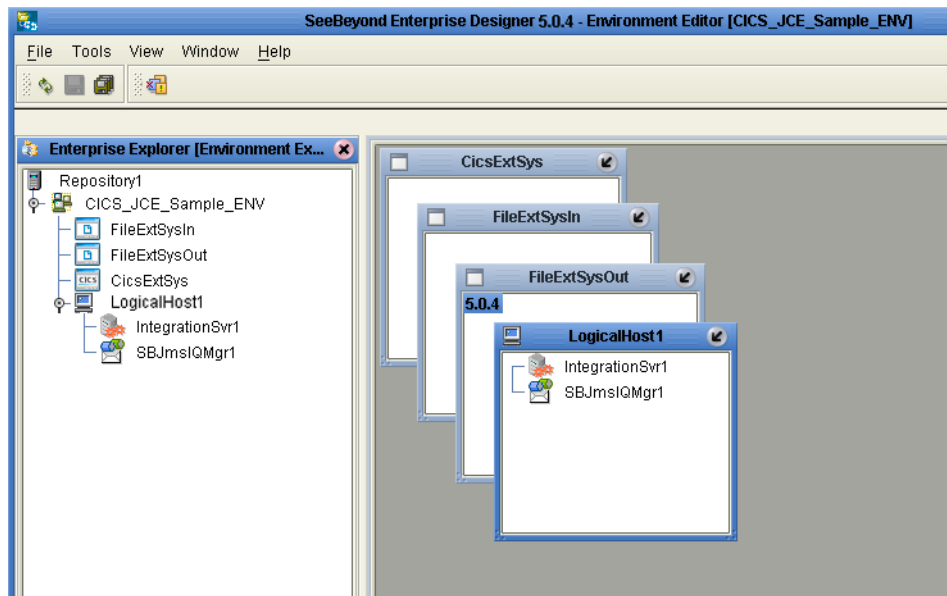
- 9 From the Connectivity Map, double-click **JMS2FileCollab**. The **JMS2FileCollab Binding dialog box** appears with the **JMS2File_Collab** Rule.
- 10 From the **JMS2FileCollab** Binding dialog box, drag **JMS input** (under Implemented Services) to **Topic1**.
- 11 From the **JMS2FileCollab** Binding dialog box, drag **FileClient_1** (under Invoked Services) to the outbound **FileOut** Application.
- 12 Minimize the Collaboration Binding dialog box, and save your current changes to the Repository

5.6.6. Creating an Environment

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

- 1 From the Enterprise Designer's Enterprise Explorer, click the **Environment Explorer** tab.
- 2 Right-click the Repository and select **New Environment**. A new Environment is added to the Environment Explorer tree.
- 3 Rename the new Environment to **CICS_JCE_Sample_Env**.
- 4 Right-click **CICS_JCE_Sample_Env** and select **New CICS External System**. Name the External System **CicsExtSys**. Click **OK**. **CicsExtSys** is added to the Environment Editor.
- 5 Right-click **CICS_JCE_Sample_Env** and select **New File External System**. Name the External System **FileExtSysIn** and select **Inbound File eWay** as the **External System Type**. Click **OK**. **FileExtSysIn** is added to the Environment Editor.
- 6 Right-click **CICS_JCE_Sample_Env** and select **New File External System**. Name this External System **FileExtSysOut** and select **Outbound File eWay** as the **External System Type**. **FileExtSysOut** is added to the Environment Editor.
- 7 Right-click **CICS_JCE_Sample_Env** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment Editor tree.
- 8 Right-click **LogicalHost1** and select **New SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under **LogicalHost1**.
- 9 Right-click **LogicalHost1** and select **New SeeBeyond JMS IQManager**. A new JMS IQ Manager (**SBJmsIQMgr1**) is added to the Environment Explorer tree under **LogicalHost1**. The Environment Explorer and Environment Editor appear as displayed in [Figure 56 on page 85](#).

Figure 56 Environment Editor - CICS_JCE_Sample_Env

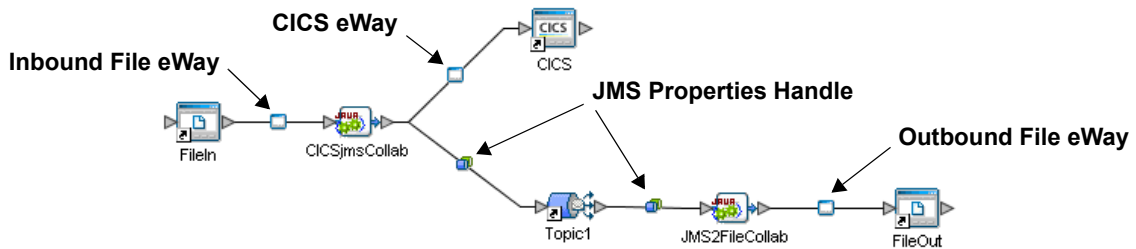


10 Save your current changes to the Repository.

5.6.7. Configuring the eWays

The CICS_JCE_Sample project uses three eWays, each represented in the Connectivity Map as a node between an External Application and a Service (see Figure 57). eWays facilitate communication and movement of data between the external applications and the eGate system.

Figure 57 Configuring the eWays and JMS Properties



The File eWay configuration parameters are configured from the Connectivity Map. The CICS eWay configuration parameters are set from both the Project Explorer or Connectivity Map and the Environment Explorer. To configure the eWays do the following:

Configuring the File eWays

- 1 Double-click the **Inbound File eWay**, select **Inbound File eWay** in the Templates dialog box and click **OK**.

- 2 The **Parameters** dialog box opens to the Inbound File eWay configuration. Modify the configuration for your system, including the settings for the **Inbound** File eWay in Table 6, and click **OK**. The configuration settings are saved for the eWay.

Table 6 Inbound File eWay Settings

Inbound eWay Connection Parameters	
Directory	C:/temp
Input file name	Input*.txt

- 3 In the same way, modify the **Outbound** File eWay configuration for your system, including the settings in Table 7, and click **OK**.

Table 7 Outbound File eWay Settings

Outbound eWay Connection Parameters	
Directory	C:/temp
Output file name	output%.dat

Configuring the CICS eWay

The CICS eWay configuration parameters must be set in both the Project Explorer and Environment Explorer. For more information on the CICS eWay configuration parameters and the **Parameters** dialog box, see [Creating and Configuring the CICS eWay](#) on page 39 or see the *eGate Integrator User's Guide*.

For the CICS_JCE_Sample project, do the following:

Modifying the CICS eWay (Project Explorer) Configuration Parameters

- 1 From the **Connectivity Map**, double-click the **CICS eWay**. The **Parameters** dialog box opens to the CICS eWay project configuration properties.
- 2 Modify the CICS eWay (Project Explorer) configuration for your system, including the settings in Table 8, and click **OK**.

Table 8 CICS Project Explorer eWay Settings

CICS eWay Project-Configuration Parameters	
Connector Set as directed, otherwise use the default settings	
Class	com.stc.eways.cics.CicsClientConnector
Connection Transport	SeeBeyond CICS Listener
Type	CICS
CICS Client Set as directed, otherwise use the default settings.	
COMMAREA Length	1000
CICS Program	<i>CICS program to be run on server</i>

Modifying the CICS eWay (Environment Explorer) Configuration Parameters

- 1 From the **Environment Explorer** tree, right-click the CICS External System (**CicsExtSys** in this sample), and select **Properties**. The **Parameters** dialog box opens to the CICS eWay environment-configuration properties.
- 2 Modify the CICS eWay environment-configuration for your system, including the settings in Table 9, and click **OK**.

Table 9 CICS Environment Explorer eWay Settings

CICS eWay Environment-Configuration Parameters	
SeeBeyond CICS Listener Settings Set as directed, otherwise use the default settings	
Host	<i>Host Name</i>
Port	<i>TCP/IP Port Number</i>
CICS Client Settings Set as directed, otherwise use the default settings.	
CICS Password	<i>Password of CICS user</i>
CICS UserId	<i>This is the Id of the CICS user</i>

Note: *CICS eWay properties may also be set from the Collaboration Editor (Java) or eInsight Business Rules Designer, in which case they will override the corresponding properties in the eWay's configuration file. Any properties that are not overridden retain their saved settings.*

5.6.8. Configuring the JMS Clients

When a Service is linked with a Queue (or Topic), the Enterprise Designer adds a JMS properties handle between the two (see [Figure 57 on page 85](#)). These JMS properties facilitate the transfer and, if necessary, translation of data within the eGate system. JMS configuration properties must be configured in both the Connectivity Map and the Environment Explorer.

For more information on JMS configuration parameters see the *eGate Integrator User's Guide*.

Modifying the JMS Connectivity Map Configuration Parameters

- 1 From the CICSjms_JCE_Sample_CM Connectivity Map, double-click the JMS properties handle located between the **CICSjmsCollab** and **Topic1**.
- 2 The **Properties sheet** appears, containing the default environment-configuration properties. Click **OK** to keep the default settings.
- 3 Double-click the JMS properties handle located between **Topic1** and the **JMS2FileCollab** Collaboration.
- 4 The **Parameters** dialog box appears, containing the default configuration parameters. Click **OK** to keep the default settings.

Modifying the JMS Environment Explorer Configuration Parameters

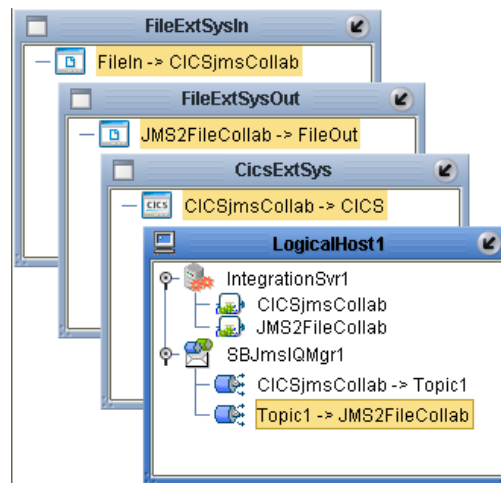
- 1 From the **Environment Explorer** tree, right-click the Message Server found under the Logical Host, and select **Properties**. The **Properties sheet** opens to the JMS Environment-configuration properties.
- 2 Modify the JMS Environment-configuration for your specific system and click **OK**.

5.6.9 Creating and Activating the Deployment Profile

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

- 1 From the Enterprise Explorer's Project Explorer, right-click the project (**CICS_JCE_Sample**) and select **New > Deployment Profile**.
- 2 Enter a name for the Deployment Profile (for this sample **CICS_JCE_Sample_DP**). Make sure that the selected Environment is **CICS_JCE_Sample_Env**. Click **OK**.
- 3 From the left pane of the Deployment Editor, drag **CICSjmsCollab -> Topic1** (External Application) to the **CicsExtSys** window.
- 4 From the left pane of the Deployment Editor, drag the **FileIn -> CICSjmsCollab** (External Application) to the **FileExtSysIn** window.
- 5 From the left pane of the Deployment Editor, drag the **JMS2FileCollab -> FileOut** (External Application) to the **FileExtSysOut** window.
- 6 Drag the following 2 Collaborations:
 - ♦ **CICSjmsCollab**
 - ♦ **JMS2FileCollab**to **IntegrationSvr1** in the **LogicalHost1** window.
- 7 Drag the two Topic components:
 - ♦ **CICSjmsCollab -> Topic1**
 - ♦ **Topic1 -> JMS2FileCollab**to **MessageSvr1** in the **LogicalHost1** window (see [Figure 58 on page 89](#)).

Figure 58 Deployment Profile



- 8 Click **Activate**. When activation succeeds, save the changes to the Repository.

5.6.10. Running the Project

The following directions assume that the Enterprise Designer was downloaded to **C:\ican50**. If this is not the case, replace that location in the following directions with the appropriate location.

- 1 From the Enterprise Manager Downloads tab, download **Logical Host - for win32**.
- 2 Extract the file to the **ican50\logicalhost2\logicalhost** directory. You must specify the **logicalhost2** directory for it to be created.
- 3 Navigate to **C:\ican50\logicalhost2\logicalhost\bootstrap\config** directory and open the **logical-host.properties** file using Notepad™.
- 4 Enter the following information in the appropriate fields:
 - ♦ Logical Host root directory: **ican50\logicalhost2\logicalhost**
 - ♦ Repository URL: **http://localhost:port number/repository name**
 - ♦ Repository user name and password: *Your user name and password*
 - ♦ Logical Host Environment name: **CICS_JCE_Sample_ENV**
 - ♦ Logical Host name: **logicalhost2**

Save your changes to **logical-host.properties** and close the file.

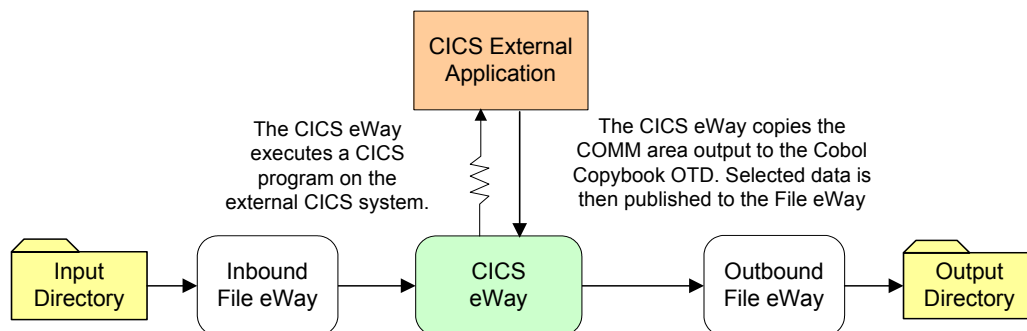
- 5 Run the **bootstrap.bat** file in the **ican50\logicalhost2\logicalhost\bootstrap\bin** directory.
- 6 Copy the sample input data file to the input directory.

5.7 The CICS_COBOL_Sample Project

The CICS eWay project, **CICS_Cobol_Sample**, demonstrates the following:

- The Inbound File eWay subscribes to an external directory, receives the data, and publishes the data to the CICS eWay.
- The CICS eWay, triggered by the data from the File eWay, does the following:
 - ♦ executes a CICS program on the external CICS system (mainframe)
 - ♦ reads the program's output (the contents of its COMM area)
 - ♦ copies the output into a Cobol Copybook OTD that is generated from the same data area description
 - ♦ OTD accessors selectively extract the value of select fields
- This data is published to the outbound File eWay, which publishes the data to an external directory.

Figure 59 CICS_Cobol_Sample Project



The **CICS_Cobol_Sample** is available to import from the Installation CD-ROM package (see [Importing a Sample Project](#) on page 73 for more information). The following pages review how the CICS_Cobol_Sample project is created.

5.7.1. Creating a Project

The first step is to create a new project in eGate Enterprise Designer.

- 1 Start the Enterprise Designer.
- 2 From the Enterprise Explorer's Project Explorer tab, right-click the Repository and select **New Project**. A new project (Project1) appears on the Project Explorer tree.
- 3 Click twice on **Project1** and rename the project (for this sample, **CICS_COBOL_Sample**).

5.7.2 Creating a Connectivity Map

The Connectivity Map provides a canvas for assembling and configuring a project's components.

- 1 In Enterprise Explorer's Project Explorer, right-click the new project (CICS_COBOL_Sample) and select **New > Connectivity Map** from the shortcut menu.
- 2 The New Connectivity Map appears and a node for the Connectivity Map is added under the project on the Project Explorer tree labeled **CMap1**. Rename the **CMap1** Connectivity Map to **CICS_COBOL_Sample_CM**

Selecting the External Applications

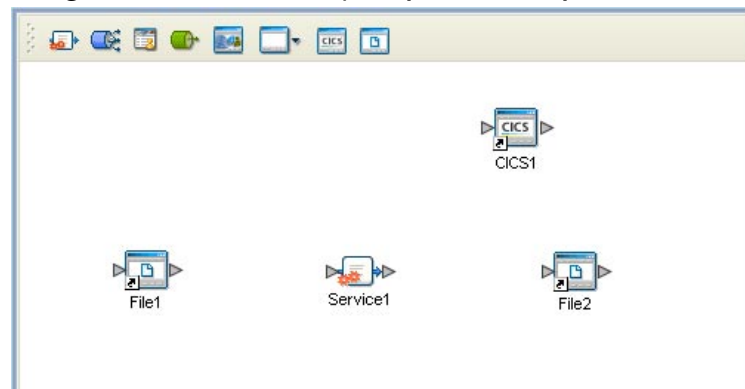
- 1 Click the **External Application** icon on the Connectivity Map toolbar,
- 2 Select the external systems needed for your project (for this sample, **CICS** and **File**). Icons representing the selected external systems are added to the Connectivity Map toolbar.

Populating the Connectivity Map

Add the project components to the Connectivity Map by dragging the icons from the toolbar to the canvas.

- 1 For the **CICS_COBOL_Sample** project, drag the following components onto the Connectivity Map canvas as displayed in Figure 60:
 - ♦ **File** External System (2)
 - ♦ **Service**
 - ♦ **CICS** External System

Figure 60 Connectivity Map with Components



- 2 Rename the objects by right-clicking the object, selecting **Rename** from the shortcut menu, and typing in the new name. Change the names as follows:
 - ♦ **File1** to **FileIn**
 - ♦ **File2** to **FileOut**

- 3 Save your current changes to the Repository.

5.7.3 Creating an Object Type Definition

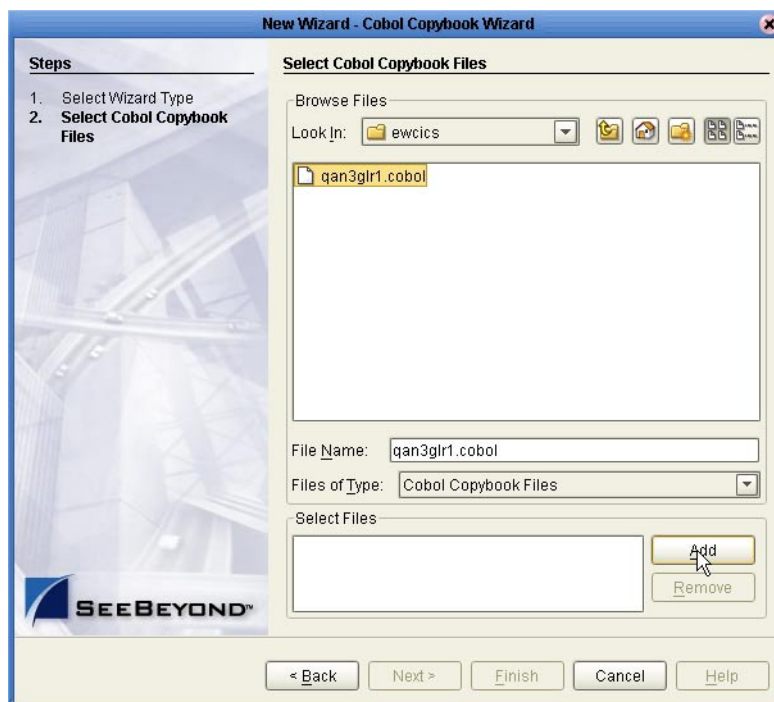
Object Type Definitions (OTDs) contain the data structures and rules that define objects. OTDs are used in the Enterprise Designer to create Collaboration Definitions for building data transformations. OTDs are created using the **Object Type Definition Wizard**.

Using the Cobol Copybook Wizard

The **CICS_COBOL_Sample** project uses an OTD that is created using the **Cobol Copybook Wizard**. A sample Cobol data file, **qan3glr1.cobol** is available from the downloaded CICS eWay Samples folder. To create the Cobol Copybook OTD using the sample Cobol file do the following:

- 1 Copy the sample Cobol file, **qan3glr1.cobol**, from the installation CD-ROM to a temporary file.
- 2 From the Project Explorer, right-click the **CICS_COBOL_Sample** project, and select **New > Object Type Definition** from the shortcut menu. The OTD Wizard Selection dialog box appears. Select the **Cobol Copybook Wizard** and click **Next**.
- 3 From the **Select Cobol Copybook Files** page of the Wizard, navigate to the directory that contains the sample Cobol file. The sample file is now visible in the Available Files field (see Figure 61).

Figure 61 Cobol Copybook Wizard - Select Cobol Files



- 4 Select the sample Cobol file and click the **Add** button. The file is added to the **Selected Files** field. Click **Finish**.
- 5 A **Message** dialog box appears if the OTD is created successfully, and the OTD is added to the Project Explorer tree as **qan3glr1**.

5.7.4. Creating a Collaboration Definition (Java)

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

Creating the CICS_COBOL_Colab Collaboration

- 1 From the Project Explorer, right-click the **CICS_COBOL_Sample** project and select **New > Collaboration Definition (Java)** from the shortcut menu. The **Collaboration Definition Wizard (Java)** appears.
- 2 Enter a Collaboration Definition name (for this sample **CICS_COBOL_Collab**) and click **Next**.
- 3 For Step 2 of the Wizard, from the Web Services Interfaces selection window, double-click **SeeBeyond > eWays > File > FileClient > receive**. The File Name field now displays **receive**. Click **Next**.
- 4 For Step 3 of the Wizard, from the Select OTDs selection window, double-click **SeeBeyond > eWays > CICS_eWay > CICS_eWay**. The **CICS_eWay** OTD is added to the Selected OTDs field.
- 5 Click the **Up One Level** button to return to the Repository directory. Double-click **SeeBeyond > eWays > File > FileClient**. The **FileClient** OTD is added to the Selected OTDs field.
- 6 Click the **Up One Level** button to return to the Repository directory. Double-click **CICS_COBOL_Sample > qan3glr1**. The **qan3glr1** OTD is added to the Selected OTDs field.
- 7 Click **Finish**. The Collaboration Editor opens to the new Collaboration in the right pane of the Enterprise Designer.

5.7.5. The CICS_COBOL_Collab Business Rules

The the **CICS_COBOL_Sample** project uses one Collaboration created in the previous section, **CICS_COBOL_Collab**.

CICS_COBOL_Collab Business Rules

The CICS_COBOL_Collab Collaboration contains the Business Rule displayed in **Figure 62 on page 94**. These rules are created using the Collaboration Editor (Java).

Figure 62 CICS_COBOL_Collab Business Rules

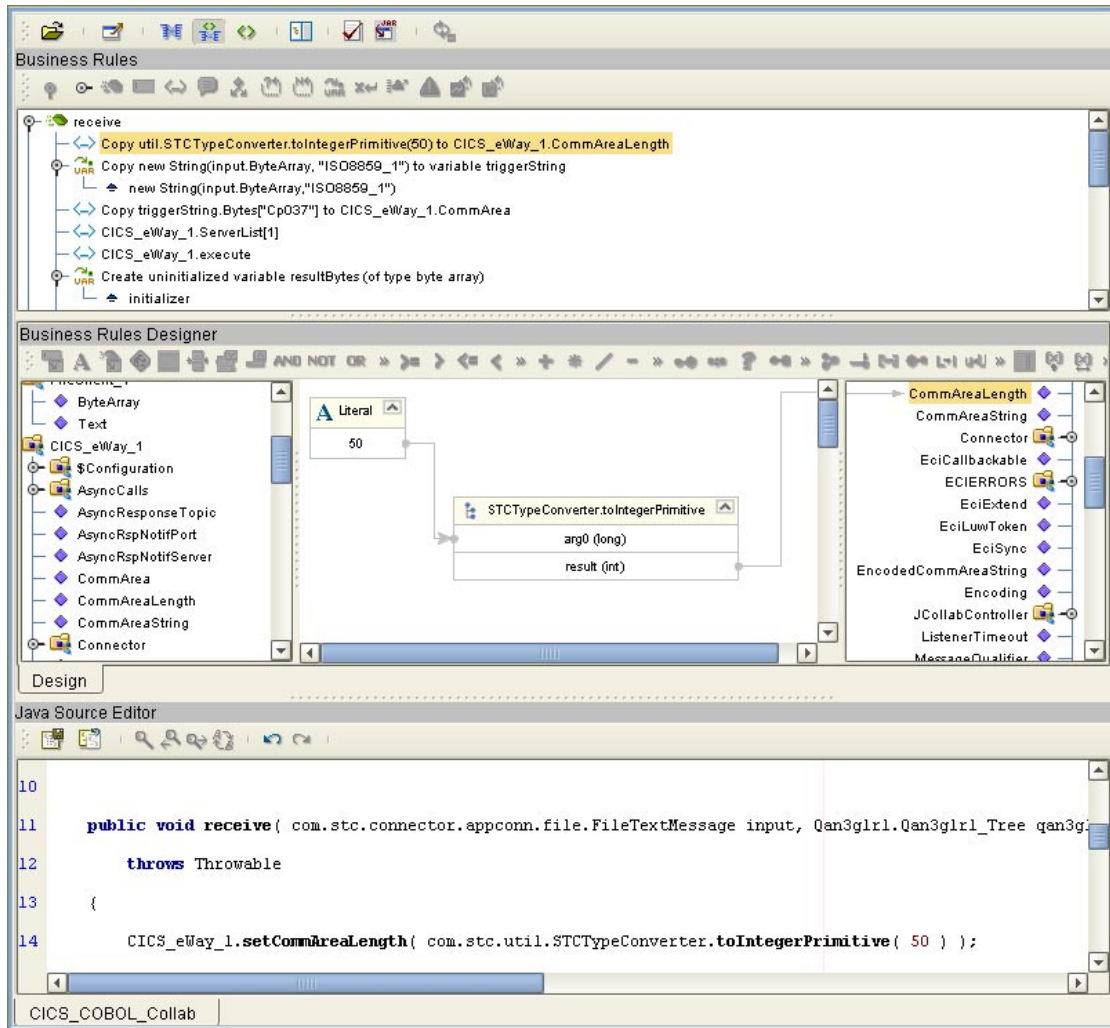


To create the CICS_COBOL_Collab Business Rules do the following:

- 1 From the Project Explorer tree, double-click **CICS_COBOL_Collab** to open the Collaboration Editor to the Collaboration.
- 2 Double-click the **input**, **FileClient_1**, and **CICCS_eWay_1** in the Transformation Designer panes to expand the nodes.
- 3 To create the **Copy util.STCTypeConverter.toIntegerPrimitive(50) to CICS_eWay_1.CommAreaLength** rule do the following:
 - A From the Collaboration Editor toolbar click the **Advance mode** icon. The Java Source Editor now appears as the bottom pane of the Collaboration Editor.
 - B From the Collaboration Editor pane, select **//New rule**, and replace it with the following code:


```
CICS_eWay_1.setCommAreaLength( com.stc.util.STCTypeConverter.toIntegerPrimitive( 50 ) );
```
 - C From the Java Source Editor toolbar click the **Commit changes** icon (see [Figure 63 on page 95](#)).

Figure 63 Collaboration Editor



- 4 To create the (variable) **Copy new String(input.ByteArray, "ISO8859_1") to variable triggerString** rule do the following:
 - A From the Business Rules toolbar, click the **local variable** icon. The **Create a Variable** dialog box appears. Select **Class** and browse to **java.lang.String** as the Class Name. Enter **triggerString** as the Variable Name, and click **OK**.
 - B From the Business Rules pane, expand the variable and select the variable's initializer.
 - C From the Business Rules Designer toolbar, click the **Call New Constructor** icon. The **Call New Constructor** dialog box appears.
 - D Select **String** for All Classes and **java.lang.String(byte[] bytes,java.lang.String charsetName)** as the Constructors. Click **OK**. The **java.lang.String** method box is appears on the Business Rules Designer canvas.
 - E From the Business Rules Designer toolbar, click the **Create Literal** icon. The **Create Literal** dialog box appears. Select **String** as the Type and **ISO8859_1** as the Value. Click **OK**.

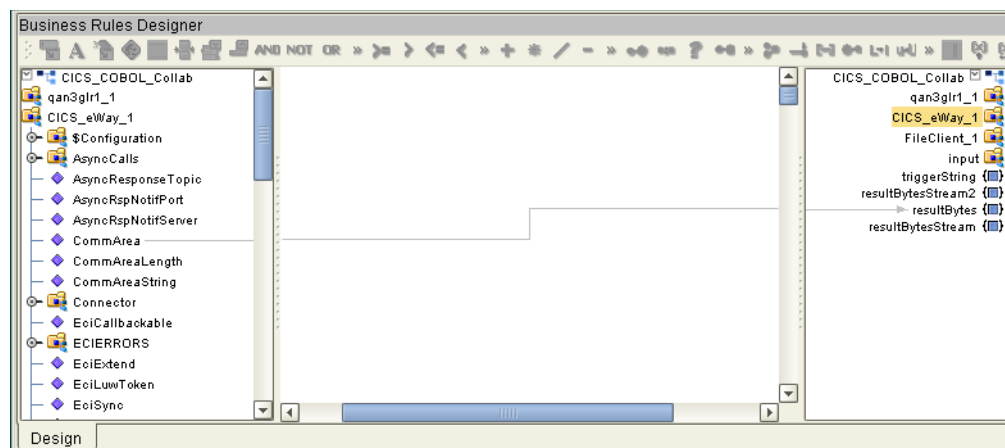
- F Map **ByteArray** under **input** in the left pane of the Business Rules Designer, to the bytes (byte[]) input node of the **java.lang.String** method box.
- G Map the **ISO8859_1** output node of the **literal** method box, to the **charsetName (String)** input node of the **java.lang.String** method box.
- 5 To create the **Copy triggerString.Bytes["Cp037"] to CICS_eWay_1.CommArea** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the left pane of the Business Rules Designer, right click **triggerString** and select **Select a method to call** from the shortcut menu.
 - C From the Method selection box, select **getBytes(java.lang.String charsetName)**. The **getBytes** method box appears on the Business Rules Designer canvas.
 - D From the Business Rules Designer toolbar, click on the **Create Literal** icon. The **Create Literal** dialog box appears.
 - E From the Create Literal dialog box, select **String** as Type, enter **cp037** as the Value, and click **OK**. The **Literal** method box appears on the Business Rules Designer canvas.
 - F Map the **cp037** output node of the Literal method box to the **charsetName (String)** input node of the **getBytes** method box. To do this, click on **cp037** in the Literal method box and drag the cursor to **charsetName (String)** in the **getBytes** method box, to create a link as displayed in Figure 50.
 - G Map the **result (byte[])** output node of the **getBytes** method box to **CommArea** under **CICS_eWay_1** in the right pane of the Business Rules Designer.
- 6 To create the **CICS_eWay_1.getServerList(1)** Business Rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Java Source Editor pane, select **//New rule**, and replace it with the following code:

```
CICS_eWay_1.getServerList( 1 );
```
 - C From the Collaboration Editor toolbar click the **Commit changes** icon.
- 7 To create the **CICS_eWay_1.execute** Business Rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the left pane of the Business Rules Designer, right click **CICS_eWay_1** and select **Select a method to call** from the shortcut menu.
 - C From the Method selection box, select **execute()**. The **execute** method box appears on the Business Rules Designer canvas.
- 8 To create the variable **Create uninitialized variable resultBytes (of type byte array)** rule do the following:
 - A From the Business Rules toolbar, click the **local variable** icon. The **Create a Variable** dialog box appears. Select **Primitive** with byte as the value, and select **Is array**, for Type. Enter **resultByte** as the Variable Name, and click **OK**.

- 9 To create the variable **Create uninitialized variable resultBytesStream (of type ByteArrayInputStream)** rule do the following:
 - A From the Business Rules toolbar, click the **local variable** icon. The **Create a Variable** dialog box appears. Select **Class** and browse to **java.io.ByteArrayInputStream** as the value. Enter **resultBytesStream** as the Variable Name, and click **OK**.
- 10 To create the variable **Create uninitialized variable otdStream (of type OtdInputStream)** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Collaboration Editor pane, select **//New rule**, and replace it with the following code:


```
com.stc.otd.runtime.OtdInputStream otdStream;
```
 - C From the Java Source Editor toolbar click the **Commit changes** icon.
- 11 To create the **Copy CICS_eWay_1.CommArea to resultBytes** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B Map **CommArea** under **CICS_eWay_1** in the left pane of the Business Rules Designer, to **resultBytes** in the right pane of the Business Rules Designer (see Figure 64).

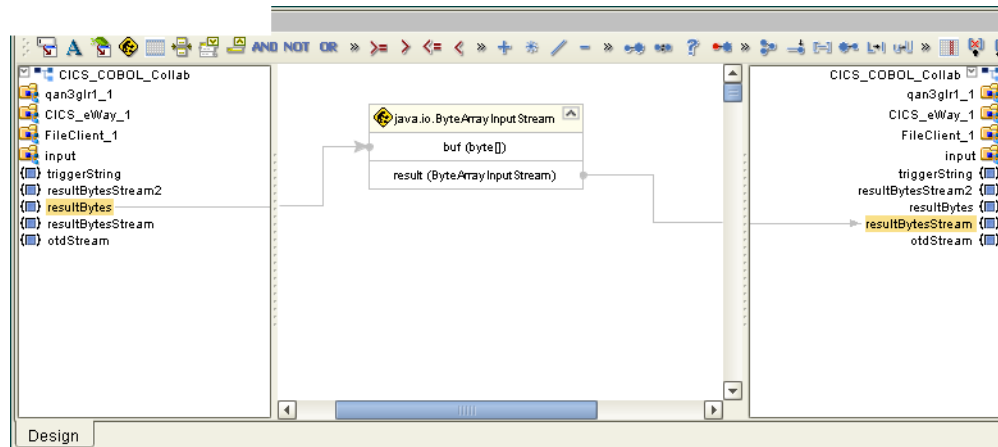
Figure 64 Copy CICS_eWay_1.CommArea to resultBytes rule



- 12 To create the **Copy new ByteArrayInputStream(resultBytes) to resultBytesStream** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Business Rules Designer toolbar, click the **Call New Constructor** icon. The **Call New Constructor** dialog box appears.
 - C Select **ByteArrayInputStream** for All Classes and **java.io.ByteArrayInputStream(byte[] buf)** as the Constructors. Click **OK**. The **java.io.ByteArrayInputStream** method box is appears on the Business Rules Designer canvas.

- D Map **resultBytes** in the left pane of the Business Rules Designer, to the **buf (Byte[])** input node of the **java.io.ByteArrayInputStream** method box.
- E Map the **result (ByteArrayInputStream)** output node of the **java.io.ByteArrayInputStream** method box.in the left pane of the Business Rules Designer, to the **buf (Byte[])** input node of the **java.io.ByteArrayInputStream** method box (see Figure 65).

Figure 65 Copy new ByteArrayInputStream(resultBytes) to resultBytesStream rule

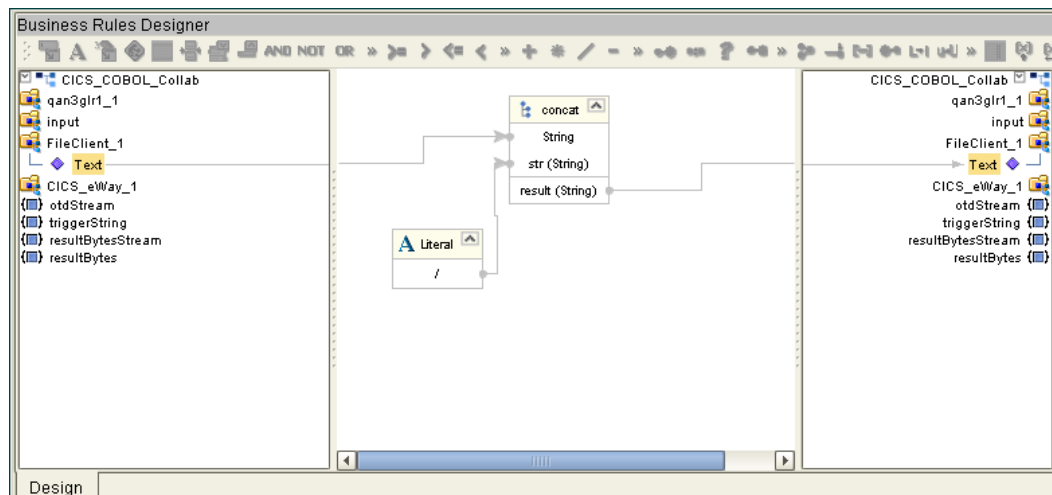


- 13 To create the **Copy new SimpleOtdInputStreamImpl(resultBytesStream) to oldStream** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Collaboration Editor toolbar click the **Advance mode** icon. The Java Source Editor now appears as the bottom pane of the Collaboration Editor.
 - C From the Collaboration Editor pane, select **//New rule**, and replace it with the following code:


```
oldStream = new com.stc.otd.runtime.provider.SimpleOtdInputStreamImpl(resultBytesStream );
```
 - D From the Collaboration Editor toolbar click the **Commit changes** icon.
- 14 To create the **qan3glr1_1.unmarshal(otdStream)** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the left pane of the Business Rules Designer, right click **qan3glr1_1** and select **Select a method to call** from the shortcut menu.
 - C From the Method selection box, select **unmarshal()**. The **unmarshal** method box appears on the Business Rules Designer canvas.
 - D Map **otdStream** from the left pane of the Business Rules Designer to the **in (OtdInputStream)** input node of the **unmarshal** method box.
- 15 To create the **Copy qan3glr1_1.DFHCOMMAREA.CA_OUT_NAME to FileClient_1.Text** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.

- B Map **CA-OUT-CITY** under **qan3glr1_1 > DFHCOMMAREA** in the left pane of the Business Rules Designer to **Text** under **FileClient_1** in the right pane of the Business Rules Designer.
- 16 To create the **Copy FileClient_1.Text.concat("/") to FileClient_1.Text** rule do the following:
- A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Business Rules Designer toolbar, click the **Create Literal** icon. The **Create Literal** dialog box appears.
 - C From the **Create Literal** dialog box, select **String** as Type, enter **/** as the Value, and click **OK**. The **Literal** method box appears on the Business Rules Designer canvas.
 - D From the Business Rules Designer toolbar, drag the **concat** icon to the Business Rules Designer canvas. The **concat** method box appears.
 - E Map the **Text** under **FileClient_1** in the left pane of the Business Rules Designer, to the String input node of the **concat** method box.
 - F Map the **/** output node of the **Literal** method box to the **str (String)** input node of the **concat** method box.
 - G Map the **result (String)** output node of the **concat** method box to **Text** under **FileClient_1** in the right pane of the Business Rules Designer (see Figure 66).

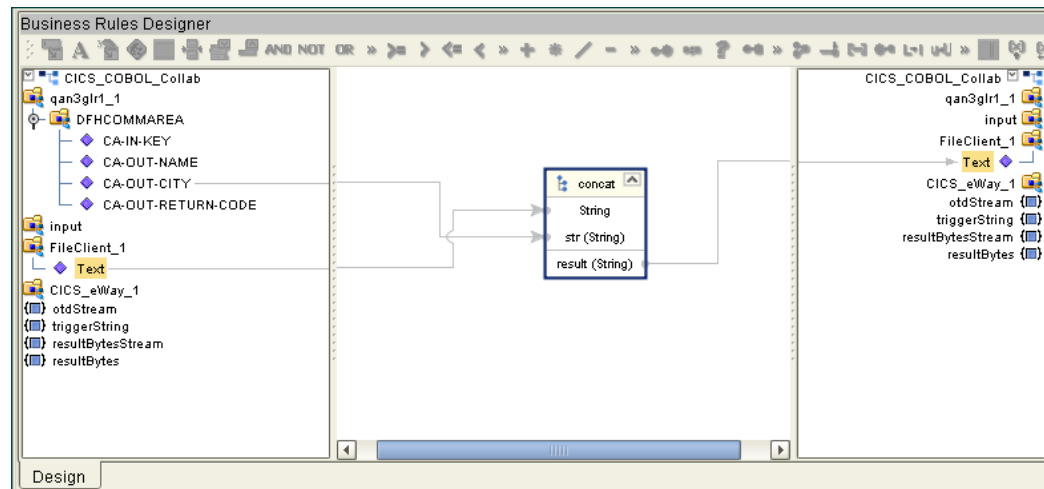
Figure 66 Copy FileClient_1.Text.concat("/") to FileClient_1.Text rule



- 17 To create the **Copy FileClient_1.Text.concat(qan3glr1_1.DFHCOMMAREA.CA_OUT_CITY) to FileClient_1.Text** rule do the following:
- A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Business Rules Designer toolbar, drag the **concat** icon to the Business Rules Designer canvas. The **concat** method box appears.
 - C Map the **Text** under **FileClient_1** in the left pane of the Business Rules Designer, to the String input node of the **concat** method box.

- D Map **CA-OUT-CITY** under **qan3glr1_1 > DFHCOMMAREA** in the left pane of the Business Rules Designer to the **str (String)** input node of the **concat** method box.
- E Map the **result (String)** output node of the **concat** method box to **Text** under **FileClient_1** in the right pane of the Business Rules Designer (see [Figure 67 on page 100](#)).

Figure 67 Copy FileClient_1.Text.concat to FileClient_1.Text rule



- 18 To create the **Copy FileClient_1.Text.concat("/")** to **FileClient_1.Text** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Business Rules Designer toolbar, click the **Create Literal** icon. The **Create Literal** dialog box appears.
 - C From the **Create Literal** dialog box, select **String** as Type, enter **/** as the Value, and click **OK**. The **Literal** method box appears on the Business Rules Designer canvas.
 - D From the Business Rules Designer toolbar, drag the **concat** icon to the Business Rules Designer canvas. The **concat** method box appears.
 - E Map the **Text** under **FileClient_1** in the left pane of the Business Rules Designer, to the **String** input node of the **concat** method box.
 - F Map the **/** output node of the **Literal** method box to the **str (String)** input node of the **concat** method box.
- 19 To create the **Copy FileClient_1.Text.concat(qan3glr1_1.DFHCOMMAREA.CA_OUT_RETURN_CODE)** to **FileClient_1.Text** rule do the following:
 - A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the Business Rules Designer toolbar, drag the **concat** icon to the Business Rules Designer canvas. The **concat** method box appears.

- C Map the **Text** under **FileClient_1** in the left pane of the Business Rules Designer, to the String input node of the **concat** method box.
 - D Map **CA-OUT-RETURN-CODE** under **qan3glr1_1 > DFHCOMMAREA** in the left pane of the Business Rules Designer to the **str (String)** input node of the **concat** method box.
 - E Map the **result (String)** output node of the **concat** method box to **Text** under **FileClient_1** in the right pane of the Business Rules Designer.
 - F Map the **result (String)** output node of the **concat** method box to **Text** under **FileClient_1** in the right pane of the Business Rules Designer.
- 20 To create the **FileClient_1.write** Business Rule do the following:
- A From the Business Rules toolbar, click the **rule** icon to add a new rule.
 - B From the left pane of the Business Rules Designer, right click **FileClient_1** and select **Select a method to call** from the shortcut menu.
 - C From the Method selection box, select **write()**. The **write** method box appears on the Business Rules Designer canvas.
- 21 From the editor's toolbar, click **Validate** to check the Collaboration for errors.
- 22 Save your current changes to the Repository.

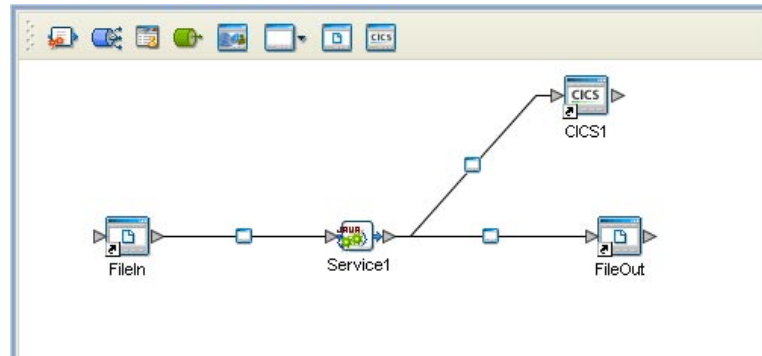
Note: See the *eGate Integrator User's Guide* for more information on editing Collaborations.

5.7.6. Creating Collaboration Bindings

After the Collaborations have been written, the components are associated and the Collaboration Bindings are created in the Connectivity Map.

- 1 From the Project Explorer, double-click the Connectivity Map **CM_CICS_COBOL_Sample**. The Enterprise Designer canvas now displays the **CM_CICS_COBOL_Sample** Connectivity Map.
- 2 Drag and drop the **CICS_COBOL_Collab** Collaboration from the Project Explorer onto **CicsCobolCollab** in the **CICS_COBOL_Sample_CM** Connectivity Map. If the Collaboration was successfully associated, the Collaboration's "gears" icon changes from red to green.
- 3 Double-click **CicsCobolCollab**. The **CicsCobolCollab** Binding dialog box appears.
- 4 From the **CicsCobolCollab** Binding dialog box, drag **FileClient input** (under Source) to the **Inbound** File external application.
- 5 From the **CicsCobolCollab** Binding dialog box, drag the **FileClient FileClient_1** OTD (under Destination) to the **outbound** File external application.
- 6 From the **CicsCobolCollab** Binding dialog box, drag the **CICS_eWay CICS_eWay_1** OTD (under Destination) to the **CICS** external application.
- 7 Minimize the **CicsCobolCollab** Binding dialog box. The Connectivity Map now appears similar to the Connectivity Map displayed in [Figure 68 on page 102](#).

Figure 68 Connectivity Map - Connecting the Project's Components



- 8 Right-click the Repository in the Project Explorer and select **Save changes to Repository** to save your changes.

5.7.7. Creating an Environment

Environments include the external systems, Logical Hosts, integration servers and message servers used by a project and contain the configuration information for these components.

- 1 From the Enterprise Designer's Enterprise Explorer, click the **Environment Explorer** tab.
- 2 Right-click the Repository and select **New Environment**. A new environment is added to the Environment Explorer tree.
- 3 Rename the new environment to **CICS_COBOL_Sample_ENV**.
- 4 Right-click **CICS_COBOL_Sample_ENV** and select **New CICS External System**. Name the External System **CicsExtSys**. Click **OK**. **CicsExtSys** is added to the Environment Editor.
- 5 Right-click **CICS_COBOL_Sample_ENV** and select **New File External System**. Name the External System **FileExtSysIn** and select **Inbound File eWay** as the **External System Type**. Click **OK**. **FileExtSysIn** is added to the Environment Editor.
- 6 Right-click **CICS_COBOL_Sample_ENV** and select **New File External System**. Name this External System **FileExtSysOut** and select **Outbound File eWay** as the **External System Type**. **FileExtSysOut** is added to the Environment Editor.
- 7 Right-click **CICS_COBOL_Sample_ENV** and select **New Logical Host**. The **LogicalHost1** box is added to the Environment and **LogicalHost1** is added to the Environment Editor tree.
- 8 Right-click **LogicalHost1** and select **New SeeBeyond Integration Server**. A new Integration Server (**IntegrationSvr1**) is added to the Environment Explorer tree under **LogicalHost1**.
- 9 Save your current changes to the repository.

5.7.8. Configuring the eWays

The CICS_COBOL_Sample project uses three eWays, each represented in the Connectivity Map as a node between an External Application and a Collaboration.

The File eWay properties are configured from the Connectivity Map. The CICS eWay properties are set from both the Project Explorer's Connectivity Map and the Environment Explorer. To configure the eWays do the following:

Configuring the File eWays

- 1 Double-click the **Inbound File eWay** and select **Inbound File eWay** in the Templates dialog box and click **OK**.
- 2 The **Properties Sheet** opens to the Inbound File eWay properties. Modify the properties for your system, including the settings for the **Inbound File eWay** in Table 6, and click **OK**.

Table 10 Inbound File eWay properties

Inbound eWay Connection Parameters	
Directory	C:/temp
Input file name	Input*.txt

- 3 In the same way, modify the **Outbound File eWay** properties for your system, including the settings in Table 7, and click **OK**.

Table 11 Outbound File eWay properties

Outbound eWay Connection Parameters	
Directory	C:/temp
Output file name	output%.dat

Configuring the CICS eWay

The CICS eWay properties must be set in both the Project Explorer and Environment Explorer. For more information on the CICS eWay properties and the **Properties Sheet**, see [Creating and Configuring the CICS eWay](#) on page 39 or see the *eGate Integrator User's Guide*.

For the CICS_JCE_Sample project, do the following:

Modify the CICS eWay Connectivity Map Properties

- 1 From the **Connectivity Map**, double-click the **CICS eWay**. The **Properties Sheet** opens to the CICS eWay Connectivity Map configuration properties.
- 2 Modify the CICS eWay Connectivity Map properties for your system, including the settings in [Table 12 on page 104](#), and click **OK**.

Table 12 CICS eWay Connectivity Map Properties

CICS eWay Connectivity Map Properties	
Connector Set as directed, otherwise use the default settings	
Class	com.stc.eways.cics.CicsClientConnector
Connection Transport	SeeBeyond CICS Listener
Type	CICS
CICS Client Set as directed, otherwise use the default settings.	
COMMAREA Length	1000
CICS Program	<i>CICS program to be run on server</i>

Modify the CICS eWay Environment Explorer Properties

- 1 From the **Environment Explorer** tree, right-click the CICS External System (**CicsExtSys** in this sample), and select **Properties**. The **Properties Sheet** opens to the CICS eWay Environment configuration .
- 2 Modify the CICS eWay Environment configuration properties for your system, including the settings in **Table 13 on page 104**, and click **OK**.

Table 13 CICS eWay Environment Explorer Properties

CICS eWay Environment Explorer Properties	
SeeBeyond CICS Listener Set as directed, otherwise use the default settings	
Host	<i>Host Name</i>
Port	<i>TCP/IP Port Number</i>
CICS Client Set as directed, otherwise use the default settings.	
CICS Password	<i>Password of CICS user</i>
CICS UserId	<i>This is the ID of the CICS user</i>

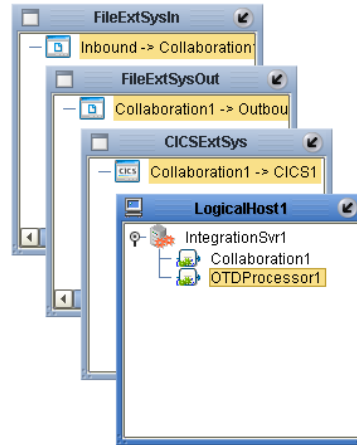
5.7.9 Creating and Activating the Deployment Profile

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

- 1 From the Enterprise Explorer’s Project Explorer, right-click the project (**CICS_COBOL_Sample**) and select **New > Deployment Profile**.
- 2 Enter a name for the Deployment Profile (for this sample **CICS_COBOL_Sample_DP**). Select **CICS_COBOL_Sample_ENV** as the Environment and click **OK**.
- 3 From the left pane of the Deployment Editor, drag the **Inbound -> Service1** (External Application) to the **FileExtSysIn** window.

- 4 From the left pane of the Deployment Editor, drag the **Service1 -> Outbound** (External Application) to the **FileExtSysOut** window.
- 5 Drag the **Service1 -> CICS1** (External Application) to the **CicsExtSys** window.
- 6 Drag **Service1** and **OTDProcessor1** to **IntegrationSvr1** in the **Logicalhost1** window (see Figure 69).

Figure 69 Deployment Profile



- 7 Click **Activate**. When activation succeeds, save the changes to the Repository.

5.7.10. Running the Project

The following directions assume that the Enterprise Designer was downloaded to C:\eGate50. If this is not the case, replace that location in the following directions with the appropriate location.

- 1 From the Enterprise Manager Downloads tab, click on **Logical Host - for win32**.
 - 2 Extract the file to the **eGate50\logicalhost3\logicalhost** directory. You must specify the **logicalhost3** directory for it to be created.
 - 3 Navigate to **C:\eGate50\logicalhost3\logicalhost\bootstrap\config** directory and open the **logical-host.properties** file using Notepad™.
 - 4 Enter the following information in the appropriate fields:
 - ♦ Logical Host root directory: **eGate50\logicalhost3\logicalhost**
 - ♦ Repository URL: **http://localhost:port number/repository name**
 - ♦ Repository user name and password: *Your user name and password*
 - ♦ Logical Host Environment name: **CICS_COBOL_Sample_ENV**
 - ♦ Logical Host name: **logicalhost3**
- Save your changes to **logical-host.properties** and close the file.
- 5 Run the **bootstrap.bat** file in the **eGate50\logicalhost3\logicalhost\bootstrap\bin** directory.
 - 6 Copy the sample input data file to the input directory.

Java Methods

A number of Java methods have been exposed to make it easier to set information in the eWay's OTD Editor and to get information from it. These methods are contained in the CicsClient Class.

Chapter Topics

- [The CicsClient Class](#) on page 106
- [Packed Decimal Java Helper Methods](#) on page 106

6.1 The CicsClient Class

The CicsClient class represents an OTD through which a Collaboration can invoke transaction programs on a CICS server. Nodes and methods are exposed so that the Collaboration can conveniently prepare a request for a CICS program, invoke the program, and get result from the program. The OTD also exposes nodes that correspond to eWay configuration parameters, to provide the means to dynamically override the preset eWay settings at runtime.

CICS Javadoc

The Javadoc is uploaded with the eWay's documentation file (**CICSeWayDocs.sar**) and downloaded from the Documentation tab of the Enterprise Manager. To access the full Javadoc, extract the Javadoc to an easily accessible folder, and double click the **index.html** file.

6.2 Packed Decimal Java Helper Methods

The Java helper methods for the toPackedDecimal class. These methods are not exposed in the Collaboration but are available for use. For example, to use the **CopyTo** method to obtain a Hex value in a byte array you would use the following code:

```
MyPacked.toPackedDecimal("327.00") ;
System.out.println( " Decimal value is : " + MyPacked.toString() ) ;
byte[] work_buf = new byte[7];
MyPacked.CopyTo(work_buf, 7);
for ( int Ii = 0 ; Ii < 7 ; Ii++ )
{
    int Ib = (int) new Byte(work_buf[Ii]).intValue();
    if (Ib < 16)
```

```
    {
      System.out.println("  Byte" + Ii + " Hexvalue = 0" + Integer.toHexString(Ib));
    }
    else
    {
      System.out.println("  Byte" + Ii + " Hexvalue = " + Integer.toHexString(Ib));
    }
  }
}
```

This produces the following output:

```
Decimal value is : 327.00
Byte0 Hexvalue = 00
Byte1 Hexvalue = 00
Byte2 Hexvalue = 00
Byte3 Hexvalue = 00
Byte4 Hexvalue = 32
Byte5 Hexvalue = 70
Byte6 Hexvalue = 0c
```

These methods are described in detail on the following pages:

[ContainerExists](#) on page 107

[GiveElem](#) on page 108

[CopyBack](#) on page 107

[SetElem](#) on page 109

[CopyTo](#) on page 108

ContainerExists

Description

Checks to see if contents of a packed decimal are available.

Syntax

```
public com.stc.eways.cics.PackedDecimal boolean ContainerExists()
```

Parameters

None.

Return Values

Boolean

Throws

None.

CopyBack

Description

Builds a packed decimal from the payload in a byte array. The byte array must contain a valid packed decimal number.

Syntax

```
public com.stc.eways.cics.PackedDecimal void CopyBack(byte Origin[],
                                                         int Size)
```

Parameters

Name	Type	Description
Origin[]	byte	Payload containing a valid packed decimal number.
Size	int	Size of the payload that contains the packed decimal number.

Return Values

None.

Throws

None.

CopyTo

Description

Copies a packed decimal number to a byte array.

Syntax

```
public com.stc.eways.cics.PackedDecimal void CopyTo(byte Dest[],
                                                    int Size)
```

Parameters

Name	Type	Description
Dest[]	byte	A byte array that the packed decimal number will be copied to.
Size	int	Size of the payload that contains the packed decimal number.

Return Values

None.

Throws

None.

GiveElem

Description

Returns a specified byte of a packed decimal number.

Syntax

```
public com.stc.eways.cics.PackedDecimal int GiveElem(int Ii)
```

Parameters

Name	Type	Description
li	int	Position of the byte to be returned.

Return Values

int
Returns the value of the position.

Throws

None.

SetElem

Description

Sets the value of a specified byte in a packed decimal number.

Syntax

```
public com.stc.eways.cics.PackedDecimal void SetElem(int Ii,  
                                                    byte Value)
```

Parameters

Name	Type	Description
li	int	Position of the byte to be set.
Value	byte	Value to set.

Return Values

None.

Throws

None.

Index

A

alerting and logging 53
 alerts/status notifications 53
 ateway 49

B

binding 82
 dialog box 65
 eWay components 64
 bindings 101
 Business Rule
 comments
 creating 78, 82

C

CEDA 16, 19
 CICS
 described 8
 overview 7
 CICS eWay
 location 40
 overview 7, 8
 properties 39
 Connectivity Map 39
 Environment Explorer properties 39
 CICS Transaction Gateway
 configuration 38
 install 38
 overview 8
 security 12
 CICSClient OTD 71
 Cobol Copybook Converter
 function described 8
 sample project 90
 Cobol Copybook Wizard 92
 Collaboration
 editor 76
 Collaboration definitions
 Java 76
 Collaboration editor
 Business Rules 77
 using 77

COMMAREA 8
 comments
 creating 78, 82
 Connectivity Map 74
 CTG
 defining a target server
 Business Process 55
 Collaboration 72
 selecting a server 55, 72
 Customer Information Control System
 described 8

D

Deployment Profile 88
 creating 68
 DFHRPL Concatenation 19
 dynamic configuration 55

E

Environment 65, 84
 creating 65
 Logical Host 65
 SeeBeyond Integration Server 66
 eWays
 creating 40
 External Application
 creating 39
 External Applications 74

H

helper methods 106

I

implementation 71

J

Java methods 106
 CicsClient Class 106
 Javadoc 106
 JMS Client
 properties 87

L

logging
 enabling and modifying 53
 levels and categories 53

M

methods

- packed decimal helper methods 106
 - ContainerExists 107
 - CopyBack 107
 - CopyTo 108
 - GiveElem 108
 - SetElem 109
- packed decimal helper methods example 109

O

operating systems

- supported 13

P

project

- creating 74
- importing 73
- JCE sample overview 73
- using Java Collaborations 71

properties

- CICS Client 43, 52
- CICS Gateway 49
- CICS Password 52
- CICS Program 44
- CICS TransId 44
- CICS UserId 53
- Class 42
- COMMAREA Length 44
- COMMAREA length 44
- COMMAREA Padding Character 46
- configuring 39
- configuring the eWays 66
- Connection Transport 43
- Connectivity Map properties 39, 42
- Dump Offset 51
- ECI call type 44
- ECI extend mode 44
- Encoding 45
- Environment Explorer
 - properties 46
- Environment Explorer properties 39
- Filename 51
- Host 47
- JMS Client 87
- KeepAlive 47
- Level 51
- Listener Timeout 47
- modifying 41
 - Connectivity Map properties 40
- NoDelay 47

Polling Rate 48

Port 48, 50

Property.Tag 43

ReceiveBufSize 48

SeeBeyond CICS Listener 46

SeeBeyond CICS Listener TransId 48

SendBufSize 48

SSL KeyRing Class 50

SSL KeyRing Password 50

Start Delay 49

Start Type 49

Timing 52

TP Timeout 49

Tracing 51

Transport Timeout 49

Truncation Size 52

Type 43

URL 50

Properties Sheet 41

R

running a project 69, 89

S

sar files

CICSeWay.sar 16

security 11

security validation

CICS Transaction Gateway 12

request start transaction mode 12

SeeBeyond Listener

business logic mode 11

request link to program mode 12

request start transaction mode 12

z/OS CICS security 11

SeeBeyond CICS Listener

configuration file maintenance screen for z/OS 20

configuration requirements 14

installation

3480 tape 18

adding the CICS eWay Load Module 19

CICS CEDA definitions 19

copying the tape contents to disk 18

verifying proper installation 19

z/OS 16

installation from CD-ROM 17

monitor screen for OS/390 25

overview 10

system requirements 14

external 14

Index

U

URL 50

W

WebLogic 15

WebSphere 15