SeeBeyond ICAN Suite

Standard e*Way Intelligent Adapter User's Guide

Release 5.0.5 for Schema Run-time Environment (SRE)

S E E B E Y O N D ®

Contents

6

45

Introduction

This document provides instructions for installing and configuring the SeeBeyond™ Technology Corporation's (SeeBeyond™) Standard e*Way™ Intelligent Adapter. This chapter provides an introduction to the e*Way and this guide.

1.1 Overview of e*Ways

e*Ways exchange data between external systems and the e*Gate Integrator system. These e*Ways perform the following main functions:

- Receive unprocessed data from external systems and package it as Events of known Event Types
- Forward Events to other components within the e*Gate Integrator system via Intelligent Queues (IQs)
- Send processed data to external systems

Standard e*Way

The Standard e*Way, provided as part of the e*Gate system, is actually a package of four individual e*Ways. They are:

- **Multi-Mode e*Way (stceway.exe):** A multi-threaded host for Java Collaborations.
- **File e*Way (stcewfile.exe)**: File-based e*Way.
- **Scheduler e*Way (stcewscheduler.exe):** Scheduler e*Way.
- **Loopback e*Way (stcewloopback.exe)**: Simulates an IP backend for the TCP/IP e*Way.

This guide explains how to configure and implement these e*Ways in the e*Gate environment.

1.2   Intended Reader

The reader of this document is presumed to be:

- A developer or system administrator with responsibility for maintaining the e*Gate system

- Someone with high-level knowledge of Windows and UNIX operations and administration

- Thoroughly familiar with Windows-style operations

1.3   Supported Operating Systems

The Standard e*Way package is available for the same operating systems as eGate Integrator. For information, refer to the *eGate Integrator Installation Guide.*

1.4   System Requirements

The Standard e*Way package is installed automatically when you install e*Gate Integrator. For complete information on system requirements and installing e*Gate, see the *e*Gate Integrator Installation Guide*.

To use a Standard e*Way, you need an e*Gate Participating Host.

JDK 1.3.1_02 is required to use the Multi-Mode e*Way. You can download JDK 1.3.1_02 (Java 2 SDK, Standard Edition, version 1.3.1_02) from the following source:

http://java.sun.com/j2se/

Be sure to select the type of Java 2 SDK that is appropriate to the desired operating system.

Using the e*Way Configuration Editor

This chapter explains the basics of how to use the e*Way Configuration Editor to configure an e*Way's parameter settings.

2.1 Overview

The e*Way Configuration Editor is a user interface that allows you to set and modify an e*Way's configuration parameters. You can access this interface through the e*Gate Schema Designer.

This chapter provides sufficient information about using this tool to configure e*Ways. For more detailed information about how to use this tool, see the online Help or the *e*Gate Integrator User's Guide*.

2.2 Configuring an e*Way

Many e*Ways function properly with the default configuration settings. However, when it is necessary, you can change the configuration parameters. To do so, complete the following steps:

1 In the e*Gate Schema Designer's Component editor, double-click the e*Way you want to configure and display its properties.

2 Under **Configuration File**, do one of the following:

   ◆ Click **New** to create a new configuration file.

   ◆ Click **Find** to select an existing configuration file.

   ◆ Click **Edit** to edit the file currently selected.

3 In the **Additional Command Line Arguments** box, type any additional command line arguments that the e*Way may require, taking care to insert them *at the end* of the existing command-line string.

It is important that you do not change any of the default arguments unless you have a specific need to do so.

Multi-Mode e*Way

This chapter describes the Multi-Mode e*Way and how to set its configuration parameters.

3.1 Functional Overview

The Multi-Mode e*Way uses **stceway.exe** as its executable file. It serves as a multi-threaded host for multiple Java Collaborations that use e*Way Connections to send and receive Events to and from multiple external systems or multiple JMS-managed queues.

Unlike other e*Ways, the Multi-Mode e*Way is not confined to a particular behavior. Instead, it derives its functionality from one or more e*Way Connections, some of which may be XA-aware.

Note: *The Multi-Mode e*Way can connect two or more external systems without an IQ. Such a connection would have to be synchronous and, if XA-noncompliant, would be subject to issues of nonpersistent data.*

3.2 Using the e*Gate Schema Designer to Create and Configure a New Multi-Mode e*Way

To create and configure a new Multi-Mode e*Way

1   Select the Navigator pane's Components tab in the Main window of the e*Gate Schema Designer.

2   Open the host and Control Broker where you want to create the e*Way.

3   On the Palette, click **new e*Way**.

4   Enter the name of the new e*Way, then click **OK**.

5   Double-click the name of the new e*Way on the Components tab.

The **e*Way Properties** dialog box opens

Note: *By default, the **Executable file** box should contain **bin\stceway.exe**. If it does not, click **Find** and navigate to that file.*

6 Under **Configuration File**, click **New**.

The e*Way Configuration Editor window opens.

7 Set the configuration parameters for this e*Way's configuration file (see **"Multi-Mode e*Way Configuration Parameters" on page 11** for details).

8 After selecting the desired parameters, click **Save** on the **File** menu to save the configuration (**.cfg**) file.

9 Close the **.cfg** file and e*Way Configuration Editor.

10 If desired, configure additional settings for the e*Way on the Startup, Advanced, and Security tabs of the e*Way Properties dialog box.

Note: *See the **e*Gate Integrator User's Guide** for more information about configuring e*Way properties on these tabs.*

11 When you are finished, click **OK** to save the e*Way configuration properties and close the dialog box.

3.3 Multi-Mode e*Way Configuration Parameters

This section describes the Multi-Mode e*Way configuration parameters. You will use the e*Way Configuration Editor to configure these settings.

This section is organized as follows:

- **"JVM Settings" on page 11**
- **"General Settings" on page 15**

3.3.1 JVM Settings

To correctly configure the Multi-Mode e*Way, you must configure the Java Virtual Machine (JVM) settings. This section explains the configuration parameters in the e*Way Configuration Editor window, which control the JVM settings.

JNI DLL Absolute Pathname

Description

Specifies the absolute path to the location on the Participating Host in which the JNI **.dll** (Windows) or shared library (UNIX) file is installed by the *Java 2 SDK*.

Required Values

A valid path.

Additional Information

This parameter must have a value. If you do not provide one, the default e*Gate installation path (**egate/client/jre**) will be used.

The JNI **.dll** or shared library file name varies, depending on the current operating system (OS). The following table lists the file names by OS:

Operating System	Java 2 JNI .dll or Shared Library Name
Windows	jvm.dll
Solaris Compaq Tru64 Red Hat Linux	libjvm.so
HP-UX	libjvm.sl
AIX	libjvm.a

The value assigned can contain a reference to an environment variable, by enclosing the variable name within a pair of "%" symbols, for example:

```
%MY_JNIDLL%
```

Such variables can be used when multiple Participating Hosts configured on different platforms or operating systems are used.

Note: *To ensure that the JNI **.dll** file loads successfully, the Dynamic Load Library search path environment variable must be set appropriately to include all the directories under the Java 2 SDK (or JDK) installation directory, which contain shared library or **.dll** files.*

CLASSPATH Prepend

Description

Specifies the paths to be prepended to the CLASSPATH environment variable for the JVM.

Required Values

An absolute path or an environmental variable. This parameter is optional.

Additional Information

If left unset, no paths are prepended to the CLASSPATH environment variable.

Existing environment variables may be referenced in this parameter by enclosing the variable name in a pair of "%" symbols, for example:

```
%MY_PRECLASSPATH%
```

CLASSPATH Override

Description

Specifies the complete CLASSPATH variable to be used by the JVM.

Note: *All necessary .**jar** and .**zip** files needed by both e*Gate and the JVM must be included. It is advised that the **CLASSPATH Prepend** parameter should be used.*

Required Values

An absolute path or an environment variable. This parameter is optional.

Additional Information

If left unset, an appropriate CLASSPATH environment variable (consisting of required e*Gate components concatenated with the system version of CLASSPATH) is set.

Existing environment variables may be referenced in this parameter by enclosing the variable name in a pair of "%" symbols, for example:

```
%MY_CLASSPATH%
```

CLASSPATH Append From Environment Variable

Description

Specifies whether user CLASSPATH environment variable is appended to the JVM settings.

Required Values

YES or **NO**. The configured default is **NO**.

Additional Information

In Windows environments, the length of the CLASSPATH variable is limited to a maximum of 2048 characters. If the CLASSPATH generated by a Collaboration exceeds that limit, then a system error condition can occur while instantiating the JVM.

Initial Heap Size

Description

Specifies the value for the initial heap size in bytes. If this value is set to 0 (zero), the preferred value for the initial heap size of the JVM is used.

Required Values

An integer from **0** to **2147483647**. This parameter is optional.

Maximum Heap Size

Description

Specifies the value of the maximum heap size in bytes. If this value is set to 0 (zero), the preferred value for the maximum heap size of the JVM is used.

Required Values

An integer from **0** to **2147483647**. This parameter is optional.

Maximum Stack Size for Native Threads

Description

Specifies the value of the maximum stack size in bytes for native threads. If this value is set to 0 (zero), the default value is used.

Required Values

An integer from **0** to **2147483647**. This parameter is optional.

Maximum Stack Size for JVM Threads

Description

Specifies the value of the maximum stack size in bytes for JVM threads. If this value is set to 0 (zero), the preferred value for the maximum heap size of the JVM is used.

Required Values

An integer between 0 and 2147483647. This parameter is optional.

Disable JIT

Description

Specifies whether the Just-In-Time (JIT) compiler is enabled or disabled.

Required Values

YES or **NO**. **NO** is the default.

*Note: The **Disable JIT** parameter is not supported for Java Release 1.*

Allow Remote Debugging of JVM

Description

Specifies the port number by which the e*Gate Java Debugger can connect with the JVM to allow remote debugging.

Required Values

An unused port number, ranging between 2000 and 65535. This parameter does not have a default value.

*Note: If you do not specify a value for this parameter, the e*Gate Java Debugger will not be able to connect to this e*Way.*

Suspend Option When Using Debugger

Description

Specifies that the e*Way should not do any processing until an e*Gate Java Debugger has successfully connected to it.

Required Values

YES or **NO**. The default value is **NO,** which allows e*Way processing to start immediately at startup.

Auxiliary JVM Configuration File

Description

Specifies the relative path to a JVM Properties file.

Required Values

A valid relative path name. This parameter is optional.

Additional Information

See the *e*Gate Integrator User's Guide* for more information about this parameter.

3.3.2 General Settings

This section describes parameter configuration for the Multi-Mode e*Way's general settings.

Rollback Wait Interval

Description

The value of the rollback wait interval in milliseconds.

Required Values

A value between 0 and 99999999. This is an optional parameter.

Standard IQ FIFO

Description

Specifies whether or not to receive messages from SeeBeyond Standard IQs in first-in-first-out (FIFO) order.

Required Values

YES or **NO**. **NO** is the default.

File e*Way

The File e*Way is a file-based e*Way that you can use to test your e*Gate configuration.

4.1 Functional Overview

The e*Way uses the **stcewfile.exe** executable file and operates as follows:

- As an inbound e*Way, the File e*Way polls an input directory for files of a specified extension (the default is **.fin**). When the e*Way detects a file with the extension, it opens the file and publishes the data to an IQ. The original data file is then renamed to the same filename with a **.~in** extension.

- As an outbound e*Way, the File e*Way appends processed data to a file in an output directory (the default filename is **output.dat**).

You can run the e*Way through the e*Gate Schema Designer or from a command line. See **"Running the File e*Way from a Command Line" on page 22** for command line instructions.

You must configure the e*Way's configuration parameters (including names and locations of source and destination files) in the e*Way Configuration Editor, which is part of the e*Gate Schema Designer. See **"Configuring an e*Way" on page 9** for specific information about using the e*Way Configuration Editor.

4.2 Configuration Parameters

The e*Way's configuration parameters are organized into the following sections:

- **"General Settings" on page 17**
- **"Outbound (send) Settings" on page 17**
- **"Poller (inbound) Settings" on page 19**
- **"Performance Testing" on page 21**

4.2.1 General Settings

The General Settings control whether the e*Way will process input or output Events.

AllowIncoming

Description

Determines whether the e*Way will process incoming (input) Events.

Required Values

YES or **NO**. Default is **YES**.

AllowOutgoing

Description

Determines whether the e*Way will process outgoing (output) Events.

Required Values

YES or **NO**. Default is **NO**.

PerformanceTesting

Description

Determines whether the e*Way will perform time tracking on processed Events based on parameters in the Performance Testing section.

Required Values

YES or **NO**. Default is **NO**.

Additional Information

See **"Performance Testing" on page 21** for additional parameters related to the PerformanceTesting parameter.

4.2.2 Outbound (send) Settings

The Outbound (send) Settings determine the e*Way's behavior for outgoing (output) operations.

OutputDirectory

Description

The directory to which output files will be written. If the **AllowOutgoing** parameter is set to **NO**, this parameter is ignored.

Required Values

A valid directory name with an absolute path (including the drive letter).

Additional Information

If you do not use an absolute path to the directory, the e*Gate system data path will be prepended. It will be relative to the e*Gate system data path specified in the **.egate.store** file. See the *e*Gate Integrator System Administration and Operations Guide* for information on the **.egate.store** file.

If you use an absolute path, the directory does not need to be within the e*Gate directory tree.

OutputFileName

Description

The name of the file to which the output will be written.

Required Values

A valid filename. The default is **output%d.dat**. The **%d** in the filename is a counter and increments for each new file.

Note: *Instead of "**%d**," you can use any other "printf" style that takes an integer or long value. For example, you can specify "**1%d**" or "**%012d**."*

Additional Information

If the output filename contains a counter variable, a new file is created based on the following parameters:

▪ If **MultipleRecordsPerFile** is set to **NO**, a new output file is created for each record.

▪ When the value of **MaxRecordsPerFile** is reached, a new output file is created.

MultipleRecordsPerFile

Description

Specifies whether multiple records can be written to the output file.

Required Values

YES or **NO**. The default is **YES**.

YES indicates that the input files contain multiple records. **NO** indicates that each input file contains a single record.

MaxRecordsPerFile

Description

Specifies the maximum number of records to be written to the output file. When the maximum is reached, a new file will be created.

Required Values

An integer between **1** and **4000000**. The default is **10000** (10,000 records).

AddEOL

Description

Specifies whether or not to add an end-of-line character to each record the e*Way sends to the output file.

Required Values

YES or **NO**. The default is **YES**.

4.2.3 Poller (inbound) Settings

The Poller (inbound) Settings determine the e*Way's behavior for inbound (input) operations.

PollDirectory

Description

The directory the e*Way will poll for input files. If the **AllowIncoming** parameter is set to **No**, this parameter is ignored.

Required Values

A valid directory name with an absolute path (including the drive letter). The default is **C:\INDATA**.

Additional Information

If you do not use an absolute path to the directory, the e*Gate system data path will be prepended. It will be relative to the e*Gate system data path specified in the **.egate.store** file. See the *e*Gate Integrator System Administration and Operations Guide* for information on the **.egate.store** file.

If you use an absolute path, the directory does not need to be within the e*Gate directory tree.

InputFileMask

Description

Specifies the file extension for input data files.

Required Values

A valid file extension. The default is ***.fin**.

The variable **%_LOGICALNAME%** is valid for this parameter. This variable causes the e*Way component's logical name to be substituted in the path, for example, **%_LOGICALNAME%.fin**.

PollMilliseconds

Description

The number of milliseconds the e*Way waits between poll attempts of the input directory.

Required Values

The acceptable range is an integer from **2** to **99999**, inclusive. The default is **1000** (1000 ms or 1 sec).

RemoveEOL

Description

Removes the end-of-line characters from each record that this e*Way processes as an input Event.

Required Values

YES or **NO**.

YES is the default. Select **NO** to disable this feature.

MultipleRecordsPerFile

Description

Indicates whether or not input files for this e*Way contain multiple records.

Required Values

YES or **NO**.

YES is the default, and indicates that the input files contain multiple records. **NO** indicates that each input file contains a single record.

Note: *If this parameter is set to **NO** and the file contains multiple records, only the first record will be read.*

MaxBytesPerLine

Description

Specifies the maximum number of bytes that each line within a multi-record data file can contain. If the input file uses a fixed record length, enter that length here.

Required Values

The acceptable range is an integer from **2** to **99999**, inclusive. The default is **4096** (4,096 bytes).

BytesPerLineIsFixed

Description

Indicates whether the value specified in the **MaxBytesPerLine** parameter represents the fixed length of each record.

Required Values

YES or **NO**.

NO is the default. Select **YES** to indicate that the value specified in the **MaxBytesPerLine** parameter represents the fixed length of each record. Select **NO** if records can have variable length.

File Records Per e*Gate Event

Description

Specifies the number of file records to combine to create a single e*Gate event.

Required Values

An integer between **1** and **9999999**.

The default is **1** (one file record to one e*Gate event).

4.2.4 Performance Testing

The Performance Testing settings determine the e*Way's behavior for testing operations.

LogEveryXEvents

Description

Specifies the number of Events for which to wait before logging the total Event count with a time stamp. This setting allows for a throughput analysis by reviewing the module's log file.

Required Values

The acceptable range is an integer from **1** to **99999**, inclusive. The default is **100** (100 Events).

InboundDuplicates

Description

Indicates the number of duplicate Events to send into e*Gate for every Event found in the input data file.

Required Values

The acceptable range is an integer from **1** to **99999999**, inclusive. The default is **1**.

4.3 Running the File e*Way from a Command Line

This section explains how to run and use the File e*Way using the command prompt.

Syntax

```
stcewfile command-flags @command-file
```

Where *command-flags* (separated by spaces) are listed in the table below, and *command-file* is an optional ASCII text file containing command flags and their arguments. Table 1 below shows the **stcewfile** command arguments.

Table 1 Command Arguments for stcewfile

Flag	Purpose	Required
-h	Displays a help message.	
-v	Verbose mode; shows additional information as commands are processed.	
--ver	Displays version information. Note that this flag requires two dashes.	
-d	Debug log on.	
-rh *host-name*	Registry host.	Yes
-rs *schema-name*	Schema name (If not specified, "default" schema will be used).	Yes
-ln *e*Way-name*	Name of the e*Way as defined within the specified schema.	Yes
-un *username*	User name as defined within the specified schema.	Yes
-up *password* **or** !*directoryname*	Password corresponding to the specified user name or name of the directory containing the *.egate.stcpass file (see **"Passwords"** below).	Yes
-ss	Run immediately, as service.*	
-sa	Install as service, start automatically on system startup.*	
-sm	Install as service, manual startup.*	
-sr	Remove service.*	

* See **"Services/Daemons Command Flag Differences for Windows and UNIX" on page 23** for differences between Windows and UNIX for these flags.

Passwords

The **-up** flag takes one of two arguments: the actual password corresponding to the user name, or an exclamation mark (!) followed by the directory containing the password file. Passwords are stored in an encrypted format within the password file; use this argument if you do not wish to have the password seen in clear text. See the *e*Gate Integrator System Administration and Operations Guide* for more information about password files.

Services/Daemons Command Flag Differences for Windows and UNIX

Table 2 lists the Windows and UNIX differences for the four services/daemons command flags.

Table 2 Service/daemon Flags

Flag	Purpose under Windows	Purpose under UNIX
-ss	Run as service. Reserved for use within Windows registry service definition; has no effect in a shell.	Run as a daemon (as a child process of init).
-sa	Install as service, start automatically on system startup.	No effect.
-sm	Install as service, manual startup.	No effect.
-sr	Remove service.	No effect.

Under Windows, you need sufficient privilege to add services to the Windows Registry (see the online Help system for the Windows utility **regedt32** for more information about registry security). Under UNIX, there are no privilege restrictions.

Scheduler e*Way

This chapter explains how to use the Scheduler e*Way.

5.1 Functional Overview

The Scheduler e*Way uses the **stcewscheduler.exe** executable file and is a self-contained entity that sends a specific Event according to a set schedule. The e*Way uses a single schedule file that contains both the schedule and the Event to be sent. You can use this e*Way to send a trigger to other modules.

You can run the e*Way through the e*Gate Schema Designer or from a command line. See **"Running the Scheduler e*Way from a Command Line" on page 26** for command line instructions.

You must configure the e*Way's configuration parameters in the e*Way Configuration Editor, which is part of the e*Gate Schema Designer. See **"Configuring an e*Way" on page 9** for specific information about using the e*Way Configuration Editor.

5.1.1 Schedule File Format

The schedule file is an ASCII text file that contains cron-style commands. A sample schedule file—**sample.sch**—is installed with the e*Way in the Registry under the **schedules** subdirectory of the default schema.

The commands in the schedule file can be in either of two formats, as described below.

Format 1

```
min hrs day_of_month month day_of_week event_name event_data
user_info
```

Where

min is 0-59 or *
hrs is 0-23 or *
day_of_month is 0-30 or *
month is 0-11, Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, or *
day_of_week is 0-6, Sun, Mon, Tues, Wed, Thu, Fri, Sat, or *
event_name is the name of the Event
event_data is the contents of the Event
user_info is any extra information

Note: *An asterisk (*) indicates that the value is inapplicable, or not used.*

You can enter frequency information in three ways:

- As a number (For example, 39)
- As a comma-delimited list (such as 1,5,10,15)
- As a range (such as 1-12 or 0-59/6)

Format 2

```
@command event_name event_data user_info
```

Where

command is annually, yearly, monthly, weekly, daily, midnight, or hourly
event_name is the name of the Event
event_data is the contents of the Event
user_info is any extra information

5.1.2 Output

The e*Way sends its output—the Event contained in the schedule file—to the queue as a set of three strings separated by pipes (|). For example:

event_name | event_data | user_info

5.2 Configuration Parameters

5.2.1 Schedule File

The Scheduler e*Way contains a single parameter, **Schedule File**.

Description

Specifies the name of the schedule file. For information about the schedule file, see **"Schedule File Format" on page 24**.

Required Values

A valid file name, including the absolute path location.

5.3 Running the Scheduler e*Way from a Command Line

This section explains how to run and use the Scheduler e*Way from the command prompt.

Syntax

```
stcewscheduler command-flags @command-file
```

Where *command-flags* (separated by spaces) are listed in the following table, and *command-file* is an optional ASCII text file containing command flags and their arguments. Table 3 shows the **stcewscheduler** command arguments.

Table 3 Command Arguments for stcewscheduler

Flag	Purpose	Required
-h	Displays a help message.	
-v	Verbose mode; shows additional information as commands are processed.	
--ver	Displays version information. Note that this flag requires two dashes.	
-d	Debug log on.	
-rh *host-name*	Registry host.	Yes
-rs *schema-name*	Schema name (If not specified, "default" schema will be used).	Yes
-ln *e*Way-name*	Name of the e*Way as defined within the specified schema.	Yes
-un *username*	User name as defined within the specified schema.	Yes
-up *password* **or** !*directoryname*	Password corresponding to the specified user name or name of directory containing the ***.egate.stcpass** file (see **"Passwords" on page 26**).	Yes
-ss	Run immediately, as service.*	
-sa	Install as service, start automatically on system startup.*	
-sm	Install as service, manual startup.*	
-sr	Remove service.*	

* See **"Services/Daemons Command Flag Differences for Windows and UNIX" on page 27** for differences between Windows and UNIX for these flags.

Passwords

The **-up** flag takes one of two arguments: the actual password corresponding to the user name, or an exclamation mark (!) followed by the directory containing the password file. Passwords are stored in an encrypted format within the password file; use this argument if you do not wish to have the password seen in clear text. See the *e*Gate Integrator System Administration and Operations Guide* for more information about password files.

Services/Daemons Command Flag Differences for Windows and UNIX

The following table lists the Windows and UNIX differences for the four services/daemons command flags.

Table 4 Service/daemon Flags

Flag	Purpose under Windows	Purpose under UNIX
-ss	Run as service. Reserved for use within Windows registry service definition; has no effect in a shell.	Run as a daemon (as a child process of init).
-sa	Install as service, start automatically on system startup.	No effect.
-sm	Install as service, manual startup.	No effect.
-sr	Remove service.	No effect.

Under Windows, you need sufficient privilege to add services to the Windows Registry (see the online Help system for the Windows utility **regedt32** for more information about registry security). Under UNIX, there are no privilege restrictions.

Loopback e*Way

This chapter explains how to use the Loopback e*Way.

6.1 Functional Overview

The Loopback e*Way uses the executable file **stcewloopback.exe** and is a self-contained module that simulates an IP backend for the TCP/IP e*Way. The e*Way sends an Event, as a request, to an external source, which returns a reply Event.

You can configure the e*Way's parameters (including names and locations of source and destination files) by editing the e*Way's configuration file in the e*Gate Schema Designer. See **"Configuring an e*Way" on page 9** for specific information about using the e*Way Configuration Editor.

6.2 Configuration Parameters

6.2.1 General Settings

The **Loopback e*Way** contains only one section, **General Settings**. The parameters in this section provide basic configuration for the e*Way, such as the host name, port numbers, and the number of ports to be used.

RRIPHostName

Description

This is the Request Reply TCP/IP host name.

Required Values

A string indicating the Request Reply TCP/IP host name. The default is **localhost**.

UseSinglePort

Description

This parameter determines whether to use a single IP port for both sending and receiving operations.

Required Values

YES or **NO**. The default is **NO** (do not use a single IP Port).

See Also:

INIPPort on page 29 and **SinglePortIsServer** on page 29.

SinglePortIsServer

Description

If the **UseSinglePort** is enabled, this parameter determines whether that port should be the server.

Required Values

YES or **NO**. The default is **NO** (The single IP Port is NOT the server).

See Also:

UseSinglePort on page 29.

INIPPort

Description

This is the IP port number on which this e*Way will listen (bind) for connections.

If UseSinglePort is selected, this is the port number this e*Way will use to send and receive Events.

Required Values

An integer between **1** and **65536**. The default is **26001**.

OUTIPPort

Description

This is the IP port number on which this e*Way will connect to the Request-Reply TCP/IP e*Way.

Required Values

An integer between **1** and **65536**. The default is **26002**.

LengthOfLength

Description

The length, in bytes, from offset zero that represents the length of the Event.

Required Values

A positive integer. The default is **10**.

UseAckNak

Description

This parameter determines whether the IP protocol uses Acknowledgment/Negative Acknowledgment.

Required Values

YES or **NO**. The default is **NO** (Do not use Acknowledgment/Negative Acknowledgment).

See Also:

AckString on page 30 and **NakString** on page 30.

AckString

Description

This parameter indicates the POSITIVE acknowledgment string.

Required Values

A string for the Positive acknowledgment message. The default is **PosAck**.

Note: This string MUST be the same length as NakString.

See Also:

UseAckNak on page 30 and **NakString** on page 30

NakString

Description

This parameter indicates the NEGATIVE acknowledgment string.

Required Values

A string for the Negative acknowledgment message. The default is **NegNak**.

Note: This string MUST be the same length as AckString.

See Also:

UseAckNak on page 30 and **AckString** on page 30.

Implementation

This chapter provides information about how to implement a Standard e*Way in a test environment, and presents a sample schema that includes the **stcewfile** and **stcewscheduler** e*Ways. By implementing the sample schema, you can ascertain that e*Gate has been successfully installed.

7.1 Overview

During installation of e*Gate, the *Host* and *Control Broker* have already been automatically created and configured. The default name of each is the name of the host on which you installed the e*Gate Schema Designer GUI.

To complete the sample implementation of an e*Way, do the following:

- Create a test schema.
- In the e*Gate Schema Designer, define and configure the following:
    - ◆ e*Ways to transport Events
    - ◆ Event Type Definitions to package the data being transported to an external destination
    - ◆ Collaboration Rules to process Events
    - ◆ Collaborations to apply the required Collaboration Rules
    - ◆ Intelligent Queues (IQ) to which data will be published prior to being sent to the external system
- Activate the test schema.

The following sections describe how create and run a test schema.

7.2 Creating the New Schema

The first task in deploying the sample implementation is to create a new schema name. While it is possible to use the default schema for the sample implementation, it is recommended that you create a separate schema for testing purposes.

To create and test schema, perform the following steps after installing e*Gate Integrator:

1 Start the e*Gate Schema Designer GUI.

2 When the Schema Designer prompts you to log in, select the host that you specified during installation, and enter your password.

3 You will then be prompted to select a schema. Click **New**.

4 Enter **Test_Schema** as a name for the new schema.

The e*Gate Schema Designer opens under your new schema. You are now ready to begin creating the necessary components for it.

The sample schema consists of the following:

- Four e*Ways
- Four Event Types
- Four Collaborations
- Two Intelligent Queues
- Four Collaboration Rules

The sample schema is illustrated in **Figure 1 on page 33**.

Figure 1 Sample Schema
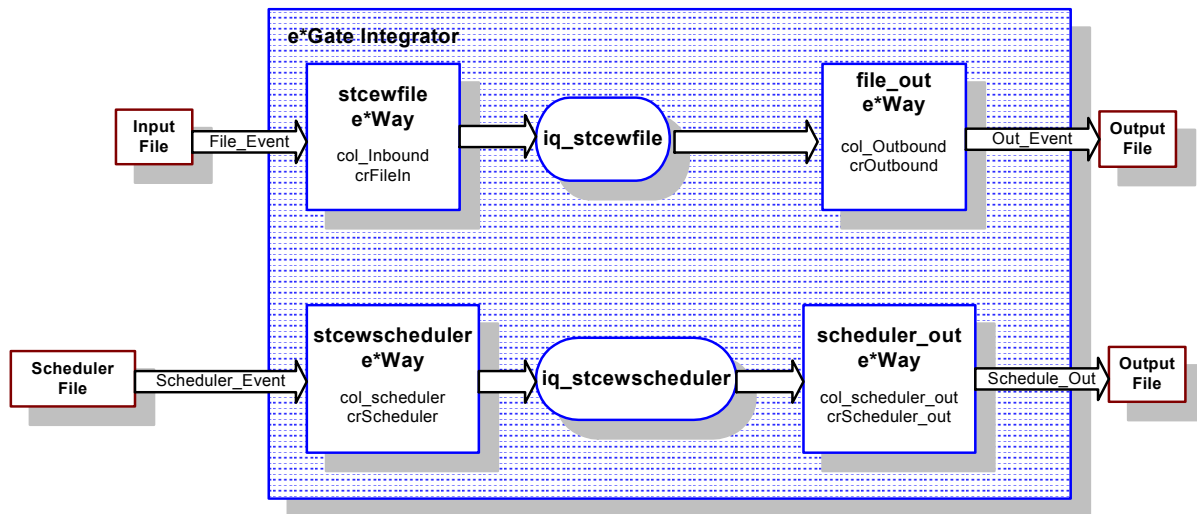
7.2.1 Creating the e*Ways

First, you will create the four e*Ways:

- **stcewfile** - Copies data from one location to another.
- **file_out** - Delivers copied data to its external destination.
- **stcewscheduler** - Processes a scheduled Event.
- **scheduler_out** - Completes the transportation of the scheduled Event.

The following sections provide instructions for creating each e*Way.

Creating the stcewfile e*Way

To create this e*Way

1   On the **Components** tab, double-click the host system, then double-click its Control Broker.

2   On the Palette, click **new e*Way**.

3   Enter **stcewfile** and click **OK**.

4   Double-click **stcewfile** on the **Components** tab.

5   When the e*Way Properties window opens, click **Find** (beneath **Executable File**), and select **stcewfile.exe**.

6   Under **Configuration File**, click **New**.

7   Select **General Settings** from the **Goto Section** menu.

8   Click **YES** under **AllowIncoming**.

9   Click **NO** under **AllowOutgoing**.

10 Select **Poller (inbound) Settings** from the **Goto Section** menu.

11 For **PollDirectory**, type **/egate/data** and press **ENTER**.

12 For **InputFileMask**, type ***.fin** and press **ENTER**.

13 On the **File** menu, click **Save** to save the file as **stcewfile.cfg**.

14 Close the e*Way Configuration Editor window.

15 Close the e*Way Properties dialog box.

Note: *You will be required to create a input file to test the* **stcewfile** *e*Way later in this process. Instructions for doing this are provided in* **"Creating the Input File" on page 43***.*

Creating the file_out e*Way

To create this e*Way

1 On the **Components** tab, double-click the host system, then double-click its Control Broker.

2 On the Palette, click **new e*Way**.

3 Enter **file_out** and click **OK**.

4 Double-click **file_out** on the **Components** tab.

5 When the e*Way Properties window opens, click **Find** (beneath **Executable File**), and select **stcewfile.exe**.

6 Under **Configuration File**, click **New**.

7 Select **General Settings** from the **Goto Section** menu.

8 Click **NO** under **AllowIncoming**.

9 Click **YES** under **AllowOutgoing**.

10 Select **Outbound (send) Settings** from the **Goto Section** menu.

11 For **OutputDirectory**, type **/egate/data** and press **ENTER**.

12 For **Output File Name**, type **Output.dat** and press **ENTER**.

13 On the **File** menu, click **Save** to save the file as **file_out.cfg**.

14 Close the e*Way Configuration Editor window.

15 Close the e*Way Properties dialog box.

Creating the stcewscheduler e*Way

To create this e*Way

1 On the **Components** tab, double-click the host system, then double-click its Control Broker.

2 On the Palette, click **new e*Way**.

3 Enter **stcewscheduler** and click **OK**.

4   Double-click **stcewscheduler** on the **Components** tab.

5   When the e*Way Properties window opens, click **Find** (beneath **Executable File**), and select **stcewscheduler.exe**.

6   For the e*Way's configuration file, you can use the file that was included with e*Gate Integrator, or you can create your own. Under **Configuration File**, do *one* of the following:

  ▪ To use the configuration file that was included with e*Gate Integrator:

    A   Click **Find**

    B   Navigate to **\configs\stcewscheduler**.

    C   Select **stcewscheduler.cfg**.

  ▪ To create your own configuration file:

    A   Click **New.**

    B   Enter a name for the file

    C   Configure it to use a Schedule file that you create, according to the instructions under **"Schedule File Format" on page 24**.

Note: *The **stcewscheduler.cfg** file in the **\configs\stcewscheduler** folder is configured to use the default Schedule file (**\schedules\sample.sch**) that is included with e*Gate Integrator.*

7   Close the e*Way Configuration Editor window.

8   Close the e*Way Properties dialog box.

Creating the schedule_out e*Way

To create this e*Way

1   On the **Components** tab, double-click the host system, then double-click its Control Broker.

2   On the Palette, click **new e*Way**.

3   Enter **schedule_out** and click **OK**.

4   Double-click **schedule_out** on the **Components** tab.

5   When the e*Way **Properties** window opens, click on the **Find** button beneath the **Executable File** field, and select **stcewfile.exe**.

6   Under **Configuration File**, click **New**.

7   Select **General Settings** from the **Goto Section** menu.

8   Click **NO** under **AllowIncoming**.

9   Click **YES** under **AllowOutgoing**.

10 Select **Outbound (send) Settings** from the **Goto Section** menu.

11 For **OutputDirectory**, type **/egate/data** and press **ENTER**.

12 For **Output File Name**, type **output2.dat** and press **ENTER**.

13 On the **File** menu, click **Save** to save the file as **schedule_out.cfg**.

14 Close the e*Way Configuration Editor window.

15 Close the e*Way Properties dialog box.

7.2.2 Creating the Event Types

There are two ways in which to create Event Type Definitions (ETD):

1 Use the ETD Editor to open an existing structure script (***.ssc**) file and modify it.

2 Create an entirely new file from scratch.

For the purposes of this sample implementation, the Event Types in the sample schema use files that are included with the e*Gate Integrator:

- **File_Event** - Inbound data received by the **stcewfile** e*Way.

- **Out_Event** - The inbound data that has been transported to an external location.

- **Scheduler_Event** - Data processed by the **stcewscheduler** e*Way.

- **Schedule_Out** - Data processed by the **stcewscheduler** e*Way and transported to an external destination.

The following sections provide basic steps for creating these Event Types.

Setting the Default Event Type Editor

To set the default editor

1 Select **Default Editor** from the e*Gate Schema Designer's **Options** menu.

2 Select **Monk**.

Creating the File_Event Event Type

To create this Event Type

1 On the **Components** tab, select the **Event Types** folder.

2 On the Palette, click **New Event Type**.

3 Enter **File_Event** and click **OK.**

4 In the Event Type list, double-click **File_Event**.

5 Click **Find**.

6 Navigate to the **\monk_scripts\common** folder and select **GenericInEvent.ssc**.

7 Click **OK**.

Creating the Out_Event Event Type

To create this Event Type

1   On the **Components** tab, select the **Event Types** folder.

2   On the Palette, click **New Event Type**.

3   Enter **Out_Event** and click **OK.**

4   In the Event Type list, double-click **File_Event**.

5   Click **Find**.

6   Select **GenericOutEvent.ssc**.

7   Click **OK**.

Creating the Scheduler_Event Event Type

To create this Event Type

1   On the **Components** tab, select the **Event Types** folder.

2   On the Palette, click **New Event Type**.

3   Enter **Scheduler_Event** and click **OK.**

4   In the Event Type list, double-click **Scheduler_Event**.

5   Click **Find**.

6   Select **Scheduler_Event.ssc**.

7   Click **OK**.

Creating the Scheduler_Out Event Type

To create this Event Type

1   On the **Components** tab, select the **Event Types** folder.

2   On the Palette, click **New Event Type**.

3   Enter **Scheduler_Out** and click **OK.**

4   In the Event Type list, double-click **Scheduler_Out**.

5   Click **Find**.

6   Select **GenericOutEvent.ssc**.

7   Click **OK**.

7.2.3 Creating the Collaboration Rules

The next step is to create the Collaboration Rules that will extract selected information from the source Event Types defined in the previous section. Each Collaboration Rule will then process the extracted information according to its associated Collaboration Service.

There are four Collaboration Rules for this sample schema:

- **crFileIn** - This Collaboration Rule is associated with the **File_Event** Event Type and is used for transporting data.

- **crOutbound** - This Collaboration Rule is associated with the **Out_Event** Event Type and is used for completing the transportation of data to its external destination.

- **crScheduler** - This Collaboration Rule is associated with the **Scheduler_Event** Event Type, and is used for processing data within the Schedule File.

- **crScheduler_out** - This Collaboration Rule is associated with the **Scheduler_Out** Event Type and is used for transporting data within the Schedule File to an external destination.

The following sections provide basic steps for creating these Collaboration Rules.

Creating the crFileIn Collaboration Rule

To create this Collaboration Rule

1 On the **Components** tab, select the **Collaboration Rule** folder.

2 On the Palette, click **New Collaboration Rule**.

3 Enter **crFileIn** and click **OK.**

4 Double-click **crFileIn**.

5 On the **General** tab, select **Pass Through** for the Collaboration Service.

6 On the **Subscriptions** tab, select **File_Event** as the required input Event Type.

7 On the **Publications** tab, select **File_Event** as the default output Event Type.

8 Click **OK**.

Creating the crOutbound Collaboration Rule

To create this Collaboration Rule

1 On the **Components** tab, select the **Collaboration Rule** folder.

2 On the Palette, click **New Collaboration Rule**.

3 Enter **crOutbound** and click **OK.**

4 Double-click **crOutbound**.

5 On the **General** tab, select **Pass Through** for the Collaboration Service.

6 On the **Subscriptions** tab, select **File_Event** as the required input Event Type.

7 On the **Publications** tab, select **Out_Event** as the default output Event Type.

8 Click **OK**.

Creating the crScheduler Collaboration Rule

To create this Collaboration Rule

1 On the **Components** tab, select the **Collaboration Rule** folder.

2 On the Palette, click **New Collaboration Rule**.

3 Enter **crScheduler** and click **OK.**

4 Double-click **crScheduler**.

5 On the **General** tab, select **Pass Through** for the Collaboration Service.

6 On the **Subscriptions** tab, select **Scheduler_Event** as the required input Event Type.

7 On the **Publications** tab, select **Scheduler_Event** as the default output Event Type.

8 Click **OK**.

Creating the crScheduler_out Collaboration Rule

To create this Collaboration Rule

1 On the **Components** tab, select the **Collaboration Rule** folder.

2 On the Palette, click **New Collaboration Rule**.

3 Enter **crScheduler_out** and click **OK.**

4 Double-click **crScheduler_out**.

5 On the **General** tab, select **Pass Through** for the Collaboration Service.

6 On the **Subscriptions** tab, select **Scheduler_Event** as the required input Event Type.

7 On the **Publications** tab, select **Scheduler_Out** as the default output Event Type.

8 Click **OK**.

7.2.4 Creating the Intelligent Queues

The next step is to create and associate an Intelligent Queue (IQ) for the e*Ways. IQs manage the exchange of information between components within the e*Gate system, providing non-volatile storage for data as it passes from one component to another.

IQs use IQ Services to transport data. IQ Services provide the mechanism for moving Events between IQs, handling the low-level implementation of data exchange (such as system calls to initialize or reorganize a database).

This sample Schema uses two IQs, one that processes data received by the **stcewfile** e*Way, and another that handles data processed by the **stcewscheduler** e*Way. The following sections explain how to create these IQs.

Setting the IQ Manager Type

To set the IQ Manager Type

1   On the **Components** tab, double-click the host system, then double-click its Control Broker.

2   Double-click *hosthame_**iqmgr**.

3   From the **IQ Manager Type** drop-down list, select **SeeBeyond JMS**.

4   Click **OK**.

Creating the iq_stcewfile IQ

To create the iq_stcewfile IQ

1   On the **Components** tab, double-click the host system, then double-click its Control Broker.

2   Select *hosthame_**iqmgr**.

3   On the Palette, click **New IQ**.

4   Enter **iq_stcewfile** and click **OK.**

5   Double-click **iq_stcewfile**.

6   On the **Advanced** tab, select **Simple publish/subscribe** under **IQ behavior**.

7   Click **OK**.

Creating the iq_stcewscheduler IQ

To create the iq_stcewscheduler IQ

1   On the **Components** tab, double-click the host system, then double-click its Control Broker.

2   Select *hosthame_**iqmgr**.

3   On the Palette, click **New IQ**.

4   Enter **iq_stcewscheduler** and click **OK.**

5   Double-click **iq_stcewscheduler**.

6   On the **Advanced** tab, select **Simple publish/subscribe** under **IQ behavior**.

7   Click **OK**.

7.2.5 Creating the Collaborations

Collaborations are e*Way components that receive and process Event Types, then forward the output to other e*Gate components. Collaborations consist of the Subscriber, which "listens" for Events of a known type (sometimes from a given source), and the Publisher, which distributes the transformed Event to a specified recipient.

The same Collaboration cannot be assigned to more than one e*Gate component. Each e*Way requires at least one Collaboration.

This sample schema consists of four Collaborations, as follows:

- **col_Inbound** - This Collaboration is a member of the **stcewfile** e*Way and applies the **crFileIn** Collaboration Rule.

- **col_Outbound** - This Collaboration is a member of the **File_Out** e*Way and applies the **crOutbound** Collaboration Rule.

- **col_Scheduler** - This Collaboration is a member of the **stcewscheduler** e*Way and applies the **crScheduler** Collaboration Rule.

- **col_Scheduler_out** - This Collaboration is a member of the **scheduler_out** e*Way and applies the **crScheduler_out** Collaboration Rule.

Details about creating and configuring each Collaboration are in the sections that follow.

Creating the col_Inbound Collaboration

To create this Collaboration

1 On the **Components** tab, double-click the host system, then double-click its Control Broker.

2 In the Editor pane, double-click the **stcewfile** e*Way.

3 On the Palette, click **New Collaboration**.

4 Enter **col_Inbound** and click **OK.**

5 Double-click **col_Inbound**.

6 From the **Collaboration Rules** list, select **crFileIn**.

7 In the **Subscriptions** area, click **Add**.

8 From the **Event Type** list, select the **File_Event** Event Type.

9 From the **Source** list, select <EXTERNAL>.

10 In the **Publications** area, click **Add**.

11 From the **Event Types** list, select the **File_Event** Event Type.

12 From the **Destination** list, select the **iq_stcewfile** IQ.

13 Click **OK**.

Creating the col_Outbound Collaboration

To create this Collaboration

1 On the **Components** tab, double-click the host system, then double-click its Control Broker.

2 In the Editor pane, double-click the **File_Out** e*Way.

3 On the Palette, click **New Collaboration**.

4   Enter **col_Outbound** and click **OK.**

5   Double-click **col_Outbound**.

6   From the **Collaboration Rules** list, select **crOutbound**.

7   In the **Subscriptions** area, click **Add**.

8   From the **Event Type** list, select the **File_Event** Event Type.

9   From the **Source** list, select the **col_Inbound** Collaboration.

10   In the **Publications** area, click **Add**.

11   From the **Event Types** list, select the **File_Out** Event Type.

12   From the **Destination** list, select <EXTERNAL>.

13   Click **OK**.

Creating the col_Scheduler Collaboration

To create this Collaboration

1   On the **Components** tab, double-click the host system, then double-click its Control Broker.

2   In the Editor pane, double-click the **stcewscheduler** e*Way.

3   On the Palette, click **New Collaboration**.

4   Enter **col_Scheduler** and click **OK.**

5   Double-click **col_Scheduler**.

6   From the **Collaboration Rules** list, select **crScheduler**.

7   In the **Subscriptions** area, click **Add**.

8   From the **Event Type** list, select the **Scheduler_Event** Event Type.

9   From the **Source** list, select <EXTERNAL>.

10   In the **Publications** area, click **Add**.

11   From the **Event Types** list, select the **Scheduler_Event** Event Type.

12   From the **Destination** list, select the **iq_stcewscheduler** Intelligent Queue.

13   Click **OK**.

Creating the col_Scheduler_out Collaboration

To create this Collaboration

1   On the **Components** tab, double-click the host system, then double-click its Control Broker.

2   In the Editor pane, double-click the **scheduler_out** e*Way.

3   On the Palette, click **New Collaboration**.

4   Enter **col_Scheduler_out** and click **OK.**

5   Double-click **col_Scheduler_out**.

6   From the **Collaboration Rules** list, select **crScheduler_out**.

7   In the **Subscriptions** area, click **Add**.

8   From the **Event Type** list, select the **Scheduler_Event** Event Type.

9   From the **Source** list, select the **Scheduler** Collaboration.

10   In the **Publications** area, click **Add**.

11   From the **Event Types** list, select the **Scheduler_Out** Event Type.

12   From the **Destination** list, select <EXTERNAL>.

13   Click **OK**.

This completes the schema configuration steps. You are now ready to test the schema.

7.2.6 Executing the Schema

To execute the schema

1   At the command line prompt type the following:

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

2   Press **ENTER**.

3   Start the Schema Manager.

4   When prompted, specify the hostname which contains the Control Broker you started in Step 1 above.

5   Select **Test_Schema**.

6   Verify that the Control Broker is connected (the message in the Control tab of the console will indicate command "succeeded" and status as "up")

7   Right-click the IQ Manager, **hostname_igmgr,** and select **Start**.

8   Right-click each e*Way and select **Start**.

7.2.7 Creating the Input File

After you start the e*Ways, the **stcewfile** e*Way creates the **/egate/data** folder (if it does not already exist. It does this in accordance with the value you entered for the PollDirectory parameter in the **stcewfile** e*Way's configuration file (see **"Creating the stcewfile e*Way" on page 33**).

You need to create an input file and place it in this folder to complete your test of the schema.

Using a text editor, create a file called **test.dat** and place it in this folder.

7.2.8 Expected Results

The e*Way outputs files under the directories you specified for the outbound settings in the configuration files for the **scheduler_out** and **File_out** e*Ways.

Additionally, the file extension of **test.dat** changes to .**~in**, indicating that the file has been received by e*Gate and processed.

Index