SeeBeyond™ eBusiness Integration Suite

e*Way Intelligent Adapter for DataChannel User's Guide

Release 4.5.2

Version 20021029110252.

Contents

Chapter 1

Introduction

This chapter describes how to install and configure the e*Way Intelligent Adapter for the DataChannel Server 4.1, as well as how to create the Java code required to enable the web server to communicate with e*Gate.

1.1 Overview

The DataChannel Server 4.1 uses Hypertext Transfer Protocol (HTTP) to deliver World Wide Web documents to clients using internet browsers such as Internet Explorer or Netscape Navigator. The DataChannel e*Way extends the functionality of the application server by making external data sources available to calls from within Java servlets and Java Server Pages (JSP files).

The DataChannel Server architecture is designed to present flexible portal solutions with open standards-based XML technology. The objective is to make real-time information available to any member of a selected group from any web-accessible device. This architecture can be divided into three specific layers:

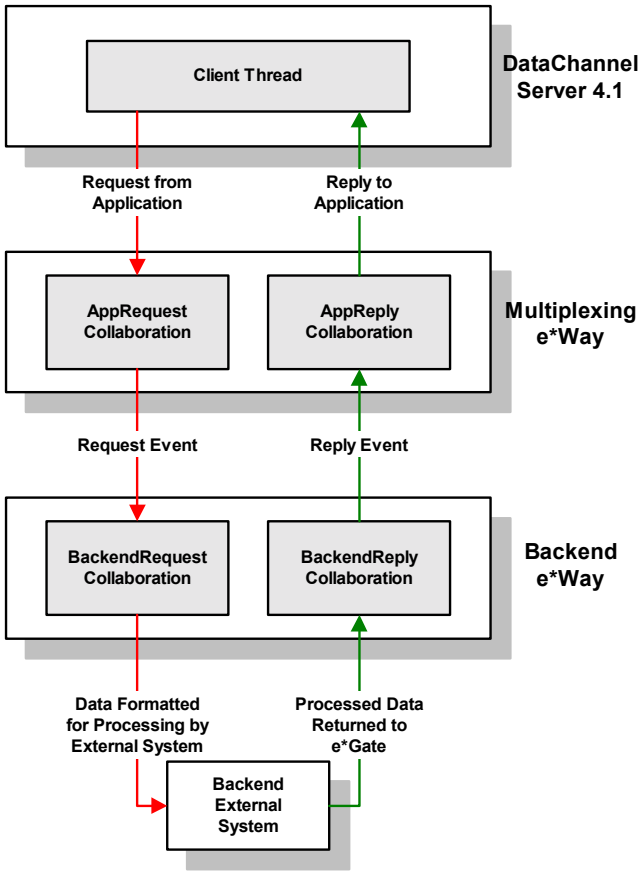
- Data Layer
- Application Layer
- Access Layer

The Data Layer uses the XML Document Object Model (DOM) open standard and DataChannel's API (RIODOM) to manage all portal features and information. The XML information and functionality is exposed by Java servlets and accessed by XSL (Extensible Stylesheet Language) style sheets.

The Application Layer expands DataChannel's APIs to allow custom development for specific tasks and operations. Java and HTML can all be used to access methods and objects managed by RIODOM.

The Access Layer uses style sheets to deliver data from the XML objects. As an example, a request is made through a browser, the XML information is placed into a referenced style sheet and assembled for the browser. Most browsers require standard HTML for viewing. For requests made from a Wireless Access Protocol (WAP) cell phone, the style sheet transforms the information into Wireless Markup Language (WML) as needed for that specific device.

Figure 1 Overview of the DataChannel e*Way implementation



1.1.1 Intended Reader

The reader of this guide is presumed to be a developer or system administrator with responsibility for maintaining the e*Gate system; to have moderate to advanced-level knowledge of operations and administration for the operating system(s) under which the DataChannel Server and e*Gate systems run; to be thoroughly familiar with Java; and to be thoroughly familiar with Windows-style GUI operations.

1.1.2 Components

The DataChannel e*Way comprises the following:

- DataChannel server components: extension packages, and supporting library files
- Participating Host components: a multiplexing e*Way and supporting library files

A complete list of installed files appear in **Table 1 on page 20** and **Table 2 on page 20**.

1.1.3 Supported Operating Systems

The DataChannel e*Way is available on the following operating systems:

- Windows 2000, Windows 2000 SP1, Windows 2000 SP2, and Windows 2000 SP3

- Windows NT 4.0 SP6a
- Solaris 2.6, 7, and 8

1.2 System Requirements

To use the DataChannel e*Way, you need the following:

- An e*Gate Participating Host, version 4.5.1 or higher.
- A TCP/IP network connection.
- Additional disk space for e*Way executable, configuration, library, and script files. The disk space is required on both the Participating and the Registry Host. Additional disk space is required to process and queue the data that this e*Way processes. The amount necessary varies based on the type and size of the data being processed and any external applications performing the processing.

1.2.1 DataChannel Server Requirements

For the DataChannel Server that will communicate with the DataChannel e*Way, you need a client system with the following:

- DataChannel Server version 4.1.
- A valid DataChannel (DCS) account created by an administrator or imported from a directory service.
- Sufficient memory and disk space to support web-server functions. See your *DataChannel Server User's Guides* for more information about server requirements.

Note: *The e*Gate Participating Host may optionally host the web server, but is not required to do so.*

1.3 External Configuration Requirements

Please see the DataChannel Documentation for detailed information regarding detailed DataChannel Configuration.

1.3.1 Creating a New Request Handler

The new request handler must implement the following methods:

```
public void initialize (DispatchMananger dispMgr)
    throws InitializationException;
public void handle (HTTPServletRequest req, HttpServletResponse)
    throws DispatchException;
```

In the initialize method, you must initialize your handler's DispatchManager variable with the supplied DispatchManager object. The handle method should be used to process the request and format the query in order for it to be sent to the e*Gate system. This method will be invoked whenever a query is made in the UI.

1.3.2 Registering a New Request Handler

New handlers need to be added to dispatch.xml as part of the XperLinX installation. The DCS instance needs to be restarted in order for this to take effect. The registration adds a line in dispatch.xml. For example:

```
<HANDLER multi_dispatch_type=”get_customer” name=”get_customers”
class=”com.datachannel.merlin.dispatch.handlers.multiDispatchHandlerB
as
<HANDLER name=”egate”
class=”com.datachannel.merlin.dispatch.handlers.egate.northwind_demo.
get_customers”>
</HANDLER>
```

1.3.3 Notes on formatting the data to be sent to e*Gate

e*Gate identifies each request by a parameter referred to as the RoutingTag. Based on this parameter's value, the e*Gate system decides the destination to which to route this request. This parameter has to be obtained from the e*Gate implementer and sent to the e*Gate system along with the specified request.

1.3.4 Installation Overview

It is recommended that the installation take place in the following order:

1. Install the e*Gate Registry, Participating Hosts, DataChannel e*Way and any other e*Ways necessary to the desired schema.
2. Install DataChannel Server 4.1.x
3. Install the XpertLinX. This will install the jar, register the handlers and register the themes associated with this XpertLinX.

1.3.5 Installing a special theme for an e*Gate based XpertLinX

The following example walks you through the steps necessary to install a new theme. In this example, the new theme is the autoelectronics theme.

1. Verify that the e*Gate XpertLinX folder is accessible via HTTP. For example:

```
http://<hostname>/xpertlinx/egate/install.xml
```

2 Verify that the theme folder has been copied to the local files system. For example:

```
<webroot>\dcs\layouts\autoelectronics
```

3 Go to <webroot>\dcs\layouts\autoelectronics\module_includes.xslt. Verify that it appears similar to the section below:

```
<!DOCTYPE wyldstallyns SYSTEM "wyldstallyns.dtd">
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:java="http://xml.apache.org/xslt/java" xmlns:dcs="dcs"
extension-
element-prefixes="java, dcs">
    <xsl:include xml:link="include" href="m_default.xslt" />
    <xsl:include xml:link="include" href="m_start_folder/start.xslt"
/>
</xsl:stylesheet>
```

No additional #include statements are necessary.

1.3.6 Configuring the DCS System to Run the Autoelectronics Theme

Follow the steps below to configure the “new” theme, in this case, the autoelectronics theme.

1 Login as admin.

2 Click on "Theme Management" link.

3 On the toolbar, click "Add Theme".

4 Under TITLE enter a title.

5 Under BASE SERVER-RESOURCES enter:

```
"file:/<drive letter>:/<web root>/dcs/layouts/autoelectronics/".
```

6 6. Under BASE CLIENT-RESOURCES enter

```
"/dcs/layouts/autoelectronics/".
```

The other 2 fields are optional.

7 Verify that the "Can display XpertLinX" option is checked.

8 Click apply, you will then go back to the theme management page.

9 Click the autoelectronics theme radio button.

1.3.7 Installing styles for the theme

Follow the steps to install styles for the theme.

1 While still at the theme management page, click on autoelectronics.

2 On the toolbar, click "Add Style".

3 Under TITLE, enter "default". Under STYLE RESOURCES enter "default/". DESCRIPTION is optional.

4 On the toolbar, again click "Add Style".

5 Under TITLE, enter "autologistics". Under STYLE RESOURCES enter "autologistics/". DESCRIPTION is optional.

6 6. Click the autologistics style radio button.

1.3.8 Installing eGate XpertLinX

Follow the steps to install the e*Gate XpertLinX.

1 On the main toolbar, click "DCS Control Panel".

2 Click "XpertLinX Management".

3 Click "Add XpertLinX".

4 Select "Download using configuration file:" radio button.

5 In the text field type the path to the XpertLinX install.xml file (e.g. "http://<path to egate xpertlinx folder>/install.xml")

6 In the HOST text field type the name of the machine the e*Gate software is on.

7 Under "Add to these themes" select ONLY the autoelectronics theme.

8 Click "Add".

1.3.9 Finalizing Configuration

1 Click "Logout".

2 Log in as a user (user profiles are set up while logged in as admin).

3 Click "Modify Start Page".

4 Click "Add or Remove XpertLinX".

5 Select check-box for the egate XpertLinX.

6 Click "Apply". This will return to the "Modify Start Page" page. Position the XpertLinX if you like.

7 Click "Apply". This returns you to the start page.

8 Click the link in the XpertLinX module on the start page to set up the user profile (you won't have to do this again).

Troubleshooting theme related problems:

- Verify the XpertLinX install process created:
  <web root>\dcs\layouts\autoelectronics\xpertlinx\datachannel\egate
- Verify that there are templates in the egate folder
- Verify that the correct host name is specified in the e*Gate external properties. Check the install file you used to be sure.

Chapter 2

Installation

This chapter covers the requirements for installing the DataChannel e*Way and how to configure the DataChannel server components needed. A list of the files and directories created by the installation is also provided.

2.1 Supporting Documents

The following documents are designed to work in conjunction with the *e*Way Intelligent Adapter for DataChannel User's Guide* and to provide additional information that may prove useful to you.

- *e*Gate Integrator Installation Guide*
- *e*Gate Integrator System Administration and Operations Guide*.
- *README.txt* file on the e*Gate installation CD ROM.

2.2 Pre-installation

The following terminology will be used in this section:

<unzip_root>
The root directory where the Patch Release file is unzipped.

<install_root>
The root directory where the DCS server files are installed. The default location for Windows NT systems is C:\DataChannel.

<web_root>
The root document directory of your Web server software. The default depends upon the Web server being used. For example, the default location for Microsoft Internet Information Server (IIS) is C:\InetPub\wwwroot.

Note: *Adjustment to the below instructions must be made for the direction of the slashes "/", depending upon the operating system upon which the installation is performed.*

2.2.1 Installation Overview

The process for installing the SeeBeyond e*Way on Windows NT computers running DCS 4.1 are:

1 Prepare system for installation

   A Copy and decompress the package.

2 Stop the Jserv servlet engine.

   A In the Windows Control Panel, double-click the Services applet icon.

   B Stop the “JServ Service”.

   C After the JServ service stops, click the Close button.

3 Create a backup of the existing DCS files.

   A Locate the DCS install directory (<install_root>). The default <install_root> is “C:\DataChannel”. These are the “server” files.

   B Copy all files and subdirectories in the “<install_root>” directory (a “recursive” copy) to your desired backup location or media. Be sure to preserve the directory structure of the backed-up files.

   C Locate the DCS Web server directory. The default location is “<web_root>\dcs” where <web_root> is the root directory of your Web server. For example, the default <web_root> for Microsoft Internet Information Server (IIS) is “C:\InetPub\wwwroot”; the default <web_root> for Netscape Enterprise Server (NES) is “C:\Netscape\SuiteSpot\docs”. These are the “Web server” or “theme” files.

   D Copy all files and subdirectories in the “dcs” directory (a “recursive” copy) to your desired backup location or media. Be sure and preserve the directory structure of the backed-up files.

4 Copy the required program files to the DCS server install root.

   This will place the DCS/eGate binary files in the program locations for use by DCS.

   A Recursively copy the contents of the ‘datachannel’ directory, from the uncompress directory, to the DCS <install_root> directory, updating any existing files. For example, copy “<unzip_root>\datachannel*.*” to “C:\datachannel*.*

5 Modify the CLASSPATH to include the new egate.jar entry.

   A Open the NT Registry Editor by clicking ‘Start’ to open the NT Start Menu, click ‘Run..’, enter ‘regedit’ in the text box. Click ‘OK’ to open the registry editor.

   B Locate the HKEY_LOCAL_MACHINE\SOFTWARE\DataChannel\Jserv registry key.

   C Right-click the “Classpath” value and click “Modify”.

   D At the beginning of this value, but after the “-cp” , enter the string “<install_root>/classes/egate.jar;” string, replacing <install_root> with the location of the DCS install. For example, the new string should begin similar to the following:

```
'..-cp C:/datachannel/classes/egate.jar;C:/detach~1/classes/
merlin.jar;...'
```

Note: *The ' ' (single quotes) represent additional text not shown in this example.*

E Click "OK", and exit out of the NT registry editor.

F Locate the "<install_root>/config" directory.

G In a text editor, open the "rioconfig.xml" file.

H Locate the string "-cp" parameter and insert the string "<insert_root>/classes/egate.jar;" prior to the existing classpath entry, replacing <install_root> with the location of the DCS install. For example, the new string should begin similar to the following:

```
'..-cp C:/datachannel/classes/egate.jar;CL/detach~1/classes/
merlin.jar;...'
```

Note: *The ' ' (single quotes) represent additional text not shown in this example.*

6 Copy new web files to the DCS server wwwroot directory.

A Recursively copy the 'wwwroot' directory from the uncompress directory to the DCS Web files directory. For example, copy "<unzip_root>\wwwroot\dcs*.*" to "C:\InetPub\wwwroot\dcs*.*" for (IIS) or to "C:\Netscape\SuiteSpot\docs\dcs*.*" for NES.

7 Modify the 'install.xml' file located in the <webroot>/xpertlinx/egate/directory.

A Locate the "<web_root>/xpertlinx/egate/" directory.

B In a text editor, open the "install.xml" file.

C Locate all instances of the string "DCS_Hostname" and replace it with the name of the local DCS Server name.

D Save and close the file.

8 Restart the Jserv servlet engine.

A In the Windows Control Panel, double-click the Services applet icon.

B Start the "JServ Service".

C After the JServ service starts, click the Close button.

9 Register the newly installed AutoElectronics Theme and set it to 'default' by performing the following steps:

A Login as the administrative user.

B Click on the "Theme Management" link.

C On the toolbar, click "Add Theme".

D In the TITLE input field, enter 'Autoelectronics' to name the new theme.

E In the SERVER-RESOURCES field, enter "file:/<web_root>/dcs/layouts/autoelectronics/". For example, "file:/C:/Inetput/wwwroot/dcs/layouts/

autoelectronics/" if the web root directory is located on "C:\Inetput\wwwroot".

F In the BASE CLIENT-RESOURCES field, insert "/dcs/layouts/autoelectronics/".

G Verify the "Can display XpertLinX" option is checked.

H Click "Add" to save the changes and return to the Theme Management page.

I Make the 'Autoelectronics' them the default them by selecting the related radio button.

10 Add the SeeBeyond/Siebel XpertLinX to the Autoelectronics theme.

A On the main toolbar, click "DCS Control Panel".

B Click "XpertLinX Management".

C Click "Add XpertLinX".

D Select "Download using configuration file:" radio button.

E In the text field type the path to the XpertLinX install file (e.g. "http://<DCS_Hostname>/xpertlinx/egate/install.xml"), replacing <DCS_Hostname> with the name of the local DCS server.

F Click "Continue" to begin the e*Gate XpertLinX installation.

G In the HOST text field type the hostname of the e*Gate machine for which DCS will use for the integration, replacing the default value of 'stcserver.domain.com'.

H Under "Add to these themes" select ONLY the 'Autoelectronics' theme.

I Click "Add" to complete the installation of the theme.

11 Create and configure a user view profile from e*Gate.

A On the main toolbar, click "DCS Control Panel".

B Click 'Account Management' under the 'General Tools' section heading on the DCS Control Panel.

C Click 'Add a new user'. If more than one directory service exists on the system, select the appropriate directory service to create the user within and then click 'Add a new user'.

D Enter the desired values for this new user.

E Click 'Add User' to save the settings and create the new user.

F From the 'Existing People' list on this page, find the newly created user account and click on the name to open the account profile.

G Set the user's default theme to 'Autoelectronics' and click 'Apply' to save the change.

H "Logout" as the administrative user.

I Log in as the user created in step (D) above.

J Click "Modify Start Page".

K Click "Add or Remove XpertLinX".

L Select check-box for the e*Gate XpertLinX

M Click "Apply" to save and return to the "Modify Start Page" page. Position the XpertLinX if necessary.

N Click "Apply" to save the layout and return to the user's start page.

O Click the link in the XpertLinX module on the start page to set up the user profile.

P Click the "View Profile" button to view the user's Siebel profile, as read-only.

Q Click the "Edit" button or "Personalize" from the page menu to edit the user's Siebel profile.

R Edit input fields as desired.

S Click the "Submit" button to save changes and return to the read-only view of the user's Siebel profile.

The preliminary installation is complete.

2.3 Installing the DataChannel e*Way

The DataChannel e*Way must be installed on the machine on which the e*Gate Participating Host is installed. Several files must be copied onto the machine on which the DataChannel Server is installed (if different from the Participating Host). (See **Table 2 on page 20** for details.)

Note: *The DataChannel e*Way Client components can only be installed on a system where the DataChannel Server is installed.*

2.3.1 Pre-installation

- Exit all programs before running the setup program, including any anti-virus applications.
- You do not require root privileges to install this e*Way. Log in under the user name that you wish to own the e*Way files. Be sure that this user has sufficient privilege to create files in the e*Gate directory tree.

2.3.2 Installation Procedure

If you are installing this e*Way as part of a complete e*Gate installation, please follow the instructions in the *e*Gate Integrator Installation Guide*. The DataChannel e*Way is installed as an "Add-on" component in the fourth phase of the installation.

If you are adding the DataChannel e*Way to an existing e*Gate installation, follow the instructions below.

To install the DataChannel e*Way Components on a Windows NT system:

1. Log in as an Administrator on the workstation on which you want to install the e*Way.
2. Close any open applications.
3. Launch the setup application on the e*Gate installation CD-ROM.
4. Follow the online prompts in the InstallShield® Wizard. When the **Select Components** dialog box appears, clear all the check boxes except **Add-ons**. Click **Next** as necessary to proceed through the setup application.
5. When the **User Information** dialog box appears, type your name and company name.
6. When the **Choose Destination Location** window appears, **do not** change the **Default Destination** folder unless you are directed to do so by SeeBeyond support personnel; simply click **Next** to continue.
7. When the **Select Components** dialog box appears, check **DataChannel e*Way.**
8. When the **Registry Hostname** window appears, enter the name of the machine on which the Registry files are installed.
9. Complete the fields in the **Administrator Account Information** dialog box.

Note: *e*Gate usernames and passwords are case-sensitive.*

10. When the **Select platforms to support for add-on** window appears, select **Windows NT service pack 4 and higher** if it is not already selected.
11. Follow the on-screen prompts to complete the installation.

Note: *If you install e*Gate add-ons on a Windows 98 system, the installation program may open a series of DOS windows. If this occurs, simply close the DOS windows, and the installation program will proceed normally.*

After the installation is complete, reboot the computer and launch the Enterprise Manager.

To install the DataChannel e*Way on a UNIX system:

1. Log on to the workstation on which you want to install the e*Way. If you do not wish to log in as root, you must log in as a user with sufficient privilege to install files in the "egate" directory tree.
2. Insert the CD-ROM into the drive.
3. If necessary, mount the CD-ROM drive. On HP-UX systems, you must mount the drive with this command:

   /etc/mount -F cdfs -o cdcase /dev/cdrom /cdrom

   where **/cdrom** is the mount point.
4. At the shell prompt, type:

   cd /cdrom/setup
5. Start the installation script by typing:

setup.sh

6 If you are not running as root, you will see a message notifying you that services will not start automatically for non-root users. Press **Enter** to continue.

7 You will see a message confirming that you are running the e*Gate installation script, and reminding you that you can type - (hyphen) to back up a step or **QUIT** (all capitals) to exit the install program. Press **Enter** to continue.

8 You will be prompted to accept the license agreement. Type **y** and press **Enter**.

The platform type and a menu of options will display:

```
Installation type (choose one):
  0. Finished with installation.  Quit.
  1. e*Gate Addon Applications
  2. e*Gate Participating Host (Client)
  3. e*Gate Registry Server
```

9 Type 1 to select the **e*Gate Add-on Applications** and press **Enter**.

10 You will be prompted for the installation path. Press **Enter** to accept the default path, or enter a new path and press **Enter**.

- If you are logged in as root, the suggested path will be **/opt/egate/client**.
- If you are logged in under any other user name, the suggested path will be **/home/*username*/egate/client**.

Whether you install e*Gate to a **/home** directory to an application directory such as **/opt**, we strongly recommend that you use the recommended relative path "egate/client" as the destination directory for the add-on-application installation.

11 When prompted, type **U** to update (overwrite) and press **Enter**.

Note: ***U*** *updates the installation, overriding files as necessary.* ***M*** *creates a directory and moves everything in the current directory to* ***directoryname.old****.*

If you selected "U," you will see a warning regarding shared EXE and DLL files. Read this warning and press **Enter** to continue.

12 Enter the name of the Registry Server supporting these add-on applications. If the installation utility detects a Registry Host running on the current server, it will suggest that host's name.

13 You will be prompted for the "administration login" (an e*Gate user with sufficient privilege to create components within a schema). The default is **Administrator**; unless you have created a different "administrative" user name, press **Enter** to accept the default. The default password is listed in the README.TXT file in the root directory of the installation CD-ROM.

14 Enter and confirm the password for the user specified in the step above.

Note: *e*Gate user names and passwords are case-sensitive.*

15 A menu of add-on options will appear. Type the number corresponding to the add-on package(s) you wish to install (**DataChannel e*Way**) and press **Enter**.

16 Follow the on-screen instructions to complete the installation.

17 After the add-on application has been installed, the "Choose add-on packages" menu will appear. Repeat step 15 to install additional packages, or **0** and press **Enter** to continue.

18 When the "installation type" menu appears, the Add-on Applications installation is complete. Do one of the following:

- To exit the setup utility, type **0** and press **Enter.**
- Select another option to continue the installation.

2.3.3 Configuring the Participating Host Components in the Enterprise Manager

Important

From the perspective of the e*Gate GUIs, the DataChannel e*Way is not a system of components distributed between the DataChannel Server and a Participating Host, but a single component that runs an executable file (the multiplexer **stcewipmp.exe**). When this manual discusses procedures within the context of any e*Gate GUI, the term "e*Way" refers only to the Participating Host component of the DataChannel e*Way system.

To configure the Participating Host components:

1 If you have not already done so, launch the Enterprise Manager.

2 Using the Component editor, create a new e*Way.

3 Display the new e*Way's properties.

4 On the General tab, under **Executable File**, click **Find**.

5 Select the file **stcewipmp.exe**.

6 Click **OK** to close the properties sheet, or continue to configure the e*Way. Configuration parameters are discussed in **Chapter 3**. The setup and requirements of schemas required to use this e*Way are discussed in **Chapter 5**.

Note: *Once you have installed and configured this e*Way, you must incorporate it into a schema by defining and associating the appropriate Collaborations, Collaboration Rules, IQs, and Event Types before this e*Way can perform its intended functions. For more information about any of these procedures, please see the online Help system.*

*For more information about configuring e*Ways or how to use the e*Way Editor, see the **e*Gate Integrator User's Guide**.*

2.4 Files/Directories Created by the Installation

The DataChannel e*Way installation process will install the following files within the e*Gate directory tree. Files will be installed within the "egate\client" tree. On the Participating Host, files will be committed to the "default" schema on the Registry Host.

Table 1 Participating Host System files installed

e*Gate Directory	File(s)
bin\	stcewipmp.exe stc_common.dll stcewipmpclnt.dll
\configs\stcewipmp\	stcewipmp.def

Table 2 lists the files that must be installed on the system running the DataChannel Server.

Table 2 DataChannel Server files

Client Directory	File(s)
bin\	stc_common.dll stc_ewipmpclnt.dll
classes\	stcph.jar

Note: *The user-defined files must be copied and moved to the appropriate DataChannel Server directory. Please see* **"Using the Java Servlet" on page 36** *for more information on deploying the JSP's and Servlets.*

The files in Table 2 must be copied to the DataChannel Server Machine. The DataChannel Server classpath must include **stcph.jar**. Please refer to the DataChannel documentation for more information regarding the recommended location to which to copy this file.

Once the files have successfully been copied onto the DCS and any applicable classpath adjustments have been made, the DCS must be stopped and restarted to insure that the new class information is recognized.

Chapter 3

Configuration

Important: *From the perspective of the e*Gate GUIs, the DataChannel e*Way is not a system of components distributed between the web server and a Participating Host, but a single component that runs an executable file (the multiplexer **stcewipmp.exe**). When this manual discusses procedures within the context of any e*Gate GUI (such this chapter, which deals in part with the e*Way Editor), the term "e*Way" refers only to the Participating Host component of the DataChannel e*Way system.*

3.1 e*Way Configuration Parameters

e*Way configuration parameters are set using the e*Way Editor.

To change e*Way configuration parameters:

1. In the Enterprise Manager's Component editor, select the e*Way you want to configure and display its properties.
2. Under **Configuration File**, click **New** to create a new file, **Find** to select an existing configuration file, or **Edit** to edit the currently selected file.
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. Be careful not to change any of the default arguments unless you have a specific need to do so.

For more information about how to use the e*Way Editor, see the e*Way Editor's online Help or the *e*Gate Integrator User's Guide*.

The e*Way's configuration parameters are organized into a single section: General Settings.

3.1.1 General Settings

The parameters in this section specify the name of the external client system and the IP port through which e*Gate and the client system communicates.

Request Reply IP Port

Description

Specifies the IP port that the e*Way will listen (bind) for client connections. This parameter is used for Request/Reply behavior.

Required Values

A valid TCP/IP port number between 1 and 65536. The default is **26051**. Normally, you only need to change the default number if the specified TCP/IP port is in use, or you have other requirements for a specific port number.

Push IP Port

Description

Specifies the IP port through which this e*Way allows an external system to connect and receive unsolicited (without submitting a request) Events.

Required Values

A valid TCP/IP port number between 0 and 65536. The default is **0**.

Additional Information

Any Event that this e*Way receives that has zero values for all fields in the 24 byte MUX header is sent to all callers of the **WaitForUnsolicited**. This parameter is optional. If set to zero, the e*Way will follow the Request/Reply scenario and not accept unsolicited Events.

Rollback if no Clients on Push Port

Description

Specifies whether the Event will continually roll back if there are no push clients connected.

Required Values

Yes or **No**. If set to **Yes**, the Event will continually roll back if there are no push clients connected.

Wait For IQ Ack

Description

Specifies whether the send client function does NOT return until the Event is committed to the IQ.

Required Values

Yes or **No**. If set to **Yes**, the send client function does NOT return until the Event is committed to the IQ.

Caution: *This parameter should be set if the data must be committed to the IQ on every transaction before the API returns to the client. Setting this parameter to Yes will*

significantly impact performance. If normal request/reply type transactions are being sent/received, and the data can be recreated at the client, this parameter should not be set.

Send Empty MSG When External Disconnect

Description

Specifies whether the e*Way sends an empty incoming message (containing only the multi-plexer header) when an external client disconnects.

Required Values

Yes or **No**. If set to **Yes**, the e*Way sends an empty incoming message when an external client disconnects.

MUX Instance ID

Description

Specifies whether the specified 8 (eight) bytes is prepended to the 24 (twenty-four) byte session ID of the request received from the external connection before sending to e*Gate.

Required Values

A string. If this value is other than "0", the 8 bytes are prepended to the 24 byte session ID. The default is **0**.

Note: *This is a string where "00" and "00000000" are valid MUX Instance IDs, while "0" is to turn this option off. Only the first 8 bytes are used.*

MUX Recovery ID

Description

Specifies whether the 8 bytes are prepended to the reply and republish back to e*Gate provided the value is other than "0" and the multi-plexer finds that the session related to the MUX ID in the return message has been dropped.

Required Values

A string. If this value is other than "0", the 8 bytes are prepended to the 24 byte session ID. The default is **0**.

Note: *This is a string where "00" and "00000000" are valid MUX Recovery IDs, while "0" is to turn this option off. Only the first 8 bytes are used.*

3.2 External Configuration Requirements

To enable the client system to communicate with the multi-plexer component of the e*Gate API Kit, you must do the following:

1. Install the required client files on the external system (see **"Files/Directories Created by the Installation" on page 20**).
2. Configure the client components as necessary to use the TCP/IP port specified above in **"Push IP Port" on page 22**.

Chapter 4

APIs

4.1 com.datachannel.egate

This class defines the set of methods used by DCS to create a user defined number of MUX (multi-plexer) connections to e*Gate and send/receive Event(s) to e*Gate:

connect

Syntax

```
connect()
```

Description

connect opens a connection to the Participating Host that is running the MUX e*Way.

Parameters

None.

Return Values

void

Throws

@exception eGateConnectionException is thrown if an error occurs with connection.

disconnect

Syntax

```
disconnect()
```

Description

disconnect closes the connection to the Participating Host that is running the MUX e*Way and waits for each connection to finish the associated transaction.

Parameters

None.

Return Values

void

Throws

@exception eGateConnectionException is thrown if an error occurs with connection.

sendMessageToeGate

Syntax

```
sendMessageToeGate(bytes_array)
```

Description

sendMessageToeGate takes the message (Event) that is to be delivered into e*Gate.

Parameters

Name	Type	Description
bytes_array	byte array	The message (Event) as a byte array

Return Values

void

Throws

@exception eGateConnectionException is thrown if an error occurs while communicating with e*Gate.

sendMessageToeGate

Syntax

```
sendMessageToeGate(message)
```

Description

sendMessageToeGate takes the message (Event) that is to be delivered into e*Gate.

Parameters

Name	Type	Description
message	string	The message (Event) to send

Return Values

void

Throws

@exception eGateConnectionException is thrown if an error occurs while communicating with e*Gate.

getResponseFromeGate

Syntax

```
getResponseFromeGate()
```

Description

getResponseFromeGate polls the e*Gate system and returns a response if available.

Parameters

None.

Return Values

string

Returns the response from e*Gate.

Throws

@exception eGateConnectionException is thrown if an error occurs while communicating with e*Gate.

Chapter 5

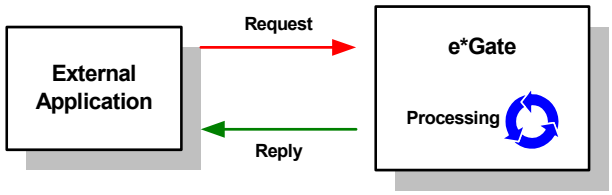
Implementation

5.1 The Request/Reply Concept

All the applications of the DataChannel e*Way are based upon the "Request/Reply" concept. At a high-level, this works as follows:

1. The web server submits data (a **request)** to the e*Gate system.
2. The e*Gate system processes the data as required, communicating with other external ("backend") systems as necessary.
3. The e*Gate system returns data (a **response)** to the same thread within the web server that submitted the request.

Figure 2 The Request/Reply concept

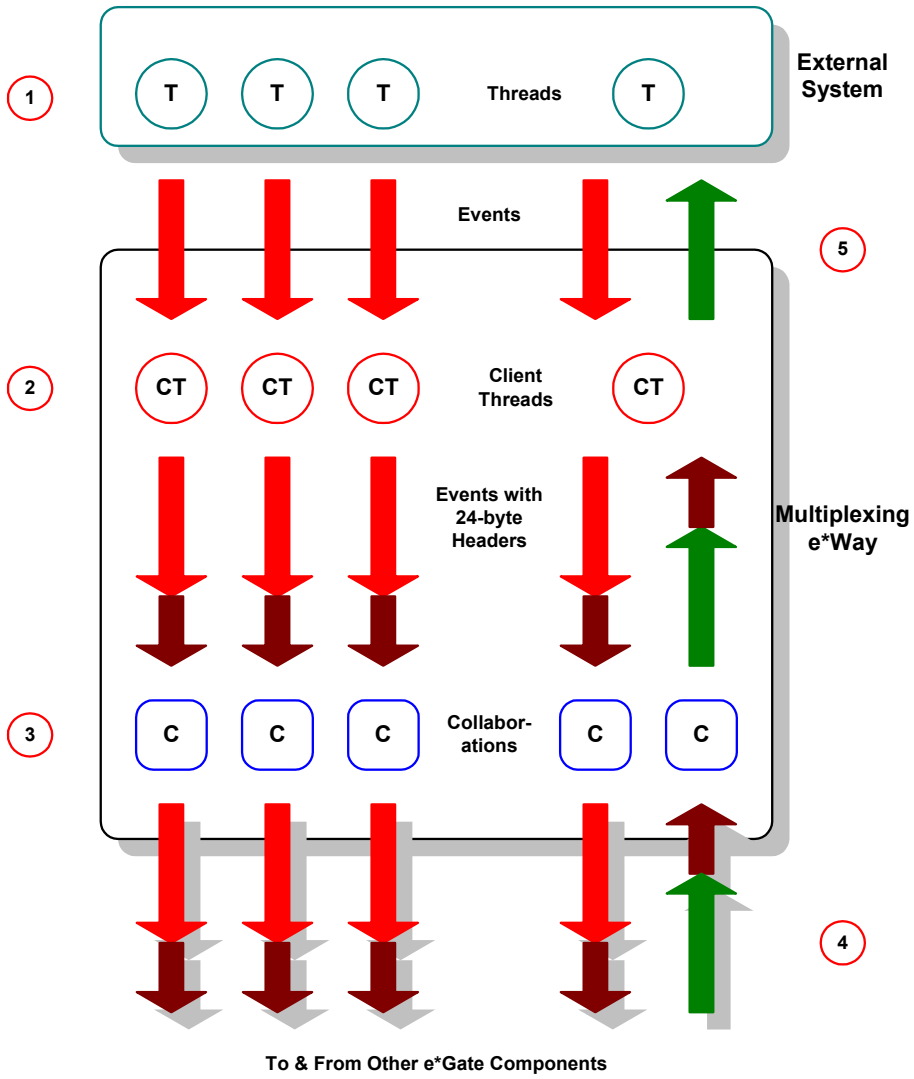


5.1.1 Request/Reply and the DataChannel e*Way Participating Host Components

The DataChannel e*Way Participating Host component is a multiplexing e*Way that uses a proprietary IP-based protocol to multi-thread Event exchange between the e*Way and external systems or other e*Gate components.

Figure 3 illustrates how the multiplexing e*Way receives data from an external application and returns processed data to the same application.

Figure 3 Data flow through the multiplexing e*Way



1 The DataChannel Server uses Java Servlets to send the data to the multiplexing e*Way using an SeeBeyond-proprietary IP-based protocol.

2 Client threads within the e*Way package the data as e*Gate Events, adding a 24-byte header. Among other functions, this header provides "return address" information that can optionally be used to return data to the client thread that originated it.

   Each e*Way can handle up to 1,000 client threads at once. If your requirements demand more processing power, you can define more DataChannel e*Ways.

3 Collaborations within the e*Way perform any appropriate processing that may be required, and route the processed Events to other destinations (such as an external system for additional data retrieval or processing).

Note: *The 24-byte header **must** be preserved as the Events are processed through the e*Gate system.*

4 Processed data, still containing the original 24-byte header, is returned to the DataChannel e*Way.

5 The e*Way uses the 24-byte "return address" to identify the destination of the data to be returned to the external system.

6 The e*Way returns the data, minus the 24-byte header, to the client thread within the web server.

5.1.2 The Request/Reply Schema

Request/Reply schemas have two classes of components:

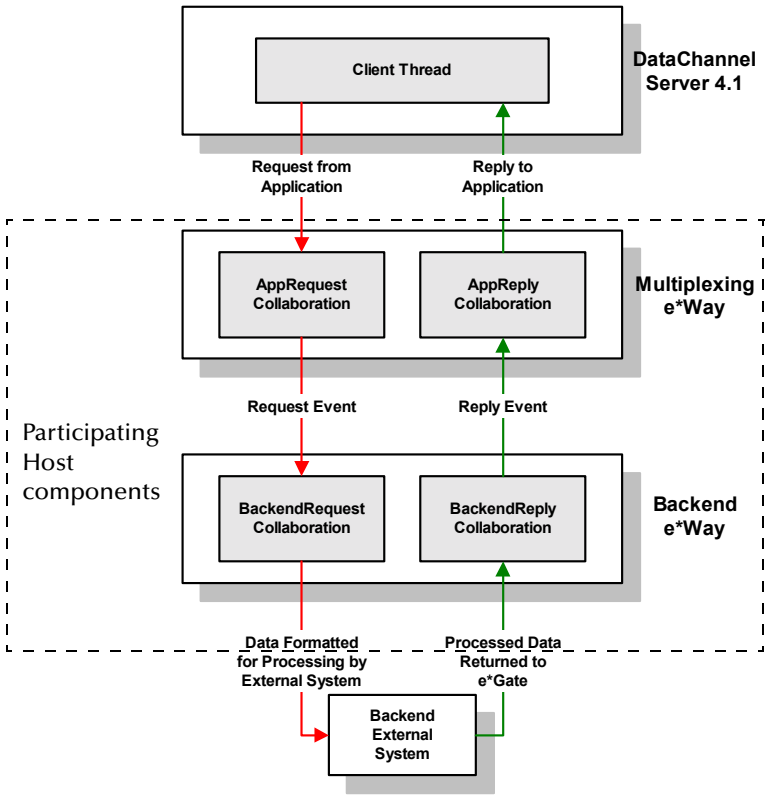
1 “Front end” components that handle communications with the web server. These components receive requests and route replies to the correct destination.

2 “Back end” components that process the requests and compose the replies. These components also provide the bridge between the e*Gate system and your existing systems.

The DataChannel e*Way and its related Collaborations comprise the front-end components. A second e*Way and its related Collaborations comprise the back-end components (more e*Ways may be added to communicate with more external systems as required). The backed e*Way(s) can be of any type required to communicate with the external system(s).

Figure 4 below illustrates a typical Request/Reply schema.

Figure 4 The Request/Reply schema



Implementation using DataChannel Server

1. The user submits data to the web server via a browser form.
2. The web server sends the request information to DCS.
3. DCS packages the request and invokes the e*Gate extension subroutines, and forwards the data to the Participating Host.
4. Data enters the Participating Host through the multiplexing e*Way's WebRequest Collaboration.
5. The WebRequest Collaboration publishes the Request Event.
6. The BackendRequest Collaboration within the back-end e*Way subscribes to the Request Event, and processes the data as appropriate, and routes it to the external backend system.
7. The external backend system processes the data and returns it to e*Gate via the Backend e*Way.
8. The Backend e*Way's BackendReply Collaboration publishes the data as the Reply Event.
9. The WebReply Collaboration within the multiplexing e*Way subscribes to the Reply Event.

10 The multiplexing e*Way returns the processed data to DCS.

11 DCS builds a response and passes it to the server.

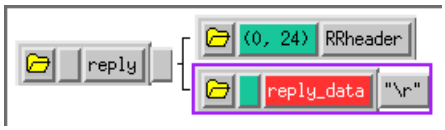
12 The web server sends the response back to the client.

5.1.3 ETDs and Form Data

As discussed in **"Request/Reply and the DataChannel e*Way Participating Host Components" on page 31**, the DataChannel e*Way maintains "return address" information in a 24-byte header that must be preserved as the data flows through the e*Gate system.

The simplest Event Type Definition (ETD) that can be used within a Request/Reply schema has two nodes: one for the header, the second for the remainder of the Event data.

Figure 5 The simplest Request/Reply ETD



This ETD is sufficient if you wish to send data through the e*Gate system simply as a blob. For example, you can compose the reply to the web server as a block of completely formatted HTML code, then return that reply using the above ETD.

Although the simple ETD in Figure 5 can be sufficient for reply data, request (input) data will probably have a more complex structure. To accommodate this, you must add the additional structure to the basic Request/Reply ETD:

1 Be sure to maintain the 24-byte "header" node unchanged.

2 Add one subnode beneath the "data" node for each element of input data.

3 Modify those subnodes as necessary (for example, to use appropriate delimiters or record lengths).

Figure 6 below illustrates an ETD that describes delimited data (for example, as in the data "First name^Last name").

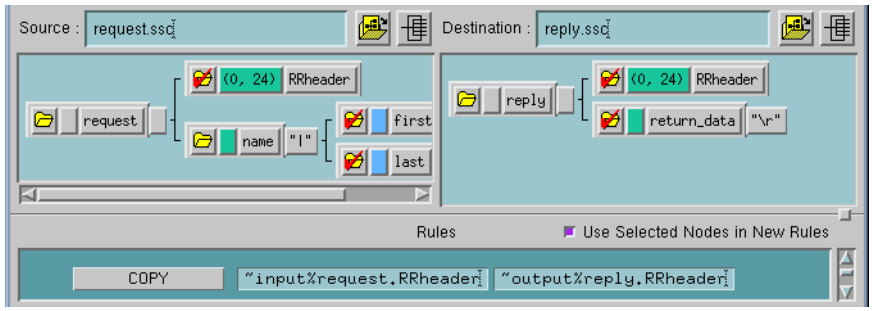
Figure 6 A Request/Reply ETD for delimited data

Of course, the more complex the form that collects user data, the more complex the ETD that describes the form data must become. Be sure that the Request ETD meets the requirements for input to the backend system.

Collaboration Rules and the DataChannel Header

Collaboration Rules that manipulate data between ETDs must preserve the Request/Reply header (in the figures above, "RRheader"). Be sure that each Collaboration Rule that manipulates Request/Reply data copies the contents of the Request/Reply header from the source ETD to the target ETD (as shown in Figure 7 below).

Figure 7 Copying the Request/Reply header



5.2 Using the Java Servlet

The minimal implementation of the DataChannel e*Way via a DataChannel Server would work as follows:

1. A Java servlet for accepting requests from the Web server or Web browser.
2. The servlet sends the user request to the e*Way and posts the result to the Web server.

The Java Servlet is a set of Java classes which defines a standard interface between a Web client and a Web servlet. Client requests are made to the Web server, which then invokes the servlet to service the request through this interface. The Java servlet API is a standard Java Extension API.

The API is composed of two packages:

* javax.servlet

*javax.servlet.http

The 'Servlet' interface class defines the methods which servlets must implement, including a 'Service0' method for the handling of requests.

'HttpServlets' provide additional methods for the processing of HTTP requests such as GET (doGet method) and POST (doPost method).

5.3 Messaging Protocol Format

DCS and the DataChannel e*Way communicate information using a pre-defined messaging protocol. The DTD of the XML representing the protocol is given below:

```
<! ELEMENT DCSMESSAGE (CONTROLINFO, DATA, ORIGINALDATA)>
<! ELEMENT CONTROLINFO (RoutingTag, ReturnCount, ERROR)>
<! ELEMENT RoutingTag (# PCDATA)>
<! ELEMENT ReturnCount (# PCDATA)>
<! ELEMENT ERROR (ErrorCode, ErrorMessage)>
<! ELEMENT ErrorCode (# PCDATA)>
<! ELEMENT ErrorMessage (# PCDATA)>
<! ELEMENT DATA ANY>
<! ELEMENT ORIGINALDATA ANY>
```

5.3.1 e*Way Component Added to DISPATCH.XML

The following text is automatically added to the dispatch.xml file during the installation of the e*Way XpertLinX:

```
<HANDLER multi_dispatch_type=" SiebelClient" name=" SiebelClient"
class=" com.datachannel.merlin.dispatch.handlers.
multiDispatchHandlerBase">
<HANDLER name="egate" class="com.datachannel.merlin.dispatch.
handlers.getHandlers.externhandlers.SiebelClient" />
</ HANDLER>
```

5.3.2 e*Way Component Added to MODULES_INCLUDE.XSLT in theme

The following text is automatically added to the include.xslt file during installation of the e*Way XpertLinX:

```
<xsl:include xml:link="include" href="xpertlinx/datachannel/egate/
start.xslt"action="simple"/>
```

5.3.3 e*Way Component Added to RIOCONFIG.XML

The following text is automatically added to the rioconfig.xml file during installation of the e*Way XpertLinX:

```
<EXTERN name="egate"reload="30" virtualACLs="false">
  <XSL_INCLUDE config_display="hidden">xpertlinx/datachannel/egate/start.xslt</XSL_INCLUDE>
  <MODULE_FOLDER config_display="addonly"
    layout_full="xpertlinx/datachannel/egate/full"
    layout_prefs="xpertlinx/datachannel/egate/prefs"
    layout_properties="module_properties"
    layouturl="xpertlinx/datachannel/egate/settings"
    preview_ image="xpertlinx/datachannel/egate/images/preview_image.gif"/>
    <STYLESHEETS config_display="hidden">
    <SHEET local_name="xpertlinx/datachannel/egate/full.xslt"
      url="http://DCS_Hostname/xpertlinx/egate/full.xslt"/>
    <SHEET local_name="xpertlinx/datachannel/egate/prefs.xslt"
      url="http://DCS_Hostname/xpertlinx/egate/prefs.xslt"/>
    <SHEET local_name="xpertlinx/datachannel/egate/settings.xslt"
      url="http://DCS_Hostname/xpertlinx/egate/settings.xslt"/>
    <SHEET local_name="xpertlinx/datachannel/egate/prefs_feedback.xslt"
      url="http://DCS_Hostname/xpertlinx/egate/prefs_feedback.xslt"/>
    <SHEET local_name="xpertlinx/datachannel/egate/settings_feedback.xslt"
      url="http://DCS_Hostname/xpertlinx/egate/settings_feedback.xslt"/>
    <SHEET local_name="xpertlinx/datachannel/egate/start.xslt"
      url="http://DCS_Hostname/xpertlinx/egate/start.xslt"/>
```

```
    <SHEET local_name="xpertlinx/datachannel/egate/templates.xslt"
      url="http://DCS_Hostname/xpertlinx/egate/templates.xslt"/>
    <SHEET local_name="xpertlinx/datachannel/egate/feedback.xslt"
      url="http://DCS_Hostname/xpertlinx/egate/feedback.xslt"/>
    <IMAGE local_name="xpertlinx/datachannel/egate/images/preview_image.gif"
      url="http://DCS_ Hostname/xpertlinx/egate/images/preview_image.gif"/>
</ STYLESHEETS>
<DISPATCH_CONFIG config_display="hidden">
    <HANDLER class="com.datachannel.merlin.dispatch.handlers.getHandlers.externhandlers.SiebelClient"
method="SiebelClient"/>
</ DISPATCH_ CONFIG>
<DRIVER config_display="readonly">com.datachannel.rio.client.service.extern.impl.egate.eGate</DRIVER>
<URL config_display="hidden"> Not Used</URL>
<HOSTCONFIG>
    <HOST> stcserver.domain.com</HOST>
    <PORT> 26051</PORT>
    <EXPIRATIONTIME> 30</EXPIRATIONTIME>
    <RESPONSETIMEOUT> 30</RESPONSETIMEOUT>
</ HOSTCONFIG>
<SERVICECONFIG>
    <MUXPOOLSIZE> 2</MUXPOOLSIZE>
<ENCODE> false</ENCODE>
    <DEBUG> false</DEBUG>
    </SERVICECONFIG>
    </EXTERN>

Where,
<HOST> - egate server's name.
<PORT> - TCP/ IP port number (default 26051)
<EXPIRATIONTIME> - Number of seconds that the message may travel within e* Gate before it is marked as
expired (default 10)
<RESPONSETIMEOUT> - Number of seconds DCS will wait to receive a message from eGate (default 10)
<MUXPOOLSIZE> - Number of mux connections maintained by the system.
<ENCODE> - Set to true if the _createDocument has to use the character encoder; set to false otherwise.
<DEBUG> - Run the extern service in debug mode.
```

Chapter 6

Using the DataChannel Converter

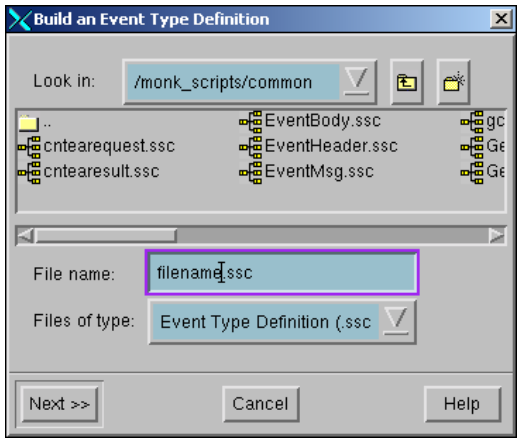
This chapter describes how to use the DataChannel Converter to convert an .xslt file to to an Event Type Definition (ETD) file that will be used to map XML messages that are sent via DCS for the specified XSLT file .

6.1 Creating Event Type Definitions from the GUI

To create an ETD using the DataChannel Converter from the GUI:

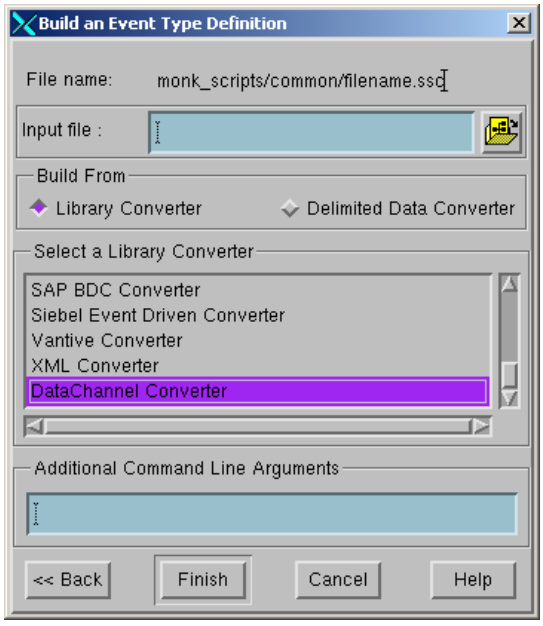
1. Launch the ETD Editor.
2. On the ETD Editor's Toolbar, click **Build**. The **Build an Event Type Definition** dialog box appears.

Figure 8 Build an Event Type Definition



3. In the **File name** field, type the name of the ETD file you wish to build. Do not specify any file extension—the Editor will supply the .ssc extension automatically.
4. Click **Next**. A new dialog box displays.

Figure 9 Build an Event Type Definition - DataChannel Converter



5 Leave the **Input file** field blank—the HTML Converter does not use this field.

6 Under **Build From**, select **Library Converter**.

7 Under **Select a Library Converter**, select **DataChannel Converter**.

8 Under **Additional Command Line Arguments**, type the following:

```
-url xsltURL
```

where

xsltURL is the URL of the XSLT File to build the .ssc file

9 Click **Finish**. The Build tool will create the ETD.

The ETD Editor will open the resulting ETD file automatically at the conclusion of the conversion process. If multiple ETD files were created, you must open each file manually.

6.2 Creating Event Type Definitions from a Command Line

To create an ETD using the DataChannel Converter from a command line:

From the command line type the following on one line:

```
java com.stc.ewdatachannel.Converter -url xsltURL outputSSCFile
```

where

- *xsltURL* is the URL of the XSLT File to build the .ssc file

- *outputSSCFile* is the filename—including the path relative to the "eGate/client" directory—of the Event Type Definition file to be created. Do not specify any file extension— the converter will supply the .ssc extension automatically.

Note: *The **output_file** argument must be the last argument listed.*

Index