

SeeBeyond ICAN Suite

eWay Intelligent Adapter for ADABAS Natural User's Guide

Release 5.0



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 This work is protected as an unpublished work under the copyright laws.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 20040903132757.

Contents

Chapter 1

Introducing the ADABAS Natural eWay	7
About ADABAS Natural Applications	7
About the ADABAS Natural eWay	8
Basic Operation	8
Architecture: Component Interrelations	9
Protocols	9
Functional Description	10
Natural Program Security	10
Outbound eWay Functionality: Sending Data to ADABAS	10
Failed Message Delivery Continuation	11
Logging	11
Errors	12
Alerting	12
Natural Program OTD Generator	12
What's in This Document	12
Scope	12
Intended Audience	13
Writing Conventions	13
SeeBeyond Web Site	13
SeeBeyond Documentation Feedback	13
Related Documents	14

Chapter 2

Installing the eWay	15
Supported Operating Systems	15
System Requirements	15
Supported External Applications	16
Software Requirements	16
Requirements for the CICS Server Using CICS Transaction Gateway	16
z/OS System Requirements (MVS)	16
To use CICS	17
To use Batch	17
Suggested User Requirements	18
Before You Install	18

Installing the eWay Product Files	18
Installing the ADABAS Natural eWay on an eGate Supported System	19
After You Install	19
z/OS Systems	19
ADABAS Natural eWay Installation for z/OS	20
Downloading the ADABAS Natural eWay Zip File	20
Installing the ADABAS Natural eWay from CD for z/OS	21
CICS Transaction Gateway 5.1 Installation and Configuration	22
Installation Procedures	22
Copying the Tape Contents to Disk	22
Installing the CICS CEDA Definitions	22
Linking the ADABAS Natural eWay Load Modules	23
Adding the ADABAS Natural eWay Load Modules to the CICS DFHRPL Concatenation	23
Creating the ADABAS Natural eWay Control VSAM File	23
Installing the Natural Programs	23
Configuration Procedures	23
Configuring NTSYS Setname	24
STCNCTL VSAM File Record Descriptions	31
CICS Installation Verification	33
Virtual Natural eWay Interactive Simulator	34
Virtual JCL Submit eWay Interactive Simulator	39
Virtual Natural eWay Monitoring Screens z/OS CICS	42

Chapter 3

Configuring the ADABAS Natural eWay	49
Creating and Configuring an ADABAS Natural eWay	49
Configuring the eWay Connectivity Map Properties	50
Configuring the ADABAS Natural eWay Properties	50
Connector	51
Class	51
Connection Transport	51
Property Tag	51
Type	51
CICS Client	52
ECI Call Type	52
ECI Extend Mode	52
ECI LUW Token	52
Encoding	53
Message Qualifier	53
Configuring the Environment Properties	53
CICS Gateway	54
Port	54
SSL KeyRing Class	54
SSL KeyRing Password	55
URL	55
Tracing	55

Dump Offset	56
Filename	56
Level	56
Timing	57
Truncation Size	57
Natural Settings	57
CICS Program to Execute Natural Business Rules	58
CICS Program to Initialize Natural Session	58
CICS Program to Terminate Natural Session	59
CICS Transaction to Execute Natural Business Rules	59
CICS Transaction to Initialize Natural Session	59
CICS Transaction to Terminate Natural Session	59
Natural Environment	59
Natural Transaction	59
CICS Client	60
CICS Password	60
CICS UserID	60
Adding Jar Files to the Logical Host	61
Adding Jar Files Manually	61

Chapter 4

ADABAS Natural Wizard Operation	62
Getting Started	62
Creation of .sag Files	62
Converting .sag Files: The ADABAS Natural Wizard	63
Using the ADABAS Natural Wizard	64

Chapter 5

Implementing the ADABAS Natural eWay	68
The ADABAS Natural eWay Sample Project	68
Sample Project Overview	69
Importing a Sample Project	69
Configure the File eWay	70
Creating the Environment	71
Creating the Deployment Profile	72
Configure the ADABAS Natural eWay	74
Activate the Project	75
Run the Sample	75
CICS CTG Concerns	75
Defining more Than One Server When Using CTG	75
Matching the NaturalPayload Size	76
Natural Sub-programs	76
Communication With eGate: Basic Steps	76
Within eGate	76
Within CICS on z/OS	77

Return Data	77
-------------	----

Appendix A

Error Return Codes	78
---------------------------	-----------

Error Return Codes	78
TCP/IP Error Return Codes	78
Sockets Extended Error Return Codes	88

Index	92
--------------	-----------

Introducing the ADABAS Natural eWay

This document gives a general overview of the ADABAS Natural eWay, part of the SeeBeyond Integrated Composite Application Network Suite™ (ICAN). The guide explains how to install, configure, and operate it. This guide also explains the eWay's features, as well as how to implement it in a typical eGate Integrator environment.

What's in This Chapter

This chapter includes a brief description of SeeBeyond® Technology Corporation's eWay Intelligent Adapter for ADABAS Natural, as well as system requirements for using the eWay and an introduction to this guide.

- [“About ADABAS Natural Applications” on page 7](#)
- [“About the ADABAS Natural eWay” on page 8](#)
- [“What's in This Document” on page 12](#)
- [“SeeBeyond Web Site” on page 13](#)
- [“Related Documents” on page 14](#)

1.1 About ADABAS Natural Applications

ADABAS, Software AG's database server, is used for large business applications that require sub-second response times for thousands of users, even in peak times.

Natural, Software AG's development environment, is a 4GL programming language that communicates with ADABAS. Natural is designed specifically for building applications that combine flexibility and scalability, using a logical syntax which mimics the structure of natural languages. It incorporates reusable components which are used in the development and modification of applications and can be Web-enabled using the Natural Web Interface.

Natural applications support all leading platforms (including mainframes, Windows, Unix and many others) and can be integrated seamlessly with all the major database systems (ADABAS, Tamino, DB2, Oracle, etc.).

Natural applications allow full interoperability with programs in other languages, such as C, Java, VisualBasic or COBOL. Modules developed in these languages can be embedded directly into Natural applications allowing programmers to make use of functions without having to master other languages.

1.2 About the ADABAS Natural eWay

The ADABAS Natural eWay provides an interface to Software AG's Natural language that allows for Natural programs to pass data to eGate in a reliable and efficient manner.

1.2.1 Basic Operation

The eWay client components reside on the Logical Host. The eWay server components reside on the host under CICS. The eWay supports the eGate outbound messaging mode following the publish/subscribe and request/reply scenarios.

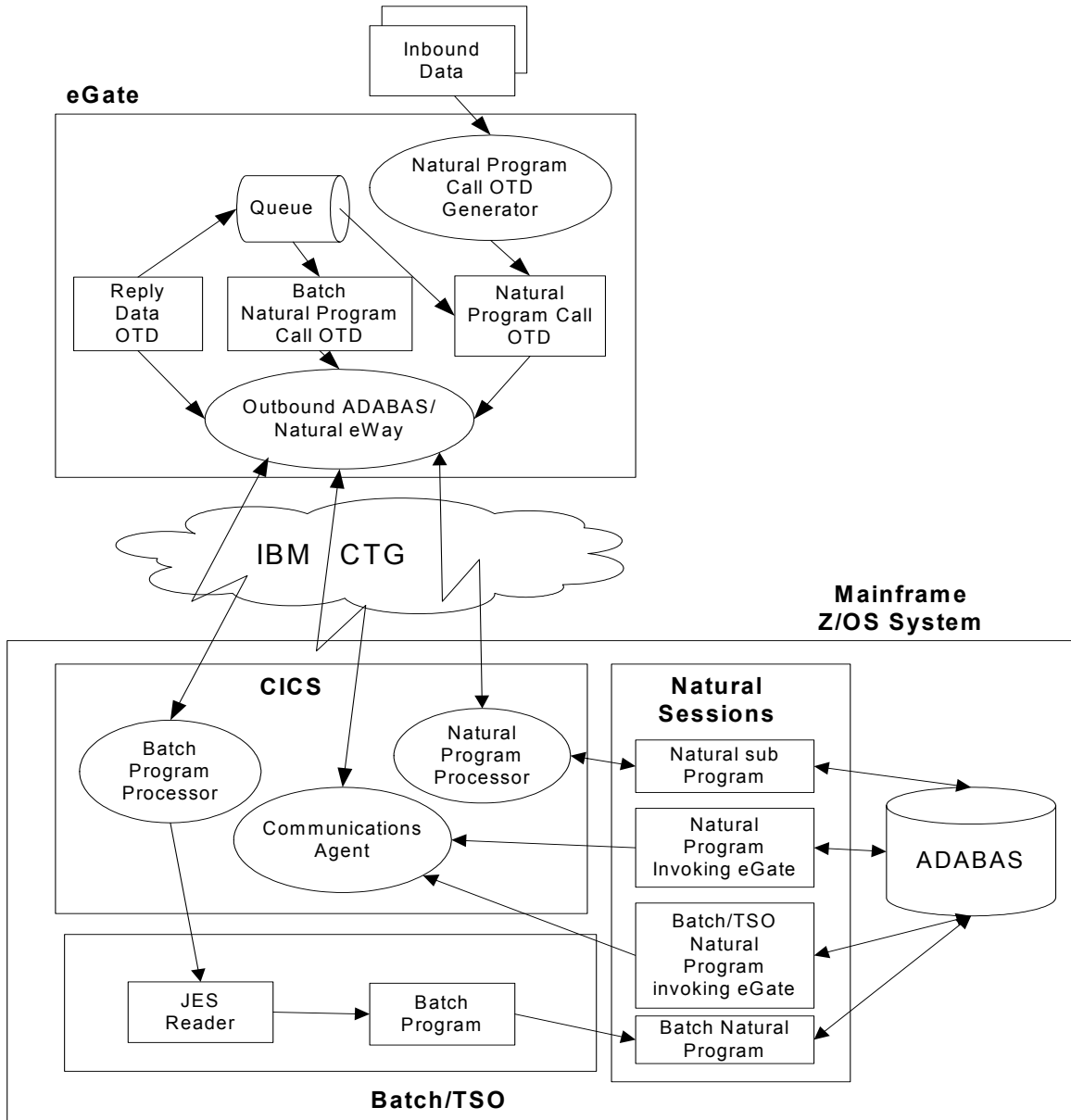
The outbound eWay calls to the Natural programs running under the control of CICS via the External Calls Interface and returns the results back to eGate. It is also able to call Natural programs running outside of CICS in the batch or TSO Batch environment. This is achieved via submission of JCL to the internal reader. Pre-defined XML (eXtensible Markup Language) templates describe the messages, which direct the eWay as to what programs to call, while providing the values of necessary parameters. This request/reply interface allows the eWay to return the result of a call to one of these programs, via eGate, to the calling application.

The Natural Program OTD Generator creates the appropriate OTD with elements for each of the specific parameters from the source code of any Natural program that must be called.

1.2.2 Architecture: Component Interrelations

Figure 1 illustrates the components involved and the interrelation of these components.

Figure 1 ADABAS Natural eWay System Architecture



1.2.3 Protocols

TCP/IP is used as the communication protocol between the COBOL client program and the eWay. The outbound eWay utilizes the transport required by IBM's Universal Client for the platform on which it is running.

The OTD library utilizes the XML standard for the structure of its messages.

1.2.4 Functional Description

The outbound eWay makes calls to Natural programs, via the External Call Interface, and returns the results back to eGate. The outbound eWay also makes calls to Natural batch programs through the execution of a batch job with parameters in the Batch or TSO environments.

The default configuration of the eWays does not require any user intervention or additional code, other than the specification of the eWay parameters via the eGate Enterprise Manager, provided the customer accepts the default behavior. This functionality is described further in the sections that follow.

Natural Program Security

The eWay supports the ability to call multiple programs. The ability to invoke these programs is controlled via the ADABAS user security facility. The user ID and password, stored in a file located on the mainframe, gives the ADABAS user access to the appropriate programs for the eWay(s). The file is read during initialization by the Natural Program Processor and the Batch Natural Program Processor.

Outbound eWay Functionality: Sending Data to ADABAS

The outbound functionality starts with the eWay reading the associated configuration file, which contains defined parameters that have been created or modified according to necessity. The configuration file contains the information that allows the eWay to communicate with multiple host environments.

Note: *Multiple instances of the eWay may be created and configured in a subscriber pool for higher throughput and reliability.*

The eWay transitions into the messaging state, where it checks the queue for a pre-defined Message Type. This Message is defined in one of the pre-built Object Type Definitions shipped with the eWay (Natural Program OTD or Natural Batch Program OTD).

Once the eWay receives a message, it determines the type of request: Natural program or Batch Natural. The eWay assembles the various parameters contained in the standard Natural/Batch Natural Program OTD into the appropriate ECI call.

The eWay proceeds by making the ECI call and processes the result. If the call is successful, the eWay checks the reply data flag for the transaction. If the flag is set, the eWay returns the result in the format defined by the reply data OTD. After this operation, or in the situation that the flag is not set, the eWay commits the Message in the queue.

Natural Program Processor

When the Message is determined to be designated for a Natural program, it is passed to a CICS transaction, which invokes the Natural Program Processor, and load-balances these requests across a user-configurable number of Natural sessions. Upon completion of the Natural program, the processor returns the output of the program (if any exists) and the return code to the eWay.

Batch Program Processor

When the Message is determined to be designated for a batch program, it is passed to a CICS transaction, which invokes the Batch Program Processor. This transaction processes the Message and calls the appropriate TSO or batch program. The eWay then waits on a successful response from the submission process.

If the "Send" is unsuccessful due to communication problems, the eWay attempts a specified number of times to call the transaction with the supplied parameters. If the eWay is unsuccessful after the configured number of times, it shuts down and sends an alert to eGate. No other transactions can be successful with a communication problem in existence.

If the error was non-communication related, the eWay attempts a configurable number of times to call the transaction with the supplied parameters. If the eWay remains unsuccessful, the eWay performs the action defined by the "Failed Message Delivery Parameter," causing it to either write the message with the result of the call pre-appended to the error queue and get the next message, or shut down the eWay for manual intervention.

If the reply data flag is set, the eWay sends the error back to the configured reply queue, continuing to the next message without shutting down.

If the eWay receives a shutdown message from eGate, it sends an alert and follows the standard shutdown procedures.

Failed Message Delivery Continuation

The eWay can be configured to either skip a message that cannot be delivered due to a non-communication related problem, or shut down. The skipping of the message is performed after the message has been safely stored in an error queue. The message then has the result of the call pre-appended to it before storage.

The eWay provides a parameter representing the number of Messages that can be skipped before the eWay shuts down. This parameter allows for the following values:

- 0 = no limit to the number of skipped messages
- 1 = shut down the eWay on the first failed message
- >1 = the actual number of skipped messages to be tolerated

Logging

In general, the eWay relays any pertinent information as to the state, protocol position, and any conditions that are helpful to the user to understand what is taking place. The eWay then either logs the information to a file or notifies the Alert Agent (according to the debug level settings).

Standard debug levels are set through the eDesigner.

Errors

Any error conditions are written to the log file. The inability to write to the log file or any fatal/unrecoverable errors result in the eWay shutting down after it sends an alert to the Alert Agent.

Alerting

Any errors that affect the operation of the eWay preventing the successful delivery of a message cause an alert to be sent. If the alert cannot be sent, the eWay shuts down.

Natural Program OTD Generator

The Natural Program OTD Generator creates the appropriate OTD containing the elements for each of the specific parameters required by the Natural program from the associated source code. This is achieved by the program reading the appropriate Natural Local Data (LDA) or Parameter Data Access (PDA) contained in a Natural SYSTRANS file.

1.3 What's in This Document

This document includes the following chapters:

- **Chapter 1 “Introducing the ADABAS Natural eWay”:** Provides an overview description of the product as well as high-level information about this document.
- **Chapter 2 “Installing the eWay”:** Describes the system requirements and provides instructions for installing the ADABAS Natural eWay.
- **Chapter 3 “Configuring the ADABAS Natural eWay”:** Provides instructions for configuring the eWay to communicate with your legacy systems.
- **Chapter 4 “ADABAS Natural Wizard Operation”:** Provides information about .sag files and using the ADABAS Natural wizard.
- **Chapter 5 “Implementing the ADABAS Natural eWay”:** Provides instructions for installing and running the sample Projects.

1.3.1 Scope

This document describes the process of installing, configuring, and running the ADABAS Natural eWay.

This document does not cover the Java methods exposed by this eWay. For information on the Java methods, download and view the ADABAS Natural eWay Javadoc files from the Enterprise Manager.

1.3.2 Intended Audience

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

1.3.3 Writing Conventions

The following writing conventions are observed throughout this document.

Table 1 Writing Conventions

Text	Convention	Example
Button, file, icon, parameter, variable, method, menu, and object names.	Bold text	<ul style="list-style-type: none">Click OK to save and close.From the File menu, select Exit.Select the logicalhost.exe file.Enter the timeout value.Use the getClassname() method.Configure the Inbound File eWay.
Command line arguments and code samples	Fixed font. Variables are shown in bold italic .	<code>bootstrap -p password</code>
Hypertext links	Blue text	http://www.seebeyond.com

1.4 SeeBeyond Web Site

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

<http://www.seebeyond.com>

SeeBeyond Documentation Feedback

If you have any feedback on any SeeBeyond documentation, please send an E-mail to:

docfeedback@seebeyond.com

1.5 Related Documents

The following SeeBeyond documents are designed to work in conjunction with the *eWay Intelligent Adapter for ADABAS Natural User's Guide* and to provide additional information:

- *SeeBeyond ICAN Suite Installation Guide*
- *eGate Integrator User's Guide*
- *eGate Integrator JMS Reference Guide*
- *Readme.txt* file on the eGate installation CD-ROM

Installing the eWay

This chapter explains procedures for installing the ADABAS Natural eWay and the CICS Transaction Gateway (CTG).

What's in This Chapter

- [“Supported Operating Systems” on page 15](#)
- [“System Requirements” on page 15](#)
- [“Supported External Applications” on page 16](#)
- [“Installing the eWay Product Files” on page 18](#)
- [“z/OS Systems” on page 19](#)
- [“CICS Installation Verification” on page 33](#)

2.1 Supported Operating Systems

The ADABAS Natural eWay is available on the following operating systems:

- Windows 2000, Windows XP, and Windows 2003
- HP-UX 11.0 and 11i (PA-RISC)
- Solaris 8 and 9

2.2 System Requirements

The system requirements for the ADABAS Natural eWay are the same as those for eGate Integrator. For more 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 ADABAS Natural 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 Supported External Applications

This section explains the ADABAS Natural eWay's supported external applications, including requirements.

2.3.1 Software Requirements

The ADABAS Natural eWay software requirements include:

- **IBM CICS Transaction Server** version 1.3, 2.1, or 2.2
- **IBM CICS Transaction Gateway 5.1** (see [Requirements for the CICS Server Using CICS Transaction Gateway](#) on page 16)
- **ADABAS Natural eWay**

2.3.2 Requirements for the CICS Server Using CICS Transaction Gateway

IBM's Customer Information Control System (CICS) provides connectivity and online transaction management for mission-critical applications. It is used to access many file systems and databases including third party products. CICS Transaction Gateway 5.1 is required for Java-enabled ADABAS Natural eWays. The following describes how to configure CICS Transaction Gateway 5.1. You can set the Transaction Gateway properties using the CTG Configuration Tool, which is located in the CICS Transaction Gateway program menu.

Note: *Use of the TCP/IP protocols CICS for MVS/ESA Version 4.1 and Transaction Server for z/OS Version 1.x can only be achieved via the TCP62 protocol. For more information, refer to the IBM documentation for your specific CICS implementation.*

For system-specific settings consult the CICS Transaction Gateway Documentation or visit the IBM CICS Library Web site at the following URL:

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

Consult this documentation for information on configuring CTG for use with RACF and the latest APARs.

Set the CICS Transaction Gateway properties using the CTG Configuration Tool, located under the CICS Transaction Gateway program menu.

2.3.3 z/OS System Requirements (MVS)

z/OS (MVS) systems use the EBCDIC character set. As a consequence, ASCII-based systems cannot directly transport data to an EBCDIC-based system. ASCII to EBCDIC data conversion is necessary when data is sent from UNIX or Windows z/OS (MVS). This data conversion takes place within the eGate Collaboration.

When you communicate with an z/OS (MVS) system, you need the following requirements:

- An eGate Logical Host

Server:

- z/OS or equivalent hardware
- Physical access to a CD-ROM
- TCP/IP connectivity
- Appropriate terminal for access to the system

To enable the eWay to communicate correctly with ADABAS Natural, you need the following external requirements:

- z/OS
- ADABAS version 7.1.2
- Natural Language version 3.1.3
- RACF or equivalent security
- CICS Transaction Server version 1.3 or higher
- COBOL for z/OS and Language Environment

To use CICS

To enable the eWay to communicate properly with the server system, you need the following requirements:

- COBOL for z/OS compiler available for use in the z/OS Language Environment (LE), with the CICS TCP/IP socket elements available

The following link to the IP CICS Sockets manual describes the setup procedures:

<http://www-1.ibm.com/servers/s390/os390/bkserv/r10pdf/secureway.html>

- z/OS
- Security package, install script RACF, ready
- CICS Transaction Server version 1.3
- CICS TCP/IP socket interface
- COBOL for z/OS
- Optional: Open Multiple Virtual System (OMVS) installed, configured, and operational

To use Batch

To enable the eWay to communicate properly with the server system, you need the following requirements:

- COBOL for z/OS compiler available for use in the z/OS Language Environment (LE), with the MVS TCP/IP socket elements available

For additional information, consult the IBM web site, document number SG24-5229-01, "z/OS eNetwork Communications Server TCP/IP Implementation Guide, Volume 3: MVS Applications."

- ◆ z/OS
- ◆ Security package - RACF
- ◆ Install script
- ◆ MVS TCP/IP socket interface must be installed, configured, and operational
- ◆ COBOL for z/OS
- ◆ Optional: Open Multiple Virtual System (OMVS) installed, configured, and operational

2.3.4 Suggested User Requirements

The following users can install and configure the ADABAS Natural eWay:

- MVS/CICS system administrator and systems programmer
- RACF or equivalent security administrator
- Dedicated integration specialist to learn and operate eGate
- Lead developer/system administrator of applications
- Natural/ADABAS DBA

2.4 Before You Install

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

2.5 Installing the eWay Product Files

During the eGate Integrator installation process, use the Enterprise Manager, a web-based application, 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, install eGate and the eWays using the Enterprise Manager from a computer running Windows, connected to the Repository server.

2.5.1 Installing the ADABAS Natural eWay on an eGate Supported System

The ADABAS Natural 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 After you upload the eGate.sar file, when you are uploading files to the eGate Repository (using the Enterprise Manager), select and upload the following files:
 - ♦ **ADABASNaturaleWay.sar** (to install the ADABAS Natural eWay)
 - ♦ **FileeWay.sar** (to install the File eWay, used in the sample project)
 - ♦ **ADABASNaturaleWayDocs.sar** (to download the ADABAS Natural eWay user's guide, sample projects, and Javadoc)
- 2 Continue installation as instructed in the *SeeBeyond ICAN Suite Installation Guide*

2.6 After You Install

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

2.7 z/OS Systems

This section explains how to install the ADABAS Natural eWay on z/OS systems (any reference to z/OS in this document includes OS/390 V2R10). The installation tape contains the datasets listed in Table 2.

Table 2 Installation Tape Datasets

Dataset Name	Contents
TAPE.STC.RESTORE.JCL	Physical Sequential Datasets containing the JCL for this tape.
TAPE.STC.NATURAL.JCLLIB	Partition Dataset that contains installation jobs and control cards for the eWay.

Table 2 Installation Tape Datasets (Continued)

Dataset Name	Contents
TAPE.STC.NATURAL.CICLOAD	Load Library that contains the CICS load modules for the eWay.
TAPE.STC.NATURAL.UNLOAD	Natural unload dataset containing a Natural object for the eWay.

2.7.1 ADABAS Natural eWay Installation for z/OS

The following section provides directions for installing the ADABAS Natural eWay 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 ADABAS Natural eWay monitoring screen to verify that all components are properly installed and working correctly.

Downloading the ADABAS Natural eWay Zip File

The ADABAS Natural eWay ZIP file, `ADABASNatural_eWay.zip`, can be downloaded from the ICAN Installation CD-ROMs. To download the `ADABASNatural_eWay.zip` to your computer do the following:

- 1 Locate the Products installation CD-ROM containing the `ADABASNaturaleWay.sar` file.
- 2 Open the CD-ROM to view and locate the `ADABASNaturaleWay.sar` file.
- 3 Right-click the `ADABASNaturaleWay.sar` file, and from the shortcut menu, select the "Open With" option and select a zip program such as WinZip™.

From the opened `ADABASNaturaleWay.sar` file, click on and drag the `ADABASNatural_eWay.zip` file to your Desktop. Close the zip program. The `ADABASNatural_eWay.zip` file is now available from your computer's Desktop file.

Installing the ADABAS Natural eWay from CD for z/OS

The following describes how to restore the ADABAS Natural eWay files from the **ADABASNatural_eWay.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 turns a PDS into FB 80 files that can be sent by FTP. Files are downloaded to a computer and then compressed with PKZIP.

- 1 Download the **ADABASNatural_eWay.zip** file to your PC, and unzip the files using a zip program such as WinZip.
- 2 Create three MVS datasets to receive the files, as follows:

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

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

//DD3      DD DSN=USER.XMIT.NATURAL.UNLOAD,
//          DISP=(NEW,CATLG,DELETE),
//          RECFM=FB,LRECL=80,BLKSIZE=3120,DSORG=PS,
//          SPACE=(3120,(30,5)),
//          UNIT=diskunit
```

- 3 Upload the unzipped files (FTP, IND\$FILE) to MVS, using a binary file transfer method (no CRLF or ASCII translation).
- 4 Restore the files to PDS by using the **Receive** command on MVS.
- 5 Issue command: TSO RECEIVE INDATASET(uploaded.dataset)
- 6 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 here in case your installation requires them.

We suggest using the following names for your received datasets:

```
STC.XMIT.NATURAL.CICSLOAD
STC.XMIT.NATURAL.JCLLIB
STC.XMIT.NATURAL.UNLOAD
```

These names match our documentation. Change them to match your facility requirements.

2.7.2 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.

2.7.3 Installation Procedures

This section explains the eWay's basic installation procedures for z/OS.

Copying the Tape Contents to Disk

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

```
//JOB CARD
//IEBGENER EXEC PGM=IEBGENER
//SYSSPRINT DD SYSOUT=*
//*
//*COPY NATURAL E*WAY NATURAL UNLOAD (INPL) TO DISK
//*
//SYSUT1 DD DSN=TAPE.STC.RESTORE.JCL,DISP=OLD,UNIT=TAPE,
//          VOL=(,RETAIN,SER=STC390),LABEL=(1,SL)
//SYSUT2 DD DSN=customers.pds(restore),DISP=SHR
//SYSIN DD DUMMY
//
```

- 2 Customize and submit the RESTORE job to copy the entire contents of the Installation tape to disk.

Installing the CICS CEDA Definitions

Customize and submit the job STCNCEDA to create the CICS CEDA definitions for the ADABAS Natural eWay and installation verification programs.

A PCT entry needs to be set up for transid K26Z. This PCT entry needs to be created exactly the same as the PCT entry that already exists at your site for your implementation of the Natural Nucleus running in CICS. The only difference in this new PCT is that the transid name is K26Z.

A PPT entry needs to be created for the Software AG-provided program NCIXCALL. See the Software AG documentation for details on creating this PPT entry. NCIXCALL is required for the 3GL external-call interface. Therefore, please refer to the 3GL sections in the Software AG documentation.

Linking the ADABAS Natural eWay Load Modules

Customize and submit the job STCNLINK to link-edit all ADABAS Natural eWay modules and installation verification programs with the LE, TCP/IP, CICS, and Natural interface modules.

Adding the ADABAS Natural eWay Load Modules to the CICS DFHRPL Concatenation

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

- // DD DSN=&PREFIX..STC.NATURAL.CICSLOAD,DISP=SHR
- The library containing the load module NCIXCALL must be added to the DFHRPL concatenation in the CICS region that is running the SeeBeyond ADABAS Natural eWay z/OS components.

Creating the ADABAS Natural eWay Control VSAM File

Customize and submit the job STCNCTL to allocate and initialize the ADABAS Natural eWay VSAM control file.

Customize the DSNAME parameter in PDS member CEDANCTL to match the VSAM file that was previously allocated by job STCNCTL.

For session ID records, you need to change the ETID and create a new ETID in the natural environment, for these natural sessions.

Customize and submit job STCNFCT to create CICS FCT definitions for the ADABAS Natural eWay VSAM Control file.

Installing the Natural Programs

Customize and submit job STCNLOAD to load the Natural programs into your Natural FUSER file.

Use the Natural SYSMAN utility to copy the following Natural modules to the SYSTEM library on both the FUSER and FNAT files:

```
STCNBEC  
STCNERR  
STCNLOG  
STCNROL
```

2.7.4 Configuration Procedures

This section explains the eWay's post-installation procedures for z/OS.

Configuring NTSYS Setname

There must be at least one Natural NTSYS SETNAME ID defined using the recommended values shown below. If the customer wants to define more than one configuration of the values shown below, there are a few values that may be tailored for optimizing performance/throughput and a few different NTSYS SETNAME IDs may be defined for that purpose. These NTSYS SETNAME IDs are then referred to and used by the ADABAS Natural eWays by specifying them in the number Configuration record types in the STCNCTL VSAM file.

The following values are required and cannot be changed in the NTSYS SETNAME definition:

ADAMODE=2

All time-out situations (NAT3009) received for nucleus calls are automatically handled by Natural and do not lead to an error message displayed to the user.

ADAMODE: ADABAS Interface Mode

This parameter controls the number of ADABAS user queue elements (UQE):

- Adaplex support with two ADABAS UQEs per Natural session (as with Natural Version 2.3), or
- Adaplex support with one ADABAS UQE per Natural session, or
- No Adaplex support (as with Natural version 2.2).

The possible values are:

ADAMODE=0

Start Natural in Natural Version 2.2 mode. The UQE is built by the ADALN_x module. All database calls, either sent by the nucleus, an application program or a 3-GL program, are considered as the same ADABAS user. Running under SYSPLEX is not possible.

ADAMODE=1

Start Natural with one user and ADABAS X48 communication. Only one UQE is initialized while all nucleus and application database calls are submitted for the same UQE, however, calls sent by 3-GL programs are excluded. Running under SYSPLEX is possible.

ADAMODE=2

Start Natural in Natural Version 2.3 mode (if supported by ADALN_x). Two UQEs are generated at Natural session startup with nucleus and application calls running separate from each other. Database calls sent by 3-GL programs are excluded from Natural transactions. Running under SYSPLEX is possible.

AUTO=OFF

This must be set to OFF to allow the Natural eWay to log on under a different user ID. If this is set to ON, the Natural eWay always logs on with the ACEE of CICS (which is usually the CICS job name).

AUTO

Automatic log-on.

AUTO=ON

An automatic log-on is executed at the start of the Natural session. The value contained in the system variable INIT-USER is used as the user ID for the log-on.

Note: *If used with Natural Security, AUTO=ON disables log-ons with another user ID (see your Natural Security documentation for further information).*

AUTO=OFF

No automatic log-on is performed.

CDYNAM=5

CDYNAM: Dynamic Loading of Non-Natural Programs

This parameter determines how many non-Natural programs can be loaded dynamically by Natural during the execution of a single Natural program.

The value specified with the CDYNAM parameter determines the maximum number of non-Natural programs which can be loaded per Natural program.

The current default for this parameter is 5. If CDYNAM=0, the call from STCNBEC to STCNBEP fails and no dynamic loading of non-Natural programs is performed by Natural.

CM=ON

This must be set to ON to allow the STACK TOP COMMAND LOGON command to change Natural libraries. Setting this parameter to OFF causes the Natural session to terminate if the Natural eWay attempts to log on to another Natural library.

CM: Command Mode

This parameter can be used to suppress the Natural command mode (NEXT and MORE).

CM=ON

NEXT and MORE are available for command input.

CM=OFF

The Natural session is terminated whenever NEXT is encountered; the MORE line is write-protected (no input is possible).

DU=OFF

This must be set to OFF to prevent Natural from coming down hard. If this parameter is set to ON, an application error such as S0C7 would force the Natural session to terminate.

DU: Dump Generation

This parameter indicates whether a memory dump is to be generated in the case of an abnormal termination during the Natural session.

DU=ON

A memory dump is produced in the case of an abnormal termination (TP-monitor dump dataset or SYSUDUMP in z/OS batch mode or TSO), followed by the Natural session terminating with error message NAT9974.

DU=SNAP

Forces an immediate dump in the case of an abnormal termination during a Natural session. The Natural session continues after the dump has been taken.

DU=FORCE

Forces an immediate dump in the case of an abnormal termination during a Natural session and terminates the Natural session immediately. This is useful for testing purposes in some environments.

DU=OFF

No memory dump is produced. In batch mode, subsequent action taken by Natural is determined by the setting of the CC profile parameter. In online mode, Natural responds with errors NAT0954, NAT0955 or NAT0956. For further information on the abnormal termination, you can use the system command DUMP (see the Natural Reference documentation).

DYNPARAM=ON

This parameter must be set to ON to allow various dynamic overrides.

DYNPARAM: Control Use of Dynamic Parameters

This parameter can only be specified dynamically and can be used only once. It controls the use of dynamic profile parameters. It corresponds to the NTDYNP macro in the parameter module.

DYNPARAM=ON

All profile parameters can be specified dynamically.

DYNPARAM=OFF

No profile parameters can be specified dynamically.

DYNPARAM=(ON,parameter-name, ...)

Only those parameters listed here can be specified dynamically. Other parameters cause Error Message NAT7008 to be issued.

DYNPARAM=(OFF,parameter-name ,...)

All profile parameters can be specified dynamically, except those listed here. These parameters cause Error Message NAT7008 to be issued.

ETID

This parameter is currently being generated by the ADABAS Natural eWay and contains the current session number. Setting the value to a blank in a Special Link, under Natural security, avoids the response code 48 problem with ADABAS. However, in this case, if an eWay is using up too much ADABAS resources, it becomes very difficult to identify which eWay is responsible.

ETID: ADABAS User ID

The value specified is used as the user ID value in an ADABAS open call. This parameter is used as an identifier for ADABAS-related information; for example, for identification of data stored as a result of an END TRANSACTION statement.

If the value specified with the ETID parameter is blank, Natural does not issue any ADABAS open and close commands; the OPRB parameter (if specified) and any ETID and OPRB specifications in Natural security are ignored.

In this case, you are recommended to set the Natural profile parameter DBCLOSE to **ON** so the close command at session end is enforced. Otherwise, the user is not logged off from ADABAS and the ADABAS user queue element is not deleted. This could cause an overflow situation in the ADABAS user queue.

If the value specified with the ETID parameter does not equal the value of the Natural system variable *INIT-USER, Natural issues an ADABAS open command with the specified ETID value (and OPRB value, if specified) at the beginning of the Natural session; this open command remains in effect until the end of the Natural session; any ETID and OPRB specifications in Natural Security are ignored.

If the value specified with the ETID parameter is the same as the value of *INIT-USER, or if the ETID parameter is not specified, Natural issues an ADABAS open command with the *INIT-USER value as ETID (and the OPRB value, if specified) at the beginning of the Natural session. If any Natural Security log-on (initial log-on or any subsequent log-on) change occurs to the currently valid ETID or OPRB value (due to the library- or user-specific ETID and OPRB specifications in Natural Security), Natural Security issues a new open command with the new ETID and OPRB values. If the values remain the same after a log-on, Natural Security does not issue a new open command.

If ETID=OFF, Natural does not issue any ADABAS open and close commands at the beginning of the Natural session. If, however, any ETID and/or OPRB specifications are present in Natural Security, these specifications are used in the subsequent open command issued by Natural Security.

This parameter value is provided for use in conjunction with Natural Security to prevent Natural batch jobs that are sent at the same time from causing duplicate user ID values in an ADABAS open call during the initialization phase.

ETID and *INIT-USER can be modified by user exit NATUEX1 during session startup. See User Exit for Authorization Control - NATUEX1 (in the Natural Operations for Mainframes documentation).

ID=,

This parameter must be set to the default of a "," (comma). The ADABAS Natural eWay currently uses this delimiter character when building the STACK TOP COMMAND to log on to a differently library.

ID: Input Delimiter Character

This parameter defines the character to be used as a delimiter character for INPUT statements.

If the input delimiter character is to be a comma, it must be specified as ID=', '.

The character specified with this parameter must not be the same as the one specified with the DC parameter (decimal character) or IA parameter (input assign character); it must not be the same as the one specified with the CF parameter (control character for terminal commands) or HI parameter (help character).

Do not use a period (.) as an input delimiter, because this symbol may lead to situations in which a program termination period would be misinterpreted as an input delimiter. Also, do not use an asterisk (*).

Within a Natural session, the profile parameter ID can be overridden by the session parameter ID.

MENU=OFF

This parameter must be set to OFF to avoid any startup programs from being invoked when a Natural session starts.

MENU: Menu Mode

This parameter is used to switch Natural menu mode on or off.

MENU=OFF

Disables menu mode. Within a Natural session, the MENU parameter can be overridden by the system command MAINMENU (described in the Natural user's guide for mainframes).

OUTDEST=CSSL

This parameter must be set to CSSL in NTSYS. It is very useful in debugging any Natural run-time errors that may have occurred.

OUTDEST: Output Destination for Asynchronous Processing

This parameter only applies to Natural under CICS, Complete and UTM. It specifies the destination to which any Natural error message produced by an asynchronous application is to be sent. After an error message has been sent, Natural terminates the asynchronous session.

Under UTM, this parameter is used to specify the ID of the terminal where output from an asynchronous application is to be displayed. When and how error messages/output from an asynchronous application are output depends on the respective TP monitor.

SENDER=CSSL

This parameter must also be set to CSSL in NTSYS. It is very useful for debugging any Natural run-time errors that may have occurred.

SENDER: Screen Output Destination for Asynchronous Processing

This parameter only applies under CICS, Complete, IMS/TM and UTM. It specifies the destination where output from an asynchronous application is to be displayed. The destination specified with the SENDER parameter applies to hardcopy output and primary reports.

Any additional reports are sent to the destinations specified with the DEFINE PRINTER statement (just as in a synchronous online session). Platform-specific characteristics of the SENDER parameter are listed below.

CICS: The SENDER parameter specifies the CICS transient data (TD) destination and the terminal or printer for terminal output from asynchronous sessions. If the specified destination does not exist, the session output is sent to the specified terminal or printer. If the specified terminal or printer does not exist either, the session terminates abnormally.

The default terminal output format for asynchronous sessions is a 3270 data stream. If the SENDER terminal specification is not a 3270 device, the Natural application must switch to the correct terminal type before the first output statement. For example, by specifying SET CONTROL 'T=PRNT' for a printer or by starting with profile parameter TTYPE=PRNT.

If you are routing all output to a (spool) destination, such as CSSL, the Natural application must be switched to line mode. For example by specifying SET CONTROL 'T=BTCH' or by starting with profile parameter TTYPE=BTCH. In this case, two other profile parameters are relevant: EJ and INTENS.

If you set EJ=ON, all lines are routed with a leading ASA control character.

With EJ=OFF, there is no leading ASA control character. INTENS must be set to "1", particularly if you have set EJ=OFF

Optional Values

These parameters are tailored specifically for the processing of different incoming transactions from the external ADABAS Natural eWays. They can be used to customize certain Natural session configurations for the purpose of separating long running inquiries against ADABAS from short lived transactions that normally require few ADABAS commands. The STCNCTL VSAM file is where incoming transactions are assigned to specific Natural session Configurations.

The following optional values can be changed in the NTSYS SETNAME definition.

LE=OFF

This parameter must be set to OFF. Setting this parameter to ON can cause the Natural session to terminate if any LIMIT conditions have been coded in a called Natural program.

This parameter may need to be coded to ON if persistent long running resource intensive statements (READ and FIND) cause undue stress on the Natural environment.

LE: Reaction when Limit for Processing Loop Exceeded

This parameter controls the action to be taken if the limit specified for processing-loop execution is exceeded.

LE=ON

Loop execution is aborted and an error message is issued at the end of the Natural program.

LE=OFF

Loop execution is aborted and processing continues without an error message.

LT=99999999

This parameter must be set to the default of 99999999 to avoid any Natural time-outs during any long running ADABAS command(s).

LT: Limit for Processing Loops

This parameter limits the number of records which can be read in processing loops within a Natural program. The limit specified with this parameter applies to loops initiated with a READ, FIND or HISTOGRAM statements only.

All records read in these loops (including rejected records from a WHERE clause) are counted against this limit. Within a Natural session, the profile parameter LT can be overridden by the session parameter LT.

MADIO=0

This parameter must be set to 0 to avoid any Natural time-outs during one or more long-running ADABAS commands.

MADIO: Maximum DBMS Calls between Screen I/O Operations

This parameter indicates the maximum number of DBMS calls permitted between two screen input-output operations (also in the batch mode). If the specified limit is exceeded, the Natural program is interrupted, and the user is notified by the Natural Error Message 1009.

MADIO=0

Indicates that no limit is in effect.

MAXCL=0

This parameter must be set to 0 to avoid any Natural time-outs when calling a large number of Natural programs.

MAXCL: Maximum Number of Program Calls

This parameter determines the maximum number of program calls permitted between two screen input-output operations. If the specified limit is exceeded, the Natural program is interrupted and the user receives Error Message NAT1029.

MAXCL=0

Indicates that no limit is to be in effect.

MT=0

This parameter applies to Batch and TSO only. It must be set to 0 to avoid any Natural time-outs during any long running Natural Batch processes.

MT: Maximum CPU Time

This parameter only applies to programs executing in batch mode or under TSO. The limit for programs operating in interactive mode is controlled by the TP monitor in use. The MT parameter determines the maximum amount of CPU time which can be used by a Natural program. The value is specified in seconds.

MT=0

Indicates that no Natural CPU time limit is in effect. The maximum value that can be used is determined by the operating system environment. Any value in excess of the maximum is reduced to the maximum supported by the operating system.

In system environments which do not support CPU time measurement, the limit is interpreted as elapsed time. The CPU time limit is ignored for systems without timer support.

Within a Natural session, the profile parameter MT can be overridden by the session parameter MT.

STCNCTL VSAM File Record Descriptions

The STCNCTL file is a keyed VSAM file (KSDS). It contains the control information necessary for properly processing incoming Business Rule transaction messages from external ADABAS Natural eWays.

There must be one STCNCTL file defined for each CICS region that is executing the eWay's z/OS components.

The first two bytes of the file key represent the environment. This enables multiple environments (e.g., development, test, and production) to run concurrently in a single CICS region. This two byte value is customer assigned and must correspond to the environment contained in the messages sent from the external ADABAS Natural eWays within the CICS region.

Within each environment, there are three record types: global, session configuration, and transaction type (TranType).

The global record controls the ETIDs, user IDs, and passwords for logging onto back-end Natural sessions in the CICS region.

The session configuration record controls the NTSYS SETNAME, Natural Nucleus name, various time-out values, and other data related to executing back-end Natural sessions in the CICS region. For example, you can configure one NTSYS SETNAME for long-running inquiry transactions and a different NTSYS SETNAME for short-lived transactions.

Transaction Type Record

The Transaction Type record (TranType) controls the processing of incoming Business Rules transaction messages, coming in from the external ADABAS Natural eWays. It contains the session configuration ID number to specify the type of back-end Natural session in which to route the incoming message. The Transaction Type record also contains:

- The name of the Natural application program that processes the request
- The Natural Library that contains the Natural application program
- Various time-out values
- Data to control the processing of the Business Rule transaction message

Global Record

Table 3 explains the fields in the global record.

Table 3 Global Record Fields

Field Name	Position	Type	Length
key-environment	1	char	2
key-rec-type (this field must be 'G')	3	char	1
key-filler (this field must be blank)	4	char	4

Table 3 Global Record Fields (Continued)

Field Name	Position	Type	Length
nat-logon-timeout-secs	8	numeric	8
use-etid-mask-flag 'Y' = use the logon-etid-mask-name to create a unique ETID blank = use the logon-etid-full-name	9	char	1
logon-etid-full-name	10	char	8
logon-etid-mask-name	10	char	8
logon-etid-mask-value	10	char	8
logon-etid-mask-nnn	10	char	5
logon-userid-flag 'G' = get the nat-logon-userid and password from the global record. 'S' = get the nat-logon-user ID and password from the session configuration record.	18	char	1
nat-logon-userid	19	char	8
nat-logon-password	27	char	8
Filler	35	char	46

Session Configuration Record

Table 4 explains the fields in the session configuration record.

Table 4 Session Configuration Record Fields

Field Name	Position	Type	Length
key-environment	1	char	2
key-rec-type (this field must be 'S')	3	char	1
key-session-config-id (this field must be 01, 02, 03, etc.)	4	numeric	2
key-filler (this field must be blanks)	6	char	2
nat-nucleus-pgm (for CICS only) (get this name from your Natural systems programmer)	8	numeric	8
natu-logon-userid	16	char	8
nat-logon-password	24	char	8
ntsys-setname-id (get these names from your Natural systems programmer)	32	char	8
max-sessions (the maximum number of Natural sessions that can be executing concurrently for this session configuration ID)	40	numeric	2

Table 4 Session Configuration Record Fields (Continued)

Field Name	Position	Type	Length
curr-sessions (set this to 00. The system increments this value in memory, <i>not</i> currently executing for this session configuration ID.)	42	numeric	2
avail-session-timeout (how long to wait for an available back-end Natural session to which to attach for processing an incoming Business Rules request)	44	numeric	8
nat-session-up-timeout	52	numeric	8
Filler	60	char	21

Transaction Type Record

Table 5 explains the fields in the transaction type record.

Table 5 Transaction Type Record Fields

Field Name	Position	Type	Length
key-environment	1	char	2
key-rec-type (this field must be 'T')	3	char	1
key-tran-type (this must be the same value that the external Natural eWay is sending in on the incoming Business Rules transaction message)	4	char	4
natural-program	8	char	8
natural-library	8	char	8
session-config-id (this is the session configuration ID that is used for routing this incoming Business Rules request to the appropriate back-end Natural session in the CICS region)	16	numeric	2
timeout-secs (this determines how long to wait for the back-end Natural session to process this Business Rules request)	18	numeric	8
Filler	26	char	55

2.8 CICS Installation Verification

The simulator virtual eWays explained in this section allow you to perform virtual tests on the ADABAS Natural eWay to verify that the CICS installation has been successful and that eWay installation can be continued.

2.8.1 Virtual Natural eWay Interactive Simulator

After the components have been installed on CICS, use the Virtual Natural (VN) eWay in CICS to verify that all of the components are properly installed and working correctly.

To verify the eWay installation and operation using the Virtual Natural eWay

- 1 Sign onto the CICS region, from a blank screen, by entering the following Transaction ID:
QANE
- 2 Press ENTER to display the screen shown in Figure 2.

Figure 2 VN eWay: Welcome Screen

```
                Welcome to
                Virtual Natural e-Way
                Interactive Simulator

                To exit.....hit PF3, PF12, or the CLEAR key

                Hit PF5 to process Initialization requests
                Hit PF6 to process Business Rules requests
                Hit PF7 to process Termination requests
```

- 3 Press PF5 to process Initialization requests.
- 4 Press ENTER to confirm the request, which displays the Call Done screen.
- 5 Press ENTER to display the Initialization screen (shown below in Figure 3).

Figure 3 VN eWay: Initialization Screen

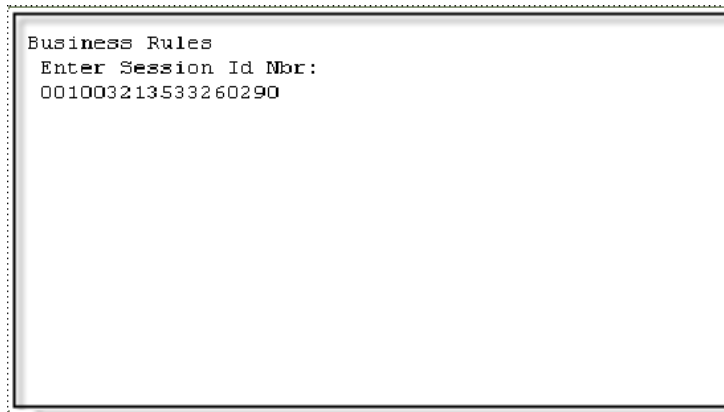
```
Initialization
return code:
0000
reason code:
0000
reason text:

session id nbr:
001003213533260290

Hit PF5 to process Initialization requests
Hit PF6 to process Business Rules requests
Hit PF7 to process Termination requests
```

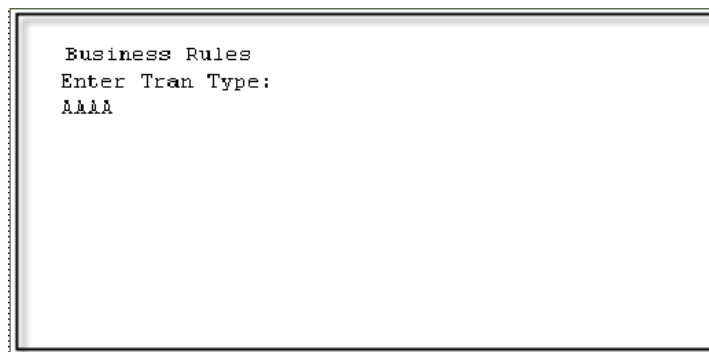
- 6 While the Initialization screen is displayed, use the mouse to highlight the eighteen-digit number directly under **session id nbr**. This is your session ID.
- 7 Copy and paste this number into a Notepad (or similar text editor) screen. You must keep this screen open for later use in this test script.
- 8 Press PF6 to display the Business Rules request screen.
- 9 Paste the number from the clipboard onto the Business Rules request screen. The screen should appear as shown in Figure 4.

Figure 4 VN eWay: Business Rules - Session ID



- 10 Press ENTER to continue to the Transaction Type.
- 11 Enter a valid transaction type that has been stored into the STCNCTL VSAM file. The screen should appear as shown in Figure 5.

Figure 5 VN eWay: Business Rules - Transaction Type



- 12 Press ENTER to continue to the Environment.
- 13 Enter a valid Environment that has been stored into the STCNCTL VSAM file. The screen should appear as shown in Figure 6.

Figure 6 VN eWay: Business Rules - Environment

```
Business Rules
Enter environment:
QA
```

- 14 Press ENTER to continue to the GUID.
- 15 Enter a valid globally unique identifier (GUID) value (any 1-to-38-character string that you have not used for this test before) as shown below in Figure 7.

Figure 7 VN eWay: Business Rules - GUID

```
Business Rules
Enter Guid:
123
```

- 16 Press ENTER to continue to the Payload.
- 17 Enter in the number 1776 as shown in Figure 8.

Figure 8 VN eWay: Business Rules - Payload

```
Business Rules
Enter Payload:
1776
```

- 18 Press ENTER to display the Call Business Rules screen (shown in Figure 9).

Figure 9 VN eWay: Call Business Rules Screen

```
About to call Business Rules....  
Session Id Nbr:  
001003213533260290  
Tran Type:  
AAAA  
Environment:  
QA  
Guid:  
123  
Payload:  
1776  
  
Hit Enter to confirm.....
```

- 19 Press ENTER to confirm the Business Rule information. The Call Done screen is displayed.
- 20 Press ENTER to display the Business Rules Return screen (shown in Figure 10).

Figure 10 VN eWay: Return Screen

```
Business Rules  
return code:  
0000  
reason code:  
0000  
reason text:  
  
data buffer(1:80):  
1776
```

- 21 Press PF7 to display the Termination screen.
- 22 Copy the session ID number from the clipboard file you opened earlier. Paste it into the Termination screen as shown in Figure 11.

Figure 11 VN eWay: Termination

```
Termination
Enter Session Id Nbr:
001003213533260290
```

- 23 Press ENTER to display the Termination Environment screen.
- 24 Enter the same environment name that you entered earlier in this script, as shown in Figure 12.

Figure 12 VN eWay: Termination Environment

```
Termination
Enter Environment:
QA
```

- 25 Press ENTER to display the Call Termination screen shown in Figure 13.

Figure 13 VN eWay: Call Termination

```
About to call Termination....
Session Id Nbr:
001003213533260290
Environment:
QA

Hit Enter to confirm.....
```

- 26 Press ENTER to confirm the termination information. The Call Done screen is displayed.
- 27 Press ENTER to display the Termination Return screen (shown in Figure 14).

Figure 14 VN eWay: Termination Return

```
Termination
return code:
0000
reason code:
0000
reason text:

Hit PF5 to process Initialization requests
Hit PF6 to process Business Rules requests
Hit PF7 to process Termination requests
```

- 28 Press CLEAR to exit the Virtual Natural eWay.

Note: You can press PF5, PF6, or PF7 to continue testing with additional test scenarios.

2.8.2 Virtual JCL Submit eWay Interactive Simulator

After the components are installed in CICS, use the Virtual JCL Submit eWay in CICS to verify that all components are properly installed and working correctly.

To verify the eWay installation and operation using the Virtual JCL Submit eWay

- 1 Sign onto the CICS region, from a blank screen, by entering the following Transaction ID:
NACJ
- 2 Press ENTER to display the screen shown in Figure 15.

Figure 15 JCL: Welcome Screen

```
Welcome to
Virtual JCL Submit e-Way
Interactive Simulator

To exit.....hit PF3, PF12, or the CLEAR key
Hit PF5 to process JCL Submit requests
```

- 3 Press PF5 to display the screen shown in Figure 16.

Figure 16 JCL: PDS Name Screen 1

```
JCL Submitter
Enter PDS name:
```

- 4 Enter your own test PDS name as shown in Figure 17.

Figure 17 JCL: PDS Name Screen 2

```
JCL Submitter
Enter PDS name:
JEFFB.PDS.CNTL
```

- 5 Press ENTER to display the screen shown in Figure 18.

Figure 18 JCL: Member Name Screen 1

```
JCL Submitter
Enter Member name:
```


- 6 Enter your own test JCL member name as shown in Figure 19.

Figure 19 JCL: Member Name Screen 2

```
JCL Submitter
Enter Member name:
TEST
```

- 7 Press ENTER to display the screen shown in Figure 20.

Figure 20 JCL: About To Call Screen

```
About to call JCL Submitter.....
PDS:
JEFFB.PDS.CNTL
Member:
TEST

Hit Enter to confirm.....
```

- 8 Press ENTER to display the screen shown in Figure 21.

Figure 21 JCL: Call Done Screen

```
STCCJCL call done...
Resp=00000000
Resp2=00000000

Hit enter to continue
```

- 9 Press ENTER to display the screen shown in Figure 22.

Figure 22 JCL: Return Screen

```
JCL Submitter
return code:
0000
reason code:
0000
reason text:
Job TEST      successfully submitted.  JCL card count = 0024.

Hit PF5 to process JCL Submit requests
```

- 10 Press CLEAR to exit the Virtual JCL eWay.

Note: You can press PF5, PF3, or PF12 to continue testing with additional test scenarios.

2.8.3 Virtual Natural eWay Monitoring Screens z/OS CICS

After the components are installed in CICS, use the Natural eWay Monitoring screens in CICS to verify that all the components are properly installed and working correctly.

To verify the eWay installation and operation using the Virtual Natural eWay Monitoring screens in CICS

- 1 Sign on to the CICS region, from a blank screen, by entering the Transaction ID shown in Figure 23.

Figure 23 JCL: Sign-on Screen



- 2 Press ENTER to display the screen shown in Figure 24.

Figure 24 CICS: Welcome Screen

```
Welcome to
Natural e-Way
Display Storage Areas

F4:GCB F5:EEAT F6:EEA F17:SEAT F18:SEA F19:Comb F20:CTLA PF3,PF12,CLR:exit
```

- 3 Press PF4 to display the Global Control Block screen (shown below).

Figure 25 CICS: Global Control Block Screen

```
STCNGCB - Global Control Block

NGCB-initial-abstime      003212926695030      10-24-01 15:38:15.030
NGCB-pointers
NGCB-ecat-pointer (dec)   249958400
NGCB-seat-pointer (dec)   249963520
NGCB-ctla-pointer (dec)   000907328
NGCB-tupt-pointer (dec)   249968640

F4:GCB F5:EEAT F6:EEA F17:SEAT F18:SEA F19:Comb F20:CTLA PF3,PF12,CLR:exit
```

The Global Control Block (GCB) screen displays the date and time it was created. It also displays the pointers to the internal control blocks, residing in the memory of CICS, that control the execution of the z/OS components of the Natural eWay.

- Press PF5 to display the e way ECB Areas screen (shown below).

Figure 26 CICS: eWay ECB Area Table Screen 1

STCNEEAT - E-Way ECB Area Table		
NEEAT-eea-status-flag	NEEAT-eea-pointer	Eway Session Id Nbr
E Exists	250000624	001003213533260290
E Exists	250014624	002003213001273370
E Exists	250020624	003003212998483090
E Exists	250079104	004003212998483740
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	
Never used	000000000	

F4:GCB F5:EEAT F6:EEA F17:SEAT F18:SEA F19:Comb F20:CTLA PF3,PF12,CLR:exit

This instance of the eWay ECB Area Table (EEAT) displays:

- ◆ The table of status flags
- ◆ Pointers to the eWay ECB Areas (EEAs)
- ◆ Current session ID numbers assigned to each eWay.

EEAs are created as each eWay is initialized and connects to the CICS region. EEAs are not deleted when an eWay terminates, therefore this screen only shows the presence of EEAs that have been created, but not the current state of each eWay.

- To view the EEA of each eWay in order to know the status, press PF6. The eWay ECB Areas screen appears (shown below).

Figure 27 CICS: eWay ECB Area Table Screen 2

STCNEEA - E-Way ECB Areas										
Status	date	time	BEP-ECB	P Tot	rc0	rc-1	rc-2	a-t	l-t	b-t
1 available	10-31-01	16:40:38	000000000	0001	0000	0000	0000	000	000	000
1 available	10-25-01	12:25:47	000000000	0001	0001	0000	0000	000	000	000
2 initialized	10-25-01	11:34:43	000000000	0000	0000	0000	0000	000	000	000
2 initialized	10-25-01	11:34:44	000000000	0000	0000	0000	0000	000	000	000

F4:GCB F5:EEAT F6:EEA F17:SEAT F18:SEA F19:Comb F20:CTLA PF3,PF12,CLR:exit

This instance of the e way ECB Areas screen displays information about each eWay ECB Area (EEA) that has been created in the memory of CICS. Each line displays on EEA. The status indicates whether the EEA is currently associated with an external Natural eWay.

When the status shows **available**, the EEA is not currently being used. It is available for use by the Natural eWay which initializes and connects to the CICS region. The remaining fields on this screen are:

- ◆ **Date:** Date of the last update to this EEA.
- ◆ **Time:** Time of the last update to this EEA.
- ◆ **BEP-ECB:** Event Control Block (ECB) used for communicating with the Natural session Back End Program (BEP).
- ◆ **P:** Previous status of this EEA, if the current status is invalid.
- ◆ **Tot:** Count of Business Rules message blocks that have been sent in from the external ADABAS Natural eWay.
- ◆ **RC0:** Number of times the business request was processed in CICS with a return code 0 (ok) back to the ADABAS Natural eWay.
- ◆ **RC-1:** Number of times the business request was processed in CICS with a return code -1 (error) back to the eWay.
- ◆ **RC-2:** Number of times the business request was processed in CICS with a return code -2 (shutdown) back to the eWay.
- ◆ **a-t:** Number of times a time-out occurred scanning for an available Natural session.
- ◆ **l-t:** Number of times a time-out occurred logging on to a new Natural session.
- ◆ **b-t:** Number of times a time-out occurred while waiting for the Natural session Back End Program (BEP) to process a Business Rule request.

There is also a set of control blocks for each back-end Natural session that is executing.

- 6 Press PF17 (SHIFT-F5) to display the Session ECB Area Table screen (shown below).

Figure 28 CICS: Session ECB Area Table Screen 1

```

          STCNSEAT - Session ECB Area Table
                NSEAT-sea-status-flag       NSEAT-sea-pointer
                      E                      250026624
                      E                      250052864
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
                      000000000
          F4:GCB F5:EEAT F6:EEA F17:SEAT F18:SEA F19:Comb F20:CTLA PF3,PF12,CLR:exit
    
```

The Session ECB Area Table screen displays the table of status flags and pointers to the Session ECB Areas (SEAs) assigned to each back-end Natural session. SEAs are created when each back-end Natural session is started in the CICS region.

SEAs are not deleted when a back-end Natural session is terminated. Therefore, this screen only shows the presence of SEAs that have been created, but not the current state of each back-end Natural session.

- 7 To view the SEA of each back-end Natural session, to ascertain its status, press PF18 (SHIFT-F6). The Session ECB Area Table screen appears (shown below).

Figure 29 CICS: Session ECB Area Table Screen 2

```

          STCNSEA - Session ECB Areas
  -----Status-----   date      time      FER-ECB      Task  Own Cur T En SC Cnt
  A available            10-25-01 12:26:36.030 0000000000  0001553 001 001  D1 01 0001
  A available            10-25-01 12:25:47.030 0000000000  0001553 002 002  D1 02 0001
          F4:GCB F5:EEAT F6:EEA F17:SEAT F18:SEA F19:Comb F20:CTLA PF3,PF12,CLR:exit
    
```

The Session ECB Area Table screen displays information about each Natural session ECB Area (SEA) that has been created in memory in CICS. Each line displays one SEA. The status indicates whether the SEA is currently associated with a back-end Natural session.

When the status shows **available**, the SEA is not currently being used. It is available for use the next time a new Natural session needs to be started in CICS. The remaining fields on this screen are:

- ◆ **Date:** Date of the last update to this SEA.
 - ◆ **Time:** Time of the last update to this SEA.
 - ◆ **FER-ECB:** Event Control Block (ECB) used for communicating with the Natural eWay Front End Business Rules Program (FER).
 - ◆ **Task:** The internal CICS task number of this Natural session. This is useful in associating a given Natural session to its task information which is displayed in the CEMT INQ TASK command screen
 - ◆ **Own:** The session ID number of the external ADABAS Natural eWay that started this back-end Natural session in CICS.
 - ◆ **Cur:** The session ID number of the most recent ADABAS Natural eWay that used this back-end Natural session to process a Business Rule request.
 - ◆ **T:** Time-out flag set when the Front End Business Rules (FER) program timed out waiting for the Natural session Back End Program (BEP) to process a Business Rules request.
 - ◆ **En:** The environment for which this Natural session is executing Business Rules requests.
 - ◆ **SC:** The session configuration ID number of this Natural session. The session Configuration is stored in the STCNCTL VSAM file.
 - ◆ **Cnt:** The number of Business Rules requests that have been processed by this back-end Natural session.
- 8 To see the external ADABAS Natural eWays and the back-end Natural sessions at the same time, press PF19 (SHIFT-F7). The eWay and Session ECB Areas screen in appears (shown below).

Figure 30 CICS: eWay and Session ECB Areas Screen

STCNEEA - E-Way ECB Areas											
—Status—	date	time	BEP-ECB	P Tot	rc0	rc-1	rc-2	a-t	l-t	b-t	
1 available	10-31-01	16:40:38	000000000	0001	0000	0000	0000	000	000	000	
1 available	10-25-01	12:25:47	000000000	0001	0001	0000	0000	000	000	000	
2 initialized	10-25-01	11:34:43	000000000	0000	0000	0000	0000	000	000	000	
2 initialized	10-25-01	11:34:44	000000000	0000	0000	0000	0000	000	000	000	

STCNSEA - Session ECB Areas											
—Status—	date	time	FER-ECB	Task	Own	Cur	T	En	SC	Cnt	
A available	10-25-01	12:26:36.030	000000000	0001553	001	001		D1	01	0001	
A available	10-25-01	12:25:47.030	000000000	0001553	002	002		D1	02	0001	

F4:GCB F5:EEAT F6:EEA F17:SEAT F10:SEA F19:Comb F20:CTLA PF3,PF12,CLR:exit

The eWay ECB Area and Session screen displays both external ADABAS Natural eWay ECB Areas (EEAs) and back-end Natural session ECB Areas (SEAs). See the previous screen's (Figure 29 on page 46) descriptions for the meaning of each field on the screen.

- 9 To view the internal memory area that contains the STCNCTL VSAM file records, press PF20 (SHIFT-F8). The Control File Area screen appears (shown below).

Figure 31 CICS: Control File Area Screen

STCNCTLA - Control File Area			
R	Ptr	Cnt	Record Area
G	000907361	0001	D1G 00000030YSTCNA002GBOTHWJG BOTHWJG
S	000907441	0002	D1S01 N3C999C5BOTHWJG BOTHWJG SB1 030000000060
T	000907601	0003	D1T1020QAN3GLR1INTFDEV101000000060
Record Image Area			
D1G	00000030YSTCNA002GBOTHWJG BOTHWJG		
D1S01	N3C999C5BOTHWJG BOTHWJG SB1		03000000006000000060
D1S02	N3C999C5BOTHWJG BOTHWJG SB1		02010000006000000060
D1T1020QAN3GLR1	INTFDEV101000000060		
D1T1030QAN3GLU1	INTFDEV101000000060		
D1T1031QAN3GLR1	INTFDEV102000000060		

F4:GCB F5:EEAT F6:EEA F17:SEAT F18:SEA F19:Comb F20:CTLA PF3,PF12,CLR:exit

The Control File Area screen displays the internal memory area (STCNCTLA) that contains the records of the STCNCTL VSAM file. These records control the processing of the Business Rules transaction messages, coming from the external ADABAS Natural eWays, and the start up (or log on) of back-end Natural sessions in CICS.

Note: See the documentation of the STCNCTL VSAM file for a complete explanation of these record layouts.

Configuring the ADABAS Natural eWay

This chapter describes how to set the properties of the ADABAS Natural eWay.

What's in This Chapter

- [Creating and Configuring an ADABAS Natural eWay](#) on page 49
- [Configuring the eWay Connectivity Map Properties](#) on page 50
- [Configuring the ADABAS Natural eWay Properties](#) on page 50
- [Configuring the Environment Properties](#) on page 53

3.1 Creating and Configuring an ADABAS Natural eWay

All eWays contain a unique set of default configuration parameters. After the eWays are established and an ADABAS Natural External System is created in the Project's Environment, the eWay parameters are modified for your specific system. The ADABAS Natural eWay configuration parameters are modified from two locations:

- From the **Connectivity Map**—which contains parameters specific to the ADABAS Natural eWay, and may vary from other eWays (of the same type) in the Project.
- From the **Environment Explorer** tree—which contains global parameters that commonly apply to all eWays (of the same type) in the Project. Saved parameters are shared by all eWays in the ADABAS Natural External System Properties window.

Note: *You must set configuration parameters for the ADABAS Natural eWay in both locations.*

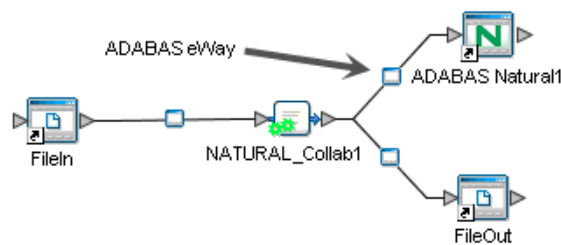
3.2 Configuring the eWay Connectivity Map Properties

When you connect an External Application to a Collaboration, Enterprise Designer automatically assigns the appropriate eWay to the link. Each eWay is supplied with a template containing default configuration properties that are accessible on the Connectivity Map.

To configure the eWay properties:

- 1 On the Enterprise Designer's Connectivity Map (see Figure 32), double-click the ADABAS Natural eWay icon. The Properties window appears.

Figure 32 Connectivity Map with Components

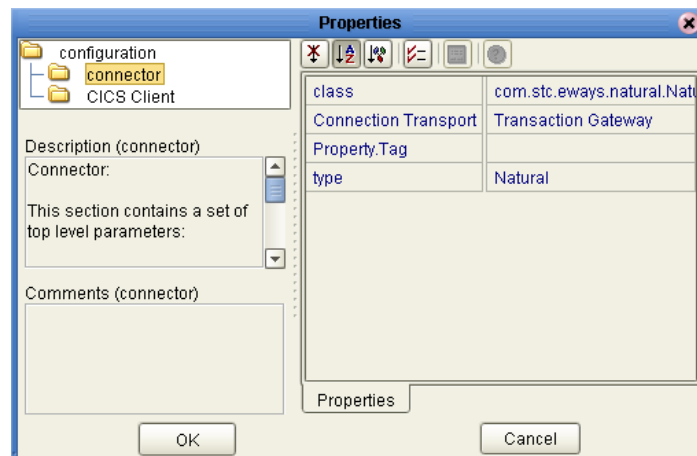


- 2 The Configuration properties window opens, displaying the default properties for the eWay.

3.3 Configuring the ADABAS Natural eWay Properties

The ADABAS Natural eWay Properties include parameters used by the external database.

Figure 33 ADABAS eWay Properties



3.3.1. Connector

This section contains the following set of top-level parameters.

- **Class**
- **Connection Transport**
- **Property Tag**
- **Type**

Class

Description

Specifies the class name of the ADABAS Natural client connector objects.

Required Values

A valid package name. The default is `com.stc.eways.natural.NaturalClientConnector`.

Connection Transport

Description

Specifies the underlying transport used to send requests to and receive responses from CICS.

Required Value

A valid connection transport name in string format. The default name is the Transaction Gateway

Property Tag

Description

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

Required Value

A valid data source package name. Accept the default.

Type

Description

Specifies the type of eWay Connection.

Required Value

A string. The value always defaults to `Natural` for ADABAS Natural connections.

3.3.2. CICS Client

This section contains the following set of top-level parameters used for setting the CICS client.

- ECI Call Type
- ECI Extend Mode
- ECI LUW Token
- Encoding
- Message Qualifier

ECI Call Type

Description

Specifies the ECI synchronous call type. Synchronous calls wait for the transaction to complete, then return the contents of the COMMAREA. Only the synchronous call type is 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. Yes indicates that the work unit is terminated at the end of the call.

Required Value

Yes or **No**. The default is **No**.

ECI LUW Token

Description

CTG specific. Specifies the logical unit of work to which a call belongs. This must be set to zero at the start of a logical work unit. The ECI updates the value on the first (or only) call of the logical work unit. If the unit of work is to be extended, this value is used as input to all subsequent calls associated with the same logical work unit.

If the return code is not ECI_NO_ERROR and a call is ending or continuing an existing logical work unit, then this field reports the state of the logical work unit. If it is zero, the logical work unit ends and updates are backed out. If it is not zero, the value is the same as the input value. The logical work unit is continuing, and updates are still pending.

Required Value

A valid integer in the range of **0** to **1000**. The configured default is **0**. This is a required input and output parameter.

Encoding

Description

Specifies the canonical name of the encoding type to be used.

Required Values

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** or **Cp500** (EBCDIC). The default is ASCII.

Message Qualifier

Description

CTG specific. Specifies the ECI Message Qualifier identifies each asynchronous call if more than one call is made. Use this security feature-related ID only on the same JavaGateway that created or assigned them.

Required Value

A valid integer in the range of 0 to 1000. This is an optional input parameter.

3.4 Configuring the Environment Properties

The eWay Environment Configuration properties contain parameters that define how the eWay connects to and interacts with other eGate components within the Environment.

To Configure the Environment Properties:

- 1 In Enterprise Designer, click the Environment Explorer tab.
- 2 Expand the Environment created for the ADABAS Natural Project and locate the ADABAS Natural External System.

Note: For more information on creating an Environment, see the eGate Integrator Tutorial.

- 3 Right-click the External System created for the ADABAS Natural Project and select Properties from the list box. The Environment Configuration Properties window appears.
- 4 Click on any folder to display the default configuration properties for that section.
- 5 Click on any property field to make it editable.
- 6 After modifying the configuration properties, click **OK** to save the changes.

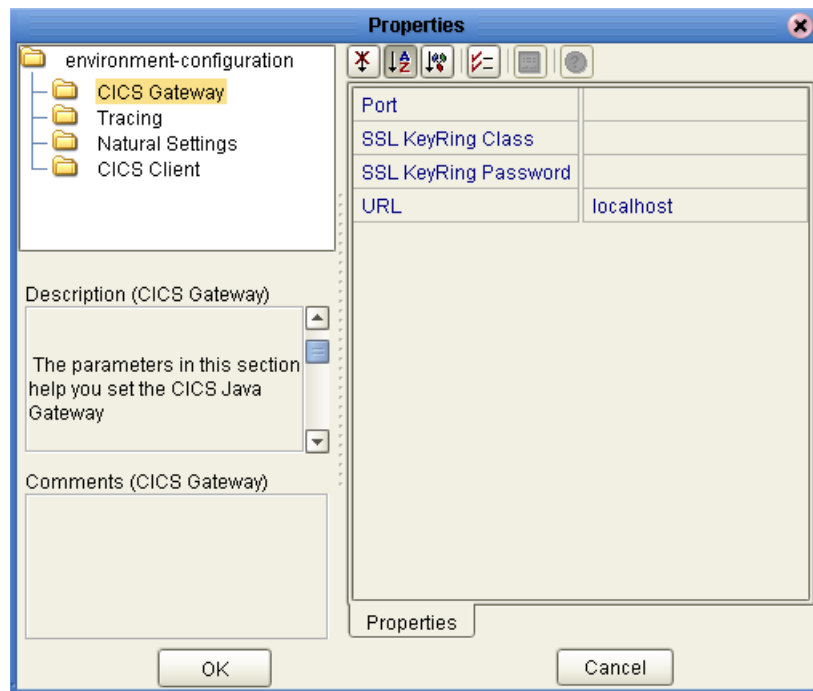
Before deploying your eWay, you must set the Environment properties. The following sections describe the properties used by the ADABAS Natural eWay Environment.

3.4.1. CICS Gateway

This section contains the following set of top-level parameters used for setting the CICS Gateway.

- Port
- SSL KeyRing Class
- SSL KeyRing Password
- URL

Figure 34 ADABAS Natural eWay Environment Configuration - CICS Gateway



Port

Description

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

Required Values

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

SSL KeyRing Class

Description

Specifies the full classname of the SSL KeyRing.

Required Value

A valid class name of the SSL KeyRing.

SSL KeyRing Password

Description

Specifies the password for the encrypted KeyRing class.

Required Value

A valid password for the encrypted KeyRing class.

URL

Description

Specifies the URL for the remote or local Gateway with which to connect.

Required Value

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

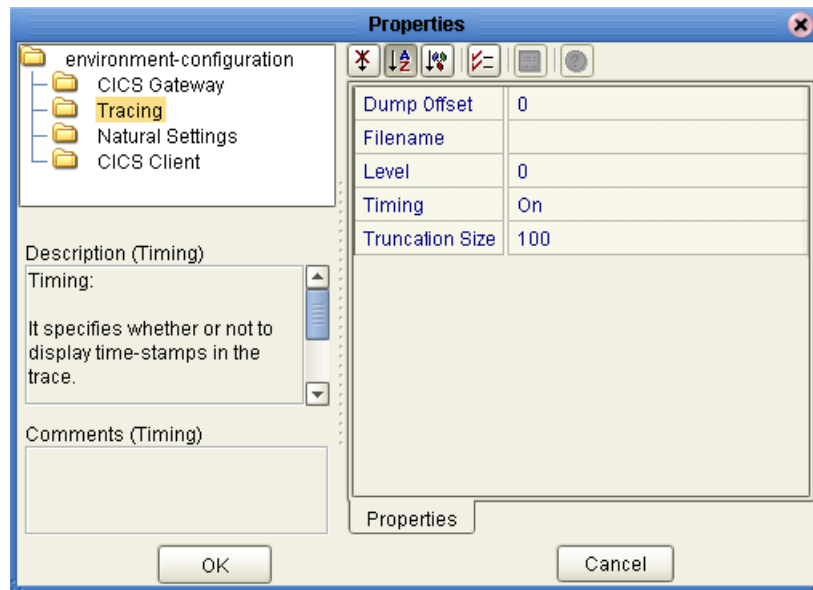
Important: This parameter requires specific jar files when utilizing **local:** as a value. For directions on installing the required jar files, see [“Adding Jar Files to the Logical Host” on page 61](#).

3.4.2. Tracing

This section contains the following top-level parameters used for setting the Tracing.

- Dump Offset
- Filename
- Level
- Timing
- Truncation Size

Figure 35 ADABAS Natural eWay Environment Configuration - Tracing



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.

A filename can also be set using the system property `gateway.T.setTFile=xxx`, where `xxx` is a file name.

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 be set using the system property **gateway.T=On**.
- **3: Exception Stacks.** Traces most Java exceptions, including exceptions 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 be set using the system property **gateway.T.stack=on**.

Timing

Description

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

Required Values

On or **Off**. **On** indicates that the time-stamp is displayed in the trace. The default is **On**. This can also be 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 Values

A number ranging from **0** to **864,000** 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.

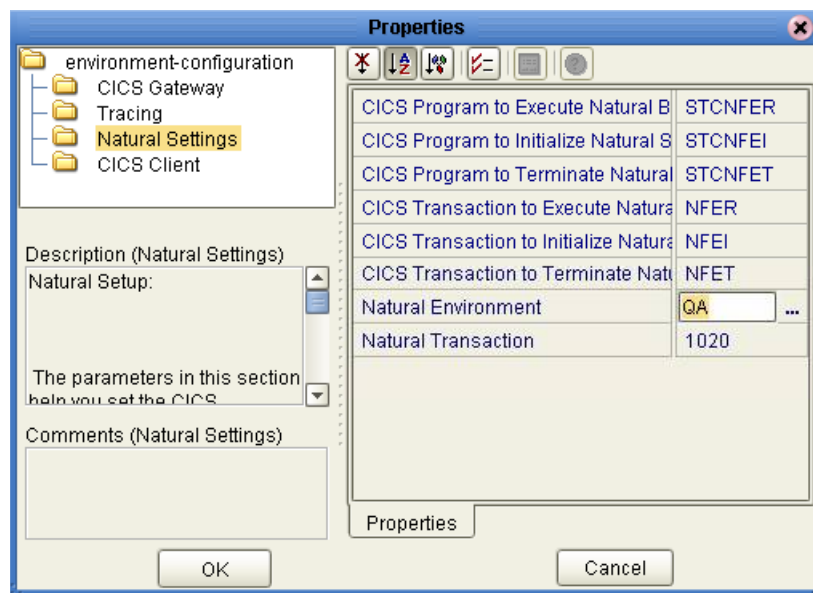
3.4.3. Natural Settings

This section contains the following top-level parameter used for setting the Tracing.

Note: *If invalid values are entered in the Natural Settings (e.g. the Program to Terminate Natural session) valuable resources are used up in the mainframe.*

- CICS Program to Execute Natural Business Rules
- CICS Program to Initialize Natural Session
- CICS Program to Terminate Natural Session
- CICS Transaction to Execute Natural Business Rules
- CICS Transaction to Initialize Natural Session
- CICS Transaction to Terminate Natural Session
- Natural Environment
- Natural Transaction

Figure 36 ADABAS Natural eWay Environment Configuration - Natural Settings



CICS Program to Execute Natural Business Rules

Description

Specifies the name of the CICS program that executes Natural Business Rules.

Required Values

STCNFER

CICS Program to Initialize Natural Session

Description

Specifies the name of the CICS program that initializes the Natural session.

Required Value

STCNFEI

CICS Program to Terminate Natural Session

Description

Specifies the name of the CICS program that terminates the Natural session.

Required Value

STCNFET

CICS Transaction to Execute Natural Business Rules

Description

Specifies the name of the CICS transaction that executes Natural Business Rules.

Required Value

NFER

CICS Transaction to Initialize Natural Session

Description

Specifies the name of the CICS transaction that initializes the Natural session.

Required Value

NFEI

CICS Transaction to Terminate Natural Session

Description

Specifies the name of the CICS transaction that terminates the Natural session.

Required Value

NFET

Natural Environment

Description

Specifies the Natural environment to be run. Multiple environments can be run concurrently in a single CICS region.

Required Value

A two byte value representing the valid Natural environment to be run.

Natural Transaction

Description

Specifies the Natural transaction type, which controls the processing of Business Rules transaction messages coming in from external ADABAS Natural eWays.

Required Value

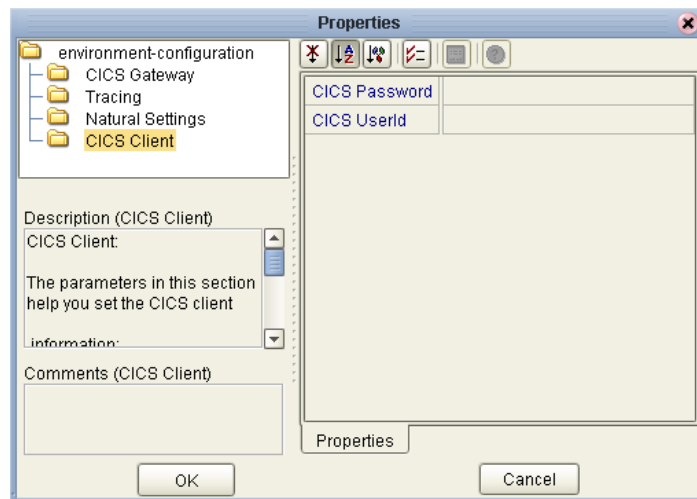
A four-byte value, in string format, representing a valid transaction type.

3.4.4. CICS Client

This section contains the following top-level parameter used for setting the CICS Client.

- CICS Password
- CICS UserID

Figure 37 ADABAS Natural eWay Environment Configuration - CICS Client



CICS Password

Description

Specifies the password associated with the specific CICS user.

Required Values

A valid CICS user password. Maximum length is eight characters.

CICS UserID

Description

Specifies the ID for the CICS user.

Required Value

A valid CICS user ID. Maximum length is eight characters.

Note: *It is recommended that you include the CICS client user ID and password in the EWC configuration. Otherwise you will be prompted for this information, which suspends all processing until valid values are entered. This recommendation is only for Windows versions of the client, because the UNIX versions generate an exception under the same conditions.*

3.5 Adding Jar Files to the Logical Host

When using **local:** as a URL value, you must install both the **ctgclient.jar** and the **ctgserver.jar** files, in the environment configuration, before building an ADABAS Natural eWay project. You can add both Jar files from the Enterprise Designer using the **Upload Files** option. This ensures that the Logical Host will reload it if rebuilt. Do not remove the location of the Jar file (since the repository remembers its location).

To add the required JAR files from the Enterprise Designer using the Upload Files option, complete the following steps:

- 1 From the Enterprise Explorer tree, right-click the Logical Host and select **Upload File** from the shortcut menu. The **Upload Third Party Files** dialog box appears.
- 2 From the **Upload Third Party Files** dialog box, click **Add**.
- 3 Browse for the appropriate Jar files and select them to be uploaded by clicking **Open**. The **Upload Third Party Files** dialog box appears as shown in Figure 38.

Figure 38 Upload Third Party Files dialog box



- 4 Click **OK**.
- 5 Right-click the Logical Host and select **Apply** from the menu.

Adding Jar Files Manually

You can add both Jar files manually by copying the Jar file into the following location:

(<ICAN Home>\logicalhost\stcis\lib)

where <ICAN Home> is the directory in which ICAN is installed.

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

ADABAS Natural Wizard Operation

This section explains how to convert **.sag** files into OTDs usable by the ADABAS Natural eWay. You can use the ADABAS Natural wizard to do this operation.

What's in This Chapter

- [“Getting Started” on page 62](#)
- [“Using the ADABAS Natural Wizard” on page 64](#)

4.1 Getting Started

Before you can use the SAG wizard to convert **.sag** files into OTDs, you must first unload the source code for the Natural Local Data Areas (LDAs) and/or Parameter Data Areas (PDAs) that their respective Natural applications are using.

4.1.1 Creation of **.sag** Files

Natural source-code objects of the type LDA and PDA are the input used to create **.sag** files. Create the **.sag** files by using the Natural Unload utility on an z/OS mainframe. Using this utility, unload each LDA or PDA Natural source-code object to an MVS Physical Sequential (PS) file with the extension **.sag**.

Create the PS files and give them the appropriate extension by using the Natural Unload utility. Make sure there is a one-to-one correspondence between each LDA or PDA object and its corresponding **.sag** file.

Note: *The Natural Unload utility is a Software AG Natural product. This utility is also referred to as the SYSTRANS utility.*

4.1.2 Converting .sag Files: The ADABAS Natural Wizard

You can use the ADABAS Natural wizard to convert .sag files to eGate OTDs. To accomplish the conversion, first use the Natural Unload utility to unload a single source-code object for each PS file (as described earlier). When this operation is done, use FTP to transfer each of the PS files (in binary mode) to a Windows system workstation running the eGate Enterprise Manager.

Note: *Unload the Natural LDAs and PDAs via the Natural Unload utility with the EBCDIC-to-ASCII option set Y (Yes).*

Once these PS files reside on a machine that is available to the Enterprise Designer, you are ready to use the ADABAS Natural wizard to convert them to OTDs.

Generate an eDesigner-compatible OTD from each PS file containing an LDA or PDA object by using the ADABAS Natural wizard. The wizard does not add or delete anything from the source objects. Instead, it simply converts them to a file format that eGate and the ADABAS Natural eWay can use.

You must use the ADABAS Natural wizard to convert .sag files in this way whenever you need to exchange data between a Natural sub-program and the ADABAS Natural eWay. The OTDs generated by the SAG wizard are used, by the eWay only, to exchange data between eGate and the Natural sub-program that is being called.

4.2 Using the ADABAS Natural Wizard

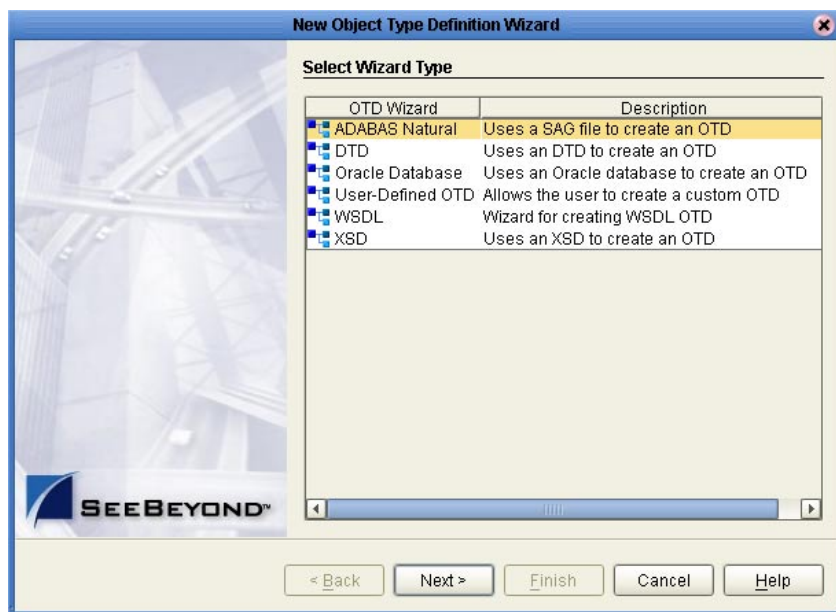
Once you have unloaded the source objects and converted them to **.sag** files (as explained earlier) you are ready to use the ADABAS Natural wizard.

To convert a **.sag** file to an OTD using the ADABAS Natural wizard

- 1 From the SeeBeyond Designer, right-click on the desired project and select **Object Type Definition**.

The New Object Type Definition Wizard dialog box appears, displaying all installed OTD wizards (see the example in Figure 39).

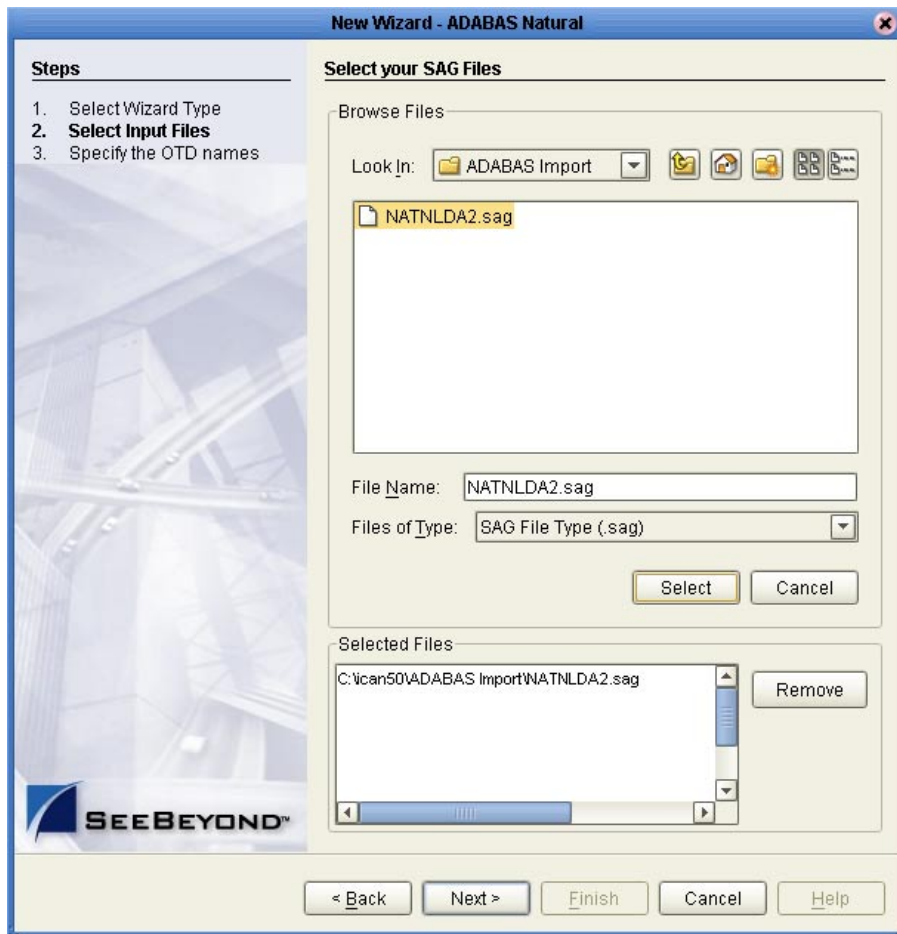
Figure 39 New Object Type Definition Wizard Dialog Box



- 2 Select ADABAS Natural from the list and click **Next**.

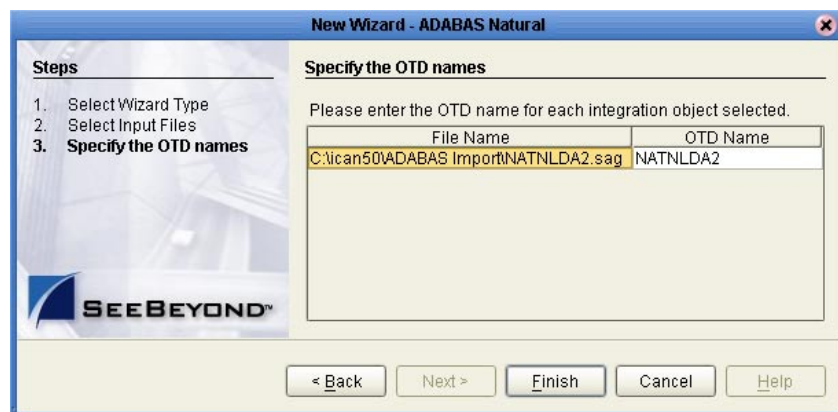
The Select Input Files dialog box appears, (see the example in Figure 40).

Figure 40 Select Input Files Dialog Box



- 3 Select the desired **.sag** file by either entering the file name or using the drop-down box and click the **Select** button.
- 4 When all applicable **.sag** files have been selected, click **Next** to continue. The Specify the OTD Names dialog box appears, (see the Figure 41).

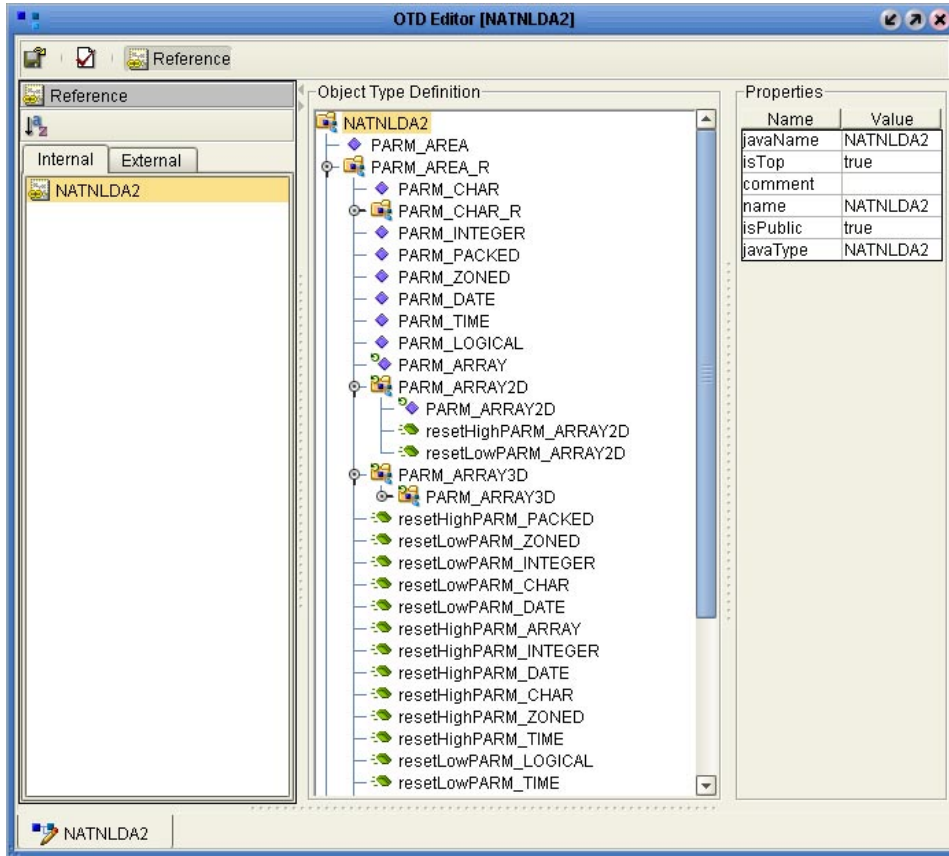
Figure 41 Specify the OTD Names Dialog Box



- 5 Enter the OTD name for each **.sag** file selected. Once an OTD name is entered, click on the File Name to enable it, then click **Finish**.

The OTD Editor dialog box appears, (see the Figure 42).

Figure 42 OTD Editor Main Window: Sample **.sag** File



- 6 Click **Compile and Save** on the **File** menu to compile and save the OTD. In the resulting **Save** dialog box, enter your desired name. This is your new OTD (**.xsc**) file based on the input **.sag** file.
- 7 Close the OTD Editor and exit back to the Enterprise Manager. Be sure you promote the new file to run time.

Example

The following text shows the source code for the Natural LDA NATNLDA2:

```

DEFINE DATA LOCAL
1 #PARM-AREA(A250)
* This is a comment line
1 REDEFINE #PARM-AREA
2 #PARM-CHAR(A36)
2 REDEFINE #PARM-CHAR
3 #PARM-CHAR2-ARRAY(A2/1:18)
2 #PARM-INTEGGER(I4)
2 #PARM-PACKED(P7)
2 #PARM-ZONED(N7)
2 #PARM-DATE(D)
2 #PARM-TIME(T)
2 #PARM-LOGICAL(L)
2 #PARM-ARRAY(A3/1:2)
2 #PARM-ARRAY2D(A5/1:2,1:2)
2 #PARM-ARRAY3D(A7/1:2,1:2,1:2)
END-DEFINE

```

The following text shows an example of a .sag file based on the Natural LDA NATNLDA2:

```

*H**ANAT310320011010852126 MVS/ESA          0AE B
*C**                                INTFDEV1NATNLDA2
L          *D01NAT3103L INTFDEV1NATNLDA2
MSCWT    MSCWT    TERM0235          *D02
2001092109313692001092109313690000000683
*D03MVS/ESA TSO    NATTSO
*D04
*S****DF      0000A 250 1#PARM-AREA
*S****C      0000* THIS IS A COMMENT LINE
*S****DRR     0000   R1#PARM-AREA
*S****DFR     0000A 36 2#PARM-CHAR
*S****DRR     0000   R2#PARM-CHAR
*S****DFRI1   0000A 2 3#PARM-CHAR2-ARRAY      (18)
*S****DFR     0000I 4 2#PARM-INTEGGER
*S****DFR     0000P 7 2#PARM-PACKED
*S****DFR     0000N 7 2#PARM-ZONED
*S****DFR     0000D 2#PARM-DATE
*S****DFR     0000T 2#PARM-TIME
*S****DFR     0000L 2#PARM-LOGICAL
*S****DFRI1   0000A 3 2#PARM-ARRAY            (2)
*S****DFRI2   0000A 5 2#PARM-ARRAY2D        (2,2)
*S****DFRI3   0000A 7 2#PARM-ARRAY3D        (2,2,2)
*E

```

Implementing the ADABAS Natural eWay

This chapter provides an introduction to the ADABAS Natural 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*.

What's in This Chapter

- “The ADABAS Natural eWay Sample Project” on page 68
- “CICS CTG Concerns” on page 75
- “Natural Sub-programs” on page 76

5.1 The ADABAS Natural eWay Sample Project

This section explains how to implement the ADABAS Natural eWay using the **ADABASNatural_JCE_Sample** project included on your installation CD-ROM. Find the sample on the CD-ROM, within **ADABASNaturaleWay.sar** file.

This sample allows you to observe an end-to-end data-exchange scenario involving eGate and the eWay. This chapter explains how to implement the sample project that uses the ADABAS Natural eWay.

Important: *This sample is provided for demonstration purposes. Because of the unique nature of your configuration, you will not be able to run this sample as shown.*

You can also use the procedures given in this chapter to create your own project based on the sample provided. You can use a combination of both methods, creating your own projects like the sample, then importing the sample into eGate to check your results.

Note: *For complete information on how to set up an eGate project, see the **Creating an End-to-end Scenario with eGate Integrator** document.*

Sample Project Overview

- 1 Import the sample Project as described in [“Importing a Sample Project” on page 69](#).
- 2 Configure each of the file eWays as described in [“Configure the File eWay” on page 70](#).
- 3 Create an Environment for the Project as described in [“Creating the Environment” on page 71](#).
- 4 Create and configure the Deployment Profile as described in [“Creating the Deployment Profile” on page 72](#).
- 5 Configure the **ADABASNatural** eWay—the eWay that exchanges the data with eGate—as described in [“Configure the ADABAS Natural eWay” on page 74](#).
- 6 Deploy the Project’s Deployment Profile as described in [“Activate the Project” on page 75](#).
- 7 Configure the Logical Host’s properties and start the bootstrap process as described in [“Run the Sample” on page 75](#).

To import and/or run the sample project, the ADABAS Natural eWay must be installed, and you must also have access to an ADABAS Natural system.

5.1.1 Importing a Sample Project

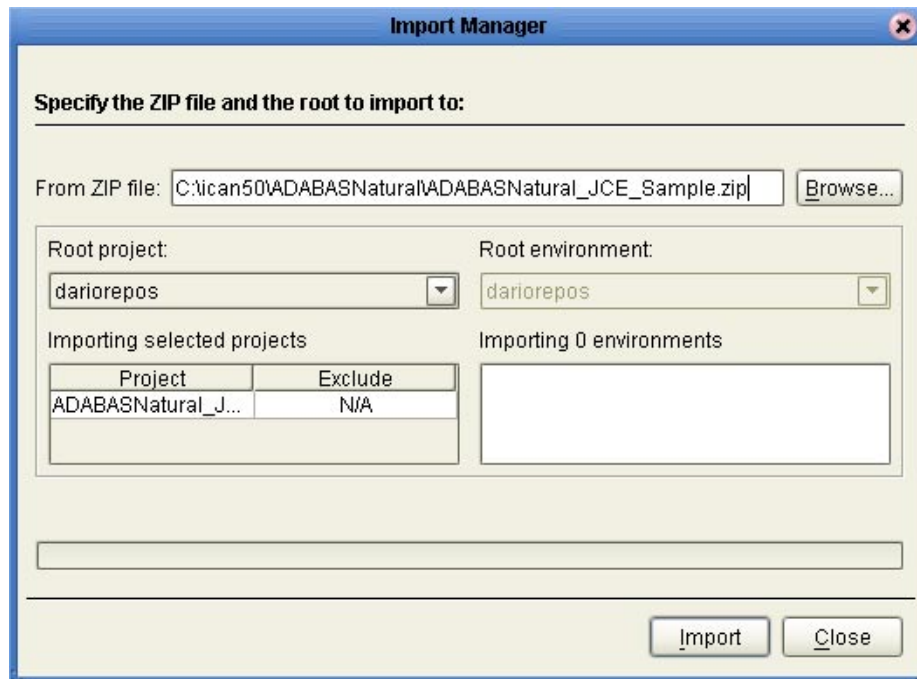
To download the sample from the Enterprise Manager

- 1 From the **Documentation** page of your Enterprise Manager, click the link for the **ADABAS Natural eWay Intelligent Adapter**.
The eGate documentation page appears.
- 2 Click **Download Sample**.
- 3 When you are prompted to open or save the file, click **Open**.
The sample .zip file opens in WinZip.
- 4 Use WinZip to extract the **ADABASNatural_JCE_Sample.zip** file to a location on your local file system, such as **C:\ICAN_Samples\ADABASNatural**. Make a note of where you extracted the file.

To import the sample in the Enterprise Designer

- 1 In the Enterprise Designer, right-click your Repository and click **Import**.
- 2 Click **Yes** to save any changes to your Repository.
The Import Manager appears.
- 3 Click **Browse** and navigate to the location where you downloaded the sample files in step 4 of the previous procedure. In the **JCE** folder, select the **ADABASNatural_JCE_Sample.zip** file and click **Open**.

Figure 43 Import Manager



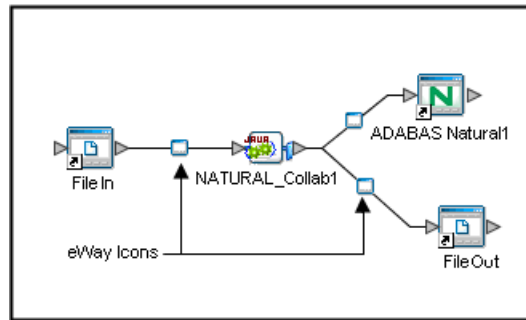
- 4 In the Import Manager, click **Import**.
- 5 An import status message appears when the import completes successfully. Click **OK** to continue.
- 6 Click **Close** to close the Import Manager and refresh the repository.
The **ADABASNatural_JCE** sample appears in the Enterprise Explorer pane of the Enterprise Designer.

5.1.2 Configure the File eWay

To configure the file eWays

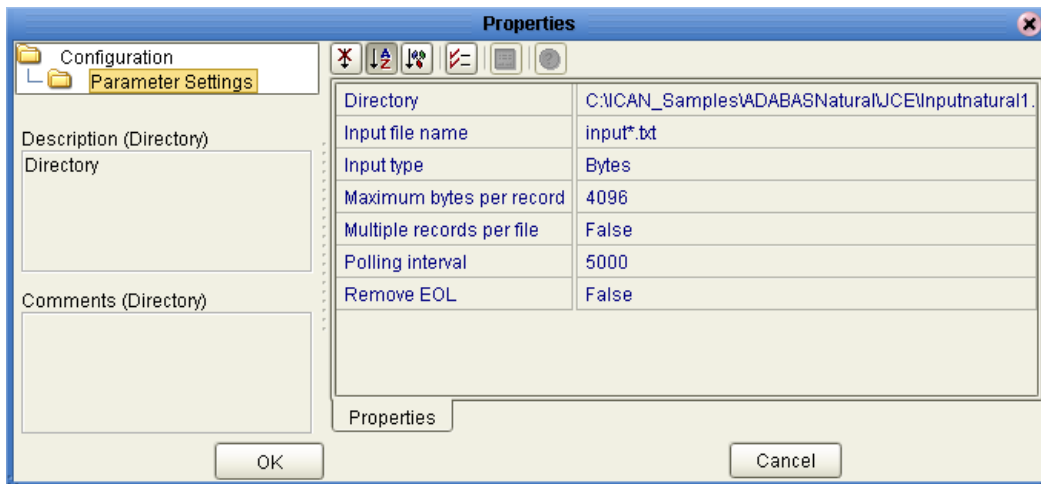
- 1 Before you can configure any of the components in the imported Project, you must first check them out of the Repository. To check out the Connectivity Map, right-click **CMap1**, click **Check Out**, and click **OK**.
- 2 Double-click **CMap1** to display the Connectivity Map.
- 3 Double-click the eWay icon between the **File 1** and the **Natural_Collab1** Service (the upper icon in the following figure) to edit the eWay's properties.

Figure 44 eWay Icons



- 4 In the **Directory** field, type the location of where you extracted the sample input files in step 2 of the previous procedure.

Figure 45 Input File Directory



- 5 Click **OK** to save the changes to the eWay's properties.
- 6 Similarly, configure the outbound file eWays to write the output files to an appropriate location, such as
C:\ICAN_Samples\ADABASNatural\JCE\output1.dat.

5.1.3 Creating the Environment

To create the Environment

- 1 In the Environment Explorer tab of the Enterprise Explorer, right-click the Repository and click **New Environment**.
- 2 Right-click the new Environment and click **Rename**. Name the Environment **ADABASNatural_JCE**.
- 3 Right-click the **ADABASNatural_JCE** Environment and click **New ADABAS Natural External System** with the following setting:
 - ♦ External System Name: **ADABAS Natural**

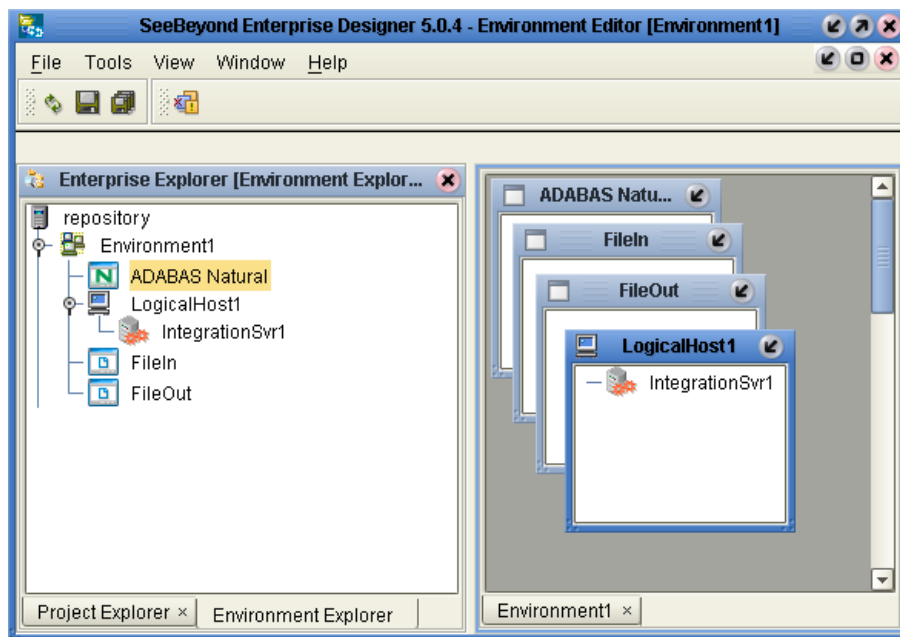
- Right-click the **ADABASNatural_JCE** Environment and click **New File External System**. Create two File External Systems with the following settings:

Table 6 File External Systems

Name	External System Type
File_In	Inbound File eWay
File_Out	Outbound File eWay

- Right-click the **ADABASNatural_JCE** Environment and click **New Logical Host**. A new Logical Host—**LogicalHost1**—appears.
- Right-click **LogicalHost1** and click **New SeeBeyond Integration Server**. A new Integration Server—**IntegrationSvr1**—appears.
- Right-click **LogicalHost1** and click **New SeeBeyond JMS IQ Manager**. The completed Environment appears as shown in [Figure 46 on page 72](#).

Figure 46 ADABAS Natural Environment



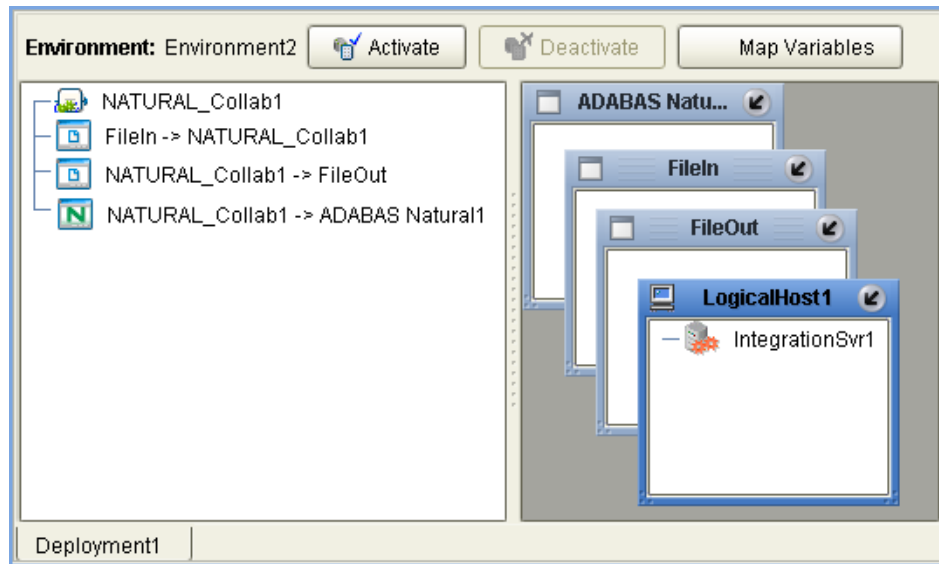
5.1.4 Creating the Deployment Profile

To create the Deployment Profile

- In the Project Explorer pane of the Enterprise Explorer, right-click the Project, click **New**, and click **Deployment Profile**.
- Name the Deployment Profile **Deployment1**, select the **ADABASNatural_JCE** Environment, and click **OK**.

The Deployment Profile displays all of the components to be deployed as well as the Logical Host and the external systems.

Figure 47 New Deployment Profile



- 3 Drag the components to their corresponding Logical Host or external systems as shown in the following figures (Figure 48 through Figure 51):

Figure 48 Logical Host

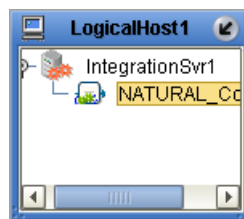


Figure 49 External File In



Figure 50 External File Out

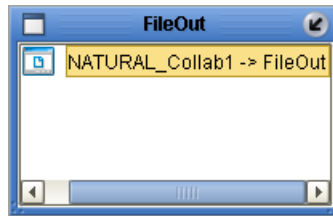
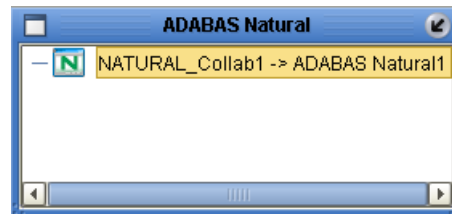


Figure 51 ADABAS Natural

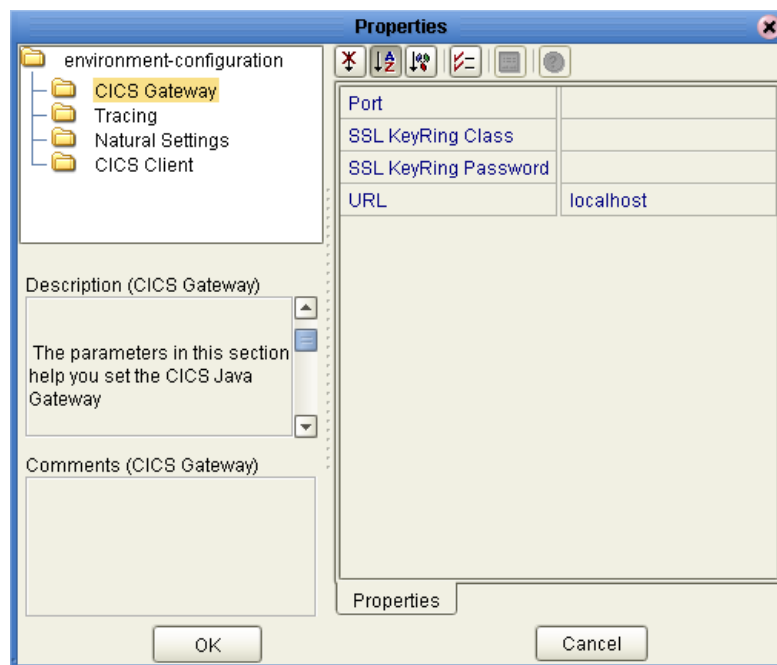


5.1.5 Configure the ADABAS Natural eWay

To configure the ADABAS Natural eWay

- 1 In the Environment Explorer tab of the Enterprise Explorer, right-click the **ADABAS Natural** external application and click **Properties**.

Figure 52 The ADABAS Natural eWay



- 2 Enter the the appropriate information for your e*Gate system.
Refer to [“Configuring the ADABAS Natural eWay Properties”](#) on page 50 for more information on configuring properties.

- 3 Click **OK** to save the properties for the **ADABAS NATural** external application.

5.1.6 Activate the Project

To deploy the Project

- 1 In the Project Explorer tab of the Enterprise Explorer pane, double-click the **Deployment1** Deployment Profile.
The Deployment Profile appears.
- 2 Click **Activate** to begin the activation process.
- 3 When the activation success message appears, click **No** to avoid updating the Logical Host immediately. Since you have not yet run the Logical Host service, there is no active Logical Host for the Deployment to update.

5.1.7 Run the Sample

To start the Bootstrap process

- 1 From a Command Prompt, navigate to the Bootstrap bin directory for your Logical Host. For example, **C:\ican50\logicalhost\Bootstrap\bin**.
- 2 To start the Bootstrap service, type the following command:

```
Bootstrap -e ADBASNatural_JCE -l LogicalHost1 -r repository_url  
-i username -p password
```

where

repository_url is the URL (including Repository name) for your Repository server. For example, **http://MyServer:15000/MyRepository**. Substitute the name and port for your Repository as well as the name of your Repository.

username/password are the user name and password for your eGate server.

5.2 CICS CTG Concerns

5.2.1 Defining more Than One Server When Using CTG

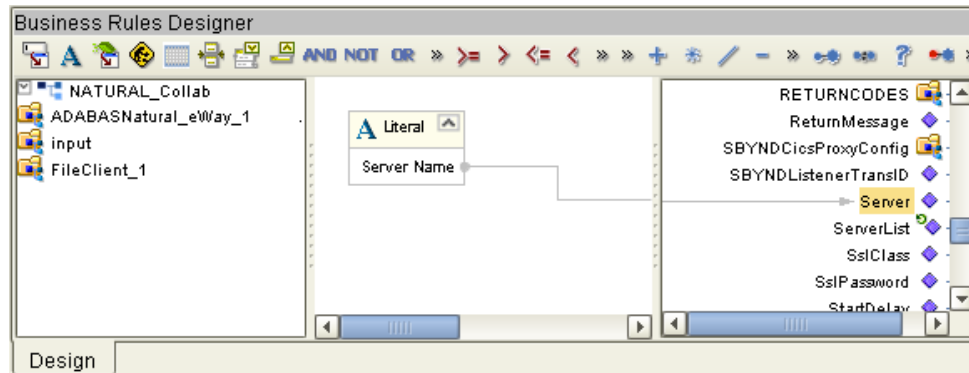
If the CICS Transaction Gateway (CTG) is used as the connection transport and more than one server is defined in a CTG installation, assign the target server name as a literal to the server node, before the ADABAS Natural_eWay.execute, in the Java Collaboration.

To Assign the Target Server

- 1 From the Java Collaboration Editor's Business Rules pane, click on the Business Rule above the ADABAS Natural_eWay.execute rule.
- 2 Click the rule icon on the Business Rules toolbar. An Empty Rule appears on the Business Rules tree.

- 3 Click the **Create Literal** icon on the Business Rules Designer toolbar. The Create Literal dialog box appears.
- 4 Select **String** as the type and enter the name of the target server as the value.
- 5 Click **OK**.
- 6 Map the output node of the Literal to the server, under **ADABASNatural_eWay_1**, in the right pane of the Business Rules Designer (see).

Figure 53 Business Rules Designer - Defining the Target Server



- 7 Continue creating your collaboration.

5.2.2 Matching the NaturalPayload Size

When using the CICS TransactionGateway transport, pad the data with spaces (if necessary) to match the full size of the NaturalPayload

5.3 Natural Sub-programs

This section explains how eGate communicates with Natural sub-programs.

Going from eGate to CICS, the ADABAS Natural eWay communicates with Natural sub-programs through the 3GL call interface using several components and programs. Upon completion of the Natural sub-programs, data is returned to eGate and the ADABAS Natural eWay through the same sequence of events, except in reverse.

5.3.1 Communication With eGate: Basic Steps

The steps in the communication from eGate to CICS-OS\390 happen as follows:

Within eGate

The ADABAS Natural eWay communicates with a SeeBeyond front-end component running inside CICS.

Within CICS on z/OS

- The SeeBeyond CICS front-end component runs a CICS Natural session.
- This session runs a back-end COBOL program within CICS.
- This program calls Natural sub-programs via the 3GL call interface.
- Upon the completion of the sub-programs, control is returned to the SeeBeyond back-end program (running under the Natural session).
- Data is also returned to the SeeBeyond program, along with control.

Note: The Natural sub-programs can optionally call a Natural main program, as long as control is returned back to the Natural sub-programs (that is, `FETCH RETURN`).

Return Data

The steps in the return of data from CICS-OS\390 back to eGate happen in the reverse order as shown in the previous lists.

Upon completion of the Natural sub-programs, the control, along with any return data, is returned to the SeeBeyond back-end program running under the Natural session. This session then sends the return data to the SeeBeyond front end-component, which then returns the data to eGate and the ADABAS Natural eWay.

Error Return Codes

This appendix lists and explains error return codes.

1.4 Error Return Codes

The following return codes can be found in the *IP CICS Sockets Guide Version 2 Release 8 and 9, OS/390 SecureWay Communications Server*. As of the date of the creation of this document, you can download it from the following URL:

<http://www-1.ibm.com/servers/s390/os390/bkserv/r10pdf/commserv.html>

The rest of this appendix explains these codes.

1.4.1 TCP/IP Error Return Codes

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
1	EPERM	All	Permission is denied. No owner exists.	Check that TCP/IP is still active; Check the protocol value of the socket call.
1	EDOM	All	Argument is too large.	Check parameter values of the function call.
2	ENOENT	All	The data set or directory was not found.	Check files used by the function call.
2	ERANGE	All	The result is too large.	Check parameter values of the function call.
3	ESRCH	All	The process was not found. A table entry was not located.	Check parameter values and structures pointed to by the function parameters.
4	EINTR	All	A system call was interrupted.	Check that the socket connection and TCP/IP are still active.

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
5	EIO	All	An I/O error occurred.	Check status and contents of source database if this occurred during a file access.
6	ENXIO	All	The device or driver was not found.	Check status of the device attempting to access.
7	E2BIG	All	The argument list is too long.	Check the number of function parameters.
8	ENOEXEC	All	An EXEC format error occurred.	Check that the target module on an exec call is a valid executable module.
9	EBADF	All	An incorrect socket descriptor was specified.	Check socket descriptor value. It may be currently not in use or incorrect.
9	EBADF	Givesocket	The socket has already been given. The socket domain is not AF_INET.	Check the validity of function parameters.
9	EBADF	Select	One of the specified descriptor sets is an incorrect socket descriptor.	Check the validity of function parameters.
9	EBADF	Takesocket	The socket has already been taken.	Check the validity of function parameters.
10	ECHILD	All	There are no children.	Check if created subtasks still exist.
11	EAGAIN	All	There are no more processes.	Retry the operation. Data or condition may not be available at this time.
12	ENOMEM	All	There is not enough storage.	Check validity of function parameters.
13	EACCES	All	Permission denied, caller not authorized.	Check access authority of file.
13	EACCES	Takesocket	The other application did not give the socket to your application. Permission denied, caller not authorized.	Check access authority of file.
14	EFAULT	All	An incorrect storage address or length was specified.	Check validity of function parameters.
15	ENOTBLK	All	A block device is required.	Check device status and characteristics.
16	EBUSY	All	Listen has already been called for this socket. Device or file to be accessed is busy.	Check if the device or file is in use.
17	EEXIST	All	The data set exists.	Remove or rename existing file.

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
18	EXDEV	All	This is a cross-device link. A link to a file on another file system was attempted.	Check file permissions.
19	ENODEV	All	The specified device does not exist.	Check file name and if it exists.
20	ENOTDIR	All	The specified device does not exist.	Use a valid file that is a directory.
21	EISDIR	All	The specified directory is a directory.	Use a valid file that is not a directory.
22	EINVAL	All types	An incorrect argument was specified.	Check validity of function parameters.
23	ENFILE	All	Data set table overflow occurred.	Reduce the number of open files.
24	EMFILE	All	The socket descriptor table is full.	Check the maximum sockets specified in MAXDESC().
25	ENOTTY	All	An incorrect device call was specified.	Check specified IOCTL() values.
26	ETXTBSY	All	A text data set is busy.	Check the current use of the file.
27	EFBIG	All	The specified data set is too large.	Check size of accessed dataset.
28	ENOSPC	All	There is no space left on the device.	Increase the size of accessed file.
29	ESPIPE	All	An incorrect seek was attempted.	Check the offset parameter for seek operation.
30	EROFS	All	The data set system is Read only.	Access data set for read only operation.
31	EMLINK	All	There are too many links.	Reduce the number of links to the accessed file.
32	EPIPE	All	The connection is broken. For socket write/send, peer has shut down one or both directions.	Reconnect with the peer.
33	EDOM	All	The specified argument is too large.	Check and correct function parameters.
34	ERANGE	All	The result is too large.	Check parameter values.
35	EWouldBLOCK	Accept	The socket is in nonblocking mode and connections are not queued. This is not an error condition.	Reissue Accept().
35	EWouldBLOCK	Read Recvfrom	The socket is in nonblocking mode and read data is not available. This is not an error condition.	Issue a select on the socket to determine when data is available to be read or reissue the Read()/Recvfrom().

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
35	EWOULDBLOCK	Send Sendto Write	The socket is in nonblocking mode and buffers are not available.	Issue a select on the socket to determine when data is available to be written or reissue the Send(), Sendto(), or Write().
36	EINPROGRESS	Connect	The socket is marked nonblocking and the connection cannot be completed immediately. This is not an error condition.	See the Connect() description for possible responses.
37	EALREADY	Connect	The socket is marked nonblocking and the previous connection has not been completed.	Reissue Connect().
37	EALREADY	Maxdesc	A socket has already been created calling Maxdesc() or multiple calls to Maxdesc().	Issue Getablesize() to query it.
37	EALREADY	Setibmopt	A connection already exists to a TCP/IP image. A call to SETIBMOPT (IBMTCP_IMAGE), has already been made.	Only call Setibmopt() once.
38	ENOTSOCK	All	A socket operation was requested on a nonsocket connection. The value for socket descriptor was not valid.	Correct the socket descriptor value and reissue the function call.
39	EDESTADDRREQ	All	A destination address is required.	Fill in the destination field in the correct parameter and reissue the function call.
40	EMSGSIZE	Sendto Sendmsg Send Write	The message is too long. It exceeds the IP limit of 64K or the limit set by the setsockopt() call.	Either correct the length parameter, or send the message in smaller pieces.
41	EPROTOTYPE	All	The specified protocol type is incorrect for this socket.	Correct the protocol type parameter.
42	ENOPROTOOPT	Getsockopt Setsockopt	The socket option specified is incorrect or the level is not SOL_SOCKET. Either the level or the specified optname is not supported.	Correct the level or optname.
42	ENOPROTOOPT	Getibmsocket opt Setibmsocket opt	Either the level or the specified optname is not supported.	Correct the level or optname.
43	EPROTONOSUPPORT	Socket	The specified protocol is not supported.	Correct the protocol parameter.
44	ESOCKTNOSUPPORT	All	The specified socket type is not supported.	Correct the socket type parameter.
45	EOPNOTSUPP	Accept Givesocket	The selected socket is not a stream socket.	Use a valid socket.

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
45	EOPNOTSUPP	Listen	The socket does not support the Listen call.	Change the type on the Socket() call when the socket was created. Listen() only supports a socket type of SOCK_STREAM.
45	EOPNOTSUPP	Getibmopt Setibmopt	The socket does not support this function call. This command is not supported for this function.	Correct the command parameter. See Getibmopt() for valid commands. Correct by ensuring a Listen() was not issued before the Connect().
46	EPFNOSUPPORT	All	The specified protocol family is not supported or the specified domain for the client identifier is not AF_INET=2.	Correct the protocol family.
47	EAFNOSUPPORT	Bind Connect Socket	The specified address family is not supported by this protocol family.	For Socket(), set the domain parameter to AF_INET. For Bind(), and Connect(), set Sin_Family in the socket address structure to AF_INET.
47	EAFNOSUPPORT	Getclient Givesocket	The socket specified by the socket descriptor parameter was not created in the AF_INET domain.	The Socket() call used to create the socket must be changed to use AF_INET for the domain parameter.
48	EADDRINUSE	Bind	The address is in a timed wait because a LINGER delay from a previous close or another process is using the address.	If you want to reuse the same address, use Setsockopt() with SO_REUSEADDR. See Setsockopt(). Otherwise, use a different address or port in the socket address structure.
49	EADDRNOTAVAIL	Bind	The specified address is incorrect for this host.	Correct the function address parameter.
49	EADDRNOTAVAIL	Connect	The calling host cannot reach the specified destinations.	Correct the function address parameter.
50	ENETDOWN	All	The network is down.	Retry when the connection path is up.
51	ENETUNREACH	Connect	The network cannot be reached.	Ensure that the target application is active.
52	ENETRESET	All	The network dropped a connection on a reset.	Reestablish the connection between the applications.
53	ECONNABORTED	All	The software caused a connection abend.	Reestablish the connection between the applications.

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
54	ECONNRESET	All	The connection to the destination host is not available.	
54	ECONNRESET	Send Write	The connection to the destination host is not available.	The socket is closing. Issue Send() or Write() before closing the socket.
55	ENOBUFS	All	No buffer space is available.	Check the application for massive storage allocation call.
55	ENOBUFS	Accept	Not enough buffer space is available to create the new socket.	Call your system administrator.
55	ENOBUFS	Send Sendto Write	Not enough buffer space is available to send the new message.	Call your system administrator.
56	EISCONN	Connect	The socket is already connected.	Correct the socket descriptor on Connect() or do not issue a Connect() twice for the socket.
57	ENOTCONN	All	The socket is not connected.	Connect the socket before communicating.
58	ESHUTDOWN	All	A Send cannot be processed after socket shutdown.	Issue read/receive before shutting down the read side of the socket.
59	ETOOMANYREFS	All	There are too many references. A splice cannot be completed.	Call your system administrator.
60	ETIMEDOUT	Connect	The connection timed out before it was completed.	Ensure the server application is available.
61	ECONNREFUSED	Connect	The requested connection was refused.	Ensure server application is available and at specified port.
62	ELOOP	All	There are too many symbolic loop levels.	Reduce symbolic links to specified file.
63	ENAMETOOLONG	All	The file name is too long.	Reduce size of specified file name.
64	EHOSTDOWN	All	The host is down.	Restart specified host.
65	EHOSTUNREACH	All	There is no route to the host.	Set up network path to specified host and verify that host name is valid.
66	ENOTEMPTY	All	The directory is not empty.	Clear out specified directory and reissue call.

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
67	EPROCLIM	All	There are too many processes in the system.	Decrease the number of processes or increase the process limit.
68	EUSERS	All	There are too many users on the system.	Decrease the number of users or increase the user limit.
69	EDQUOT	All	The disk quota has been exceeded.	Call your system administrator.
70	ESTALE	All	An old NFS** data set handle was found.	Call your system administrator.
71	EREMOTE	All	There are too many levels of remote in the path.	Call your system administrator.
72	ENOSTR	All	The device is not a stream device.	Call your system administrator.
73	ETIME	All	The timer has expired.	Increase timer values or reissue function.
74	ENOSR	All	There are no more stream resources.	Call your system administrator.
75	ENOMSG	All	There is no message of the desired type.	Call your system administrator.
76	EBADMSG	All	The system cannot read the file message.	Verify that CS for z/OS installation was successful and that message files were properly loaded.
77	EIDRM	All	The identifier has been removed.	Call your system administrator.
78	EDEADLK	All	A deadlock condition has occurred.	Call your system administrator.
78	EDEADLK	Select Selectex	None of the sockets in the socket descriptor sets is either AF_NET or AF_IUCV sockets, and there is no time-out or no ECB specified. The select/selectex would never complete.	Correct the socket descriptor sets so that an AF_NET or AF_IUCV socket is specified. A time-out of ECB value can also be added to avoid the select/selectex from waiting indefinitely.
79	ENOLCK	All	No record locks are available.	Call your system administrator.
80	ENONET	All	The requested machine is not on the network.	Call your system administrator.
81	ERREMOTE	All	The object is remote.	Call your system administrator.
82	ENOLINK	all	The link has been severed.	Release the sockets and reinitialize the client-server connection.
83	EADV	All	An ADVERTISE error has occurred.	Call your system administrator.

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
84	ESRMNT	All	AnSRMOUNT error has occurred.	Call your system administrator.
85	ECOMM	All	A communication error has occurred on a Send call.	Call your system administrator.
86	EPROTO	All	A protocol error has occurred.	Call your system administrator.
87	EMULTIHOP	All	A multihop address link was attempted.	Call your system administrator.
88	EDOTDOT	All	A cross-mount point was detected. This is not an error.	Call your system administrator.
89	EREMCHG	all	The remote address has changed.	Call your system administrator.
90	ECONNCLOSED	All	The connection was closed by a peer.	Check that the peer is running.
113	EBADF	All	Socket descriptor is not in correct range. The maximum number of socket descriptors is set by MAXDESC(). The default range is 0 to 49.	Reissue function with corrected socket descriptor.
113	EBADF	Bind socket	The socket descriptor is already being used.	Correct the socket descriptor.
113	EBADF	Givesocket	The socket has already been given. The socket domain is not AF_INET.	Correct the socket descriptor.
113	EBADF	Select	One of the specified descriptor sets is an incorrect socket descriptor.	Correct the socket descriptor. Set on Select() or Selectex().
113	EBADF	Takesocket	The socket has already been taken.	Correct the socket descriptor.
113	EBADF	Accept	A Listen() has not been issued before the Accept()	Issue Listen() before Accept().
121	EINVAL	All	An incorrect argument was specified.	Check and correct all function parameters.
145	E2BIG	All	The argument list is too long.	Eliminate excessive number of arguments.
156	EMVSiNITIAL	All	Process initialization error. This indicates a z/OS UNIX process initialization failure. This is usually an indication that a proper OMVS RACF/E segment is not defined for the user ID associated with application. The RACF OMVS segment may not be defined or may contain errors such as an improper HOME() directory specification	Attempt to initialize again.
1002	EIBMSOCKOUTOFRANGE	Socket	A socket number assigned by the client interface code is out of range.	check the socket descriptor parameter.
1003	EIBMSOCKINUSE	Socket	A socket number assigned by the client interface code is already in use.	Use a different socket descriptor.

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
1004	EIBMIUCVERR	All	The request failed because of an IUCV error. This error is generated by the client stub code.	Ensure IUCV/VMCF is functional.
1008	EIBMCONFLICT	All	This request conflicts with a request already queued on the same socket.	Cancel the existing call or wait for its completion before reissuing this call.
1009	EIMBCANCELLED	All	The request was cancelled by the CANCEL call.	Informational, no action needed.
1011	EIBMADTCPNAME	All	A TCP/IP name that is not valid was detected.	Correct the name specified in the IBM_TCPIIMAGE structure.
1011	EIBMADTCPNAME	Setibmopt	A TCP/IP name that is not valid was detected.	Correct the name specified in the IBM_TCPIIMAGE.
1011	EIBMADTCPNAME	INITAPI	A TCP/IP name that is not valid was detected.	Correct the name specified on the IDENT option TCPNAME field.
1012	EIBMADREQUESTCODE	All	A request code that is not valid was detected.	Contact your system administrator.
1013	EIBMADCONNECTIONSTATE	All	A connection token that is not valid was detected; bad state.	Verify TCP/IP is active.
1014	EIBMUNAUTHORIZED CALLER	All	An unauthorized caller specified an authorized keyword.	Ensure user ID has authority for the specified operation.
1015	EIBMADCONNECTIONMATCH	All	A connection token that is not valid was detected. There is no such connection.	Verify TCP/IP is active.
1016	EIBMTCPABEND	All	An abend occurred when TCP/IP was processing this request.	Verify that TCP/IP has restarted.
1023	EIBMTERMERROR	All	Encountered a terminating error while processing.	Call your system administrator
1026	EIBMINVDELETE	All	Delete requestor did not create the connection.	Delete the request from the process that created it.
1027	EIBMINVSOCKET	All	A connection token that is not valid was detected. No such socket exists.	Call your system programmer.
1028	EIBMINTCPCONNECTION	All	Connection terminated by TCP/IP. The token was invalidated by TCP/IP.	Reestablish the connection to TCP/IP.
1032	EIBMCALLINPROGRESS	All	Another call was already in progress.	Reissue after previous call has completed.
1036	EIBMNOACTIVETCP	Getibmopt	No TCP/IP image was found.	Ensure TCP/IP is active.

Table 7 TCP/IP for MVS Error Return Codes

Error Number	Message Name	Socket Type	Error Description	Programmer's Response
1037	EIBMINTSRBUSERDATA	All	The request control block contained data that is not valid.	check your function parameters and call your system programmer.
1038	EIBMINTUSERDATA	All	The request control block contained user data that is not valid.	Check your function parameters and call your system programmer.
1040	EIBMSELECTEXPOST	SELECTEX	SELECTEX passed an ECB that was already posted.	Check whether the user's ECB was already posted.
2001	EINVALIDRXSOCKETCALL	REXX	A syntax error occurred in the RXSOCKET parameter list.	Correct the parameter list passed to the REXX socket call.
2002	ECONSOLEINTERRUPT	REXX	A console interrupt occurred.	Retry the task.
2003	ESUBTASKINVALID	REXX	The subtask ID is incorrect.	Correct the subtask ID on the INITIALIZE call.
2004	ESUBTASKALREADYACTIVE	REXX	The subtask is already active.	Only issue the INITIALIZE call once in your program.
2005	ESUBTASKALNOTACTIVE	REXX	The subtask is not active.	Issue the INITIALIZE call before any other socket call.
2006	ESOCKETNOTALLOCATED	REXX	The specified socket could not be allocated.	Increase the user storage allocation for this job.
2007	EMAXSOCKETSREACHED	REXX	The maximum number of sockets has been reached.	Increase the number of allocate sockets, or decrease the number of sockets used by your program.
2009	ESOCKETNOTDEFINED	REXX	The socket is not defined.	Issue the SOCKET call before the call that fails.
2011	EDOMAINSERVERFAILURE	REXX	A Domain Name Server failure occurred.	Call your MVS system programmer.
2012	EINVALIDNAME	REXX	An incorrect name was received from the TCP/IP server.	Call your MVS system programmer.
2013	EINVALIDCLIENTID	REXX	An incorrect client ID was received from the TCP/IP server.	Call your MVS server.
2014	EINVALIDFILENAME	REXX	An error occurred during NUCEXT processing.	Specify the correct translation table file name, or verify that the translation table is valid.
2016	EHOSTNOTFOUND	REXX	The host is not found.	Call your MVS system programmer.
2017	EIPADDRNOTFOUND	REXX	Address not found.	Call your MVS system programmer.

1.4.2 Sockets Extended Error Return Codes

Table 8 explains the Sockets Extended error return codes.

Table 8 Sockets Extended Error Return Codes

Error Code	Problem Description	System Action	Programmer's Response
10100	An ESTATE macro did not complete normally.	End the call.	Call your MVS system programmer.
10101	A STORAGE OBTAIN failed.	End the call.	Increase MVS storage in the application's address space.
10108	The first call from TCP/IP was not INITAPI or TAKESOCKET.	End the call.	Change the first TCP/IP call to INITAPI or TAKESOCKET.
10110	LOAD of EZBSOH03 (alias EZASOH03) failed.	End the call.	Call the IBM Software Support Center.
10154	Errors were found in the parameter list for an IOCTL call.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the IOCTL call. You may have incorrect sequencing of socket calls.
10155	The length parameter for an IOCTL call is 3200 (32 x 100).	Disable the subtask for interrupts. Return an error code to the caller.	Correct the IOCTL call. You may have incorrect sequencing of socket calls.
10156	The length parameter for an IOCTL call is 3200 (32 x 100).	Disable the subtask for interrupts. Return an error code to the caller.	Correct the IOCTL call. You may have incorrect sequencing of socket calls.
10159	A zero or negative data length was specified for a READ or READV call.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the length in the READ call.
10161	The REQARG parameter in the IOCTL parameter list is zero.	End the call.	Correct the program.
10163	A 0 or negative data length was found for a RECV, RECVFROM, or RECVMSG call.	Disable the subtask for interrupts. Sever the DLC path. Return an error code to the caller.	Correct the data length.
10167	The descriptor set size for SELECT or SELECTEX call is less than or equal to zero.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the SELECT or SELECTEX call. You may have incorrect sequencing of socket calls.
10168	The descriptor set size in bytes for a SELECT or SELECTEX call is greater than 252. A number greater than the maximum number of allowed sockets (2000 is maximum) has been specified.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the descriptor set size.
10170	A zero or negative data length was found for a SEND or SENDMSG call.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the data length in the SEND call.
10174	A zero or negative data length was found for a SENDTO call.	Disable the subtask for interrupts. Return an error code to the caller.	correct the data length in the SENDTO call.

Table 8 Sockets Extended Error Return Codes

Error Code	Problem Description	System Action	Programmer's Response
10178	The SETSOCKOPT option length is less than the minimum length.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the OPTLEN parameter.
10179	The SETSOCKOPT option length is greater than the maximum length.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the OPTLEN parameter.
10184	A data length of zero was specified for a WRITE call.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the data length in the WRITE call.
10186	A negative data length was specified for a WRITE or WRITEV call.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the data length in the WRITE call.
10190	The GETHOSTNAME option length is less than 24 or greater than the maximum length.	Disable the subtask for interrupts. Return an error code to the caller.	Correct the length parameter.
10193	The GETSOCKOPT option length is less than the minimum or greater than the maximum length.	End the call.	Correct the length parameter.
10197	The application issued an INITAPI call after the connection was already established.	Bypass the call.	Correct the logic that produces the INITAPI call that is not valid.
10198	The maximum number of sockets specified for an INITAPI exceeds 2000.	Return to the user.	Correct the INITAPI call.
10200	The first call issued was not a valid first call.	End the call.	For a list of valid first calls, refer to the section on special considerations in the chapter on general programming.
10202	The RETARG parameter in the IOCTL call is zero.	End the call.	Correct the parameter list. You may have incorrect sequencing of socket calls.
10203	The requested socket number is a negative value.	End the call.	Correct the requested socket number.
10205	The requested socket number is a negative value.	End the call.	Correct the requested socket number.
10208	the NAMELEN parameter for a GETHOSTYNAME call was not specified.	End the call.	Correct the NAMELEN parameter. You may have incorrect sequencing of socket calls.
10209	The NAME parameter on a GETHOSTBYNAME call was not specified.	End the call.	Correct the NAME parameter. You may have incorrect sequencing of socket calls.
10210	The HOSTENT parameter on a GETHOSTBYNAME call was not specified.	End the call.	Correct the HOSTENT parameter. You may have incorrect sequencing of socket calls.

Table 8 Sockets Extended Error Return Codes

Error Code	Problem Description	System Action	Programmer's Response
10211	The HOSTADDR parameter on a GETHOSTBYNAME or GETHOSTBYADDR call is incorrect.	End the call.	Correct the HOSTENT parameter. You may have incorrect sequencing of socket calls.
10212	The resolver program failed to load correctly for GETHOSTBYNAME or GETHOSTBYADDR call.	End the call.	Check the JOBLIB, STEPLIB, and LINKLIB data sets and rerun the program.
10213	Not enough storage is available to allocate the HOSTENT structure.	End the call.	Increase the use storage allocation for this job.
10214	The HOSTENT structure was not returned by the resolver program.	End the call.	Ensure that the domain name server is available. This can be a nonerror condition indicating that the name or address specified in a GETHOSTBYADDR or GETHOSTBYNAME call could not be matched.
10215	The APITYPE parameter on an INITAPI call instruction was not 2 or 3.	End the call.	Correct the APITYPE parameter.
10218	The application programming interface (API) cannot locate the specified TCP/IP.	End the call.	Ensure that an API that supports the performance improvements related to CPU conservation is installed on the system and verify that a valid TCP/IP name was specified on the INITAPI call. This error call may also mean that EZASOKIN could not be loaded.
10219	The NS parameter is greater than the maximum socket for this connection.	End the call.	Correct the NS parameter on the ACCEPT, SOCKET or TAKESOCKET call.
10221	The AF parameter of a SOCKET call is not AF_INET.	End the call.	Set the AF parameter equal of AF_INET.
10222	the SOCTYPE parameter of a SOCKET call must be stream, datagram, or raw (1, 2, or 3).	End the call.	Correct the SOCTYPE parameter.
10223	No ASYNC parameter specified for INITAPI with APITYPE=3 call.	End the call.	Add the ASYNC parameter to the INITAPI call.
10224	The IOVCNT parameter is less than or equal to zero, for a READV, RECVMSG, SENDMSG, or WRITEV call.	End the call.	correct the IOVCNT parameter.
10225	The IOVCNT parameter is greater than 120, for a READV, RECVMSG, SENDMSG, or WRITEV call.	End the call.	Correct the IOVCNT parameter.

Table 8 Sockets Extended Error Return Codes

Error Code	Problem Description	System Action	Programmer's Response
10226	Invalid COMMAND parameter specified for a GETIBMOPT call.	End the call.	Correct the command parameter. of the GETIBMOPT call.
10229	A call was issued on an APITYPE=3 connection without an ECB or REQAREA parameter.	End the call.	Add an ECB or REQAREA parameter to the call.
10300	Termination is in progress for either the CICS transaction or the sockets interface.	End the call.	None.
10331	A call that is not valid was issued while in SRB mode.	End the call.	Get out of SRB mode and reissue the call.
10332	A SELECT call is invoked with a MAXSOC value greater than that which was returned in the INITAPI function (MAXSNO field).	End the call.	Correct the MAXSOC parameter and reissue the call.
10334	An error was detected in creating the data areas required to process the socket call.	End the call.	Call the IBM Software Support Center
10999	An abend has occurred in the subtask.	Write message EZY1282E to the system console. End the subtask and post the TRUE ECB.	If the call is correct, call your system programmer.
20000	An unknown function code was found in the call.	End the call.	Correct the SOC-FUNCTION parameter.
20001	The call passed an incorrect number of parameters.	End the call.	Correct the parameter list.
20002	The CICS Sockets Interface is not in operation.	End the call.	Start the CICS Sockets Interface before executing this call.

Index

A

- ADABAS Natural
 - applications 7
- ADABAS Natural eWay
 - architecture 9
 - basic operation 8
 - configuring properties 50
 - creating and configuring 49
 - creating the control VSAM file 23
 - errors 12
 - functional description 10
 - installation for z/OS 20
 - installing eWay Product Files 18
 - installing on an eGate Supported System 19
 - load modules 23
 - logging 11
 - properties 51
 - protocols 9
 - sample project 68
 - software requirements 16
 - supported external applications 16
 - system requirements 15
 - wizard 63
 - zip file 20
- ADABASNaturaleWay.sar 19, 20
- ADABASNaturaleWayDocs.sar 19
- applications
 - ADABAS Natural 7
- assign target server 75

B

- bootstrap process 75

C

- CEDA 20
- CEDA definitions
 - installation 22
- CICS
 - installation 33
- CICS requirements 17
- CICS server
 - requirements using CICS Transaction Gateway

- 16
- CICS Transaction Gateway
 - concerns 75
 - installation and configuration 22
 - requirements for the CICS server 16
- configuration
 - CICS Transaction Gateway 22
- Configuration Procedures 23
- configuring
 - ADABAS eWay properties 50
 - ADABAS Natural eWay 49
 - environment properties 53
 - eWay Connectivity Map properties 50
 - NTSYS setname 24
 - the File eWay 70
- Connectivity Map
 - configuration 50
- copying
 - tape contents to disk 22

D

- deploying project 75
- Deployment Profile 72
- downloading
 - ADABAS Natural eWay zip file 20

E

- environment properties
 - configuration 53
- errors 12
- External Applications
 - supported 16

F

- functional description 10

G

- global record 31

I

- installation
 - CICS
 - verification 33
- installing
 - ADABAS Natural eWay for z/OS 20
 - ADABAS Natural eWay from CD for z/OS 21
 - ADABAS Natural eWay on an eGate Supported System 19

- after installation 19
- before installing 18
- CICS CEDA definitions 22
- CICS Transaction Gateway 22
- eWay Product Files 18
- Natural programs 23
- procedures 22
- Interactive Simulator
 - Virtual JCL Submit eWay 39

L

- load modules
 - CICS DFHRPL concatenation 23
- linking 23

N

- Natural programs
 - installation 23
- Natural Sub-programs 76
- NaturalPayload 76
- NTSYS setname
 - configuration 24

O

- Operating Systems
 - supported 15
- OTD Wizard 64

P

- properties
 - CICS Client 52
 - CICS Password 60
 - CICS UserID 60
 - ECI Call Type 52
 - ECI Extend Mode 52
 - ECI LUW Token 52
 - Encoding 53
 - Message Qualifier 53
 - CICS Gateway
 - Port 54
 - SSL KeyRing Class 54
 - SSL KeyRing Password 55
 - URL 55
 - Connector 51
 - Class 51
 - Connection Transport 51
 - Property Tag 51
 - Type 51
 - Natural Settings

- CICS Program to Execute Natural Business Rules 58
- CICS Program to Initialize Natural Session 58
- CICS Program to Terminate Natural Session 59
- CICS Transaction to Execute Natural Business Rules 59
- CICS Transaction to Initialize Natural Session 59
- CICS Transaction to Terminate Natural Session 59
- Natural Environment 59
- Natural Transaction 59

- Tracing
 - Dump Offset 56
 - Filename 56
 - Level 56
 - Timing 57
 - Truncation Size 57

R

- Readme.txt file
 - specific version information 14
- Return Data 77

S

- sag files
 - converting 63
 - creating 62
- sample project 68
 - activate project 75
 - importing 69
 - overview 69
 - run sample 75
- Scope
 - ADABAS Natural eWay 12
- SeeBeyond Web Site 13
- software requirements 16
 - z/OS 16
- sub-programs 76
- system requirements 15

T

- target server 75
- transaction type record 31

V

- verification
 - CICS installation 33

Index

verifying
 eWay installation and operation 42
Virtual JCL Submit eWay Interactive Simulator 39
Virtual Natural eWay Interactive Simulator 34
Virtual Natural eWay monitoring screens z/OS
CICS 42
VSAM file 23

Z

z/OS system requirements 16
z/OS Systems 19
zip file
 downloading the ADABAS Natural eWay 20