SeeBeyond ICAN Suite

e*Way Intelligent Adapter for SAP (EDI) User's Guide

Release 5.0.5 for Schema Run-time Environment (SRE)

Version 20050406035005.

Contents

Preface

This Preface contains information regarding the User's Guide itself.

P.1 Intended Reader

The reader of this guide is presumed to be a developer or system administrator with responsibility for maintaining the SeeBeyond™ e*Gate™ Integrator system, and have a working knowledge of:

- Windows and/or UNIX operations and administration
- Windows-style GUI operations
- SAP EDI and IDoc format
- Integrating SAP R/3 with external systems

P.2 Organization

This User's Guide is organized roughly into two parts. The first part, consisting of Chapters 1-5, introduces the e*Way and describes the procedures for installing the e*Way and implementing a working system incorporating the e*Way. Chapter 3 also contains descriptions of the sample schemas provided with the product. These can be used to test your system following installation and, if appropriate, as templates you can modify to produce your own custom schemas. This part should be of particular interest to a System Administrator or other user charged with the task of getting the system up and running.

The second part, consisting of Chapters 6-7 and the Appendix, describes the architecture and internal functionality of the e*Way. This part should be of particular interest to a Developer involved in customizing the e*Way for a specific purpose. Information contained in this part that is necessary for the initial setup of the e*Way is cross-referenced in the first part of the guide, at the appropriate points in the procedures. The Appendix contains information on SeeBeyond™ Utility routines that are referred to elsewhere in the User's Guide.

P.3 Nomenclature

Note that for purposes of brevity, the e*Way Intelligent Adapter for SAP (EDI) is frequently referred to as the SAP EDI e*Way, or simply the e*Way.

P.4 Online Use

This User's Guide is provided in Adobe Acrobat's Portable Document Format (PDF). As such, it can be printed out on any printer or viewed online. When viewing online, you can take advantage of the extensive hyperlinking imbedded in the document to navigate quickly throughout the Guide.

Hyperlinking is available in:

- The Table of Contents
- The Index
- Within the chapter text, indicated by **blue print**

Existence of a hyperlink *hotspot* is indicated when the hand cursor points to the text. Note that the hotspots in the Index are the *page numbers*, not the topics themselves. Returning to the spot you hyperlinked from is accomplished by right-clicking the mouse and selecting **Go To Previous View** on the resulting menu.

P.5 Writing Conventions

The writing conventions listed in this section are observed throughout this document.

Monospaced (Courier) Font

Computer code and text to be typed at the command line are set in Courier as shown below:

```
Configuration for BOB_Promotion

java -jar ValidationBuilder.jar
```

Variables within a command line are set in the same font within brackets <> as shown below:

```
stcregutil -rh <host-name> -un <user-name> -up <password> -sf
```

Bold Sans-serif Font

- User Input: Click **Apply** to save, or **OK** to save and close.
- File Names and Paths: In the **Open** field, type **D:\setup\setup.exe**.
- Parameter, Function, and Command Names: The default parameter **localhost** is normally only used for testing; the Monk function **iq-put** places an Event into an IQ.

Chapter 1

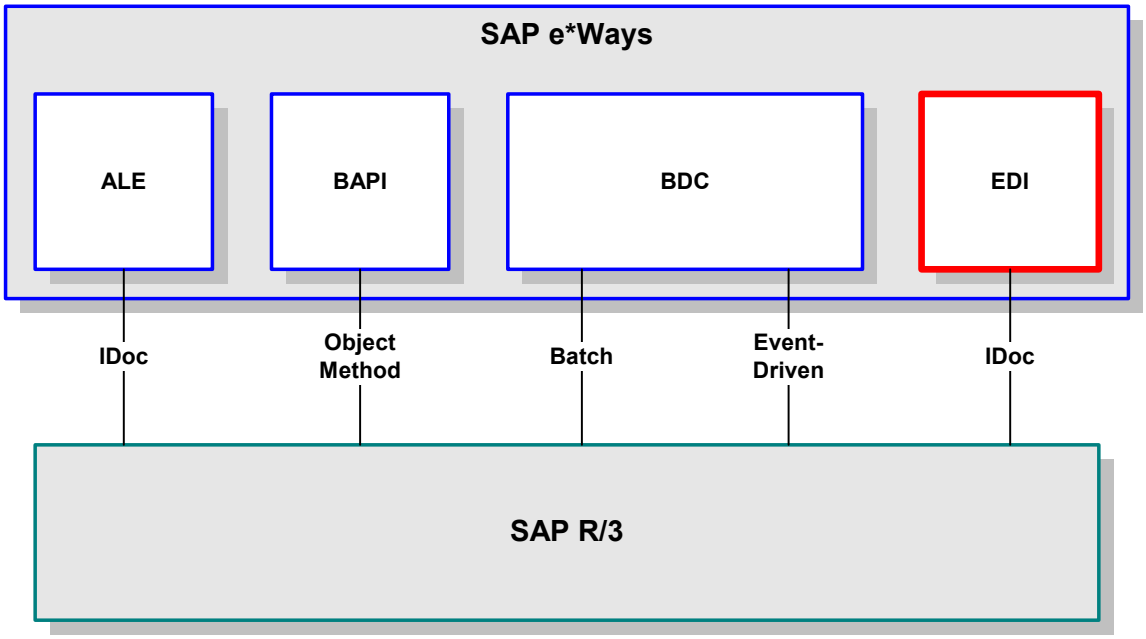
Introduction

The e*Way Intelligent Adapter for SAP (EDI) allows bidirectional data exchange between e*Gate and an SAP R/3 system via EDI (Electronic Data Interchange). It is SAP-Certified for SAP R/3 Release 3.x and 4.x. The SAP EDI e*Way controls the communication protocol layer between the SAP R/3 host and e*Gate, and can be configured to process data in either direction.

1.1 SAP Interface Options

SAP offers several interface options, including Application Link Enabling (ALE), Business Application Programming Interface (BAPI), and Electronic Data Interchange (EDI). The Batch Data Communication (BDC) interface actually is a user-emulation method that can be either batch or event-driven.

Figure 1 SAP Interface Options



All of these interfacing methods are supported by SeeBeyond, for both SAP-inbound and SAP-outbound data transfer.

1.1.1 The EDI Interface

The SAP EDI e*Way is intended for those sites that prefer to exchange IDocs with the SAP host via batch files, rather than in real time. This would be required in the absence of a constant connection using a protocol such as TCP/IP, for example.

Transactions are exchanged using the SAP Intermediate Document (IDoc) format, which is basically a fixed message. The IDoc file is interpreted by correlating with an IDoc Description file, which specifies all relevant fields.

1.2 SAP Intermediate Document Format

Intermediate Documents are used as containers for information, and are used to upload data to and download data from other systems. IDocs allow independence between the format and content of the message.

1.2.1 Nomenclature

Several hundred IDocs are supplied with each R/3 system, serving as templates for a wide variety of applications. The IDoc hierarchy is represented by the following terminology:

- **Message Types** are related to specific applications such as **Orders**.
- **IDoc Types** are different versions of standard Message Types, such as orders for specific items or services.

1.2.2 Structure

A typical SAP IDoc consists of Control, Data, and Status records, as shown in **Figure 2 on page 11**. Status records are used only by the EDI interface.

Figure 2 SAP IDoc Structure

Control Record

Sender	Recvr	Messg. Type	IDoc-Type	Status

Data Record/Segments

HEADER1	XXXXXXXXXXXX
ITEMS	XXXXXXXXXXXXXXXXXXXXXXXXX
SUBITEMA	XXXXXX
SUBITEMB	XXXXXX
TEXT	XXXXXXXXXXXX
ITEMS	XXXXXXXXXXXXXXXXXXXXXXXXXXX
SUBITEMC	XXXXXXX
SUBITEMD	XXXXXX
ACCUM	XXXXXXXXXXXXXXX

Status Record

‘To be processed’	14:33:48

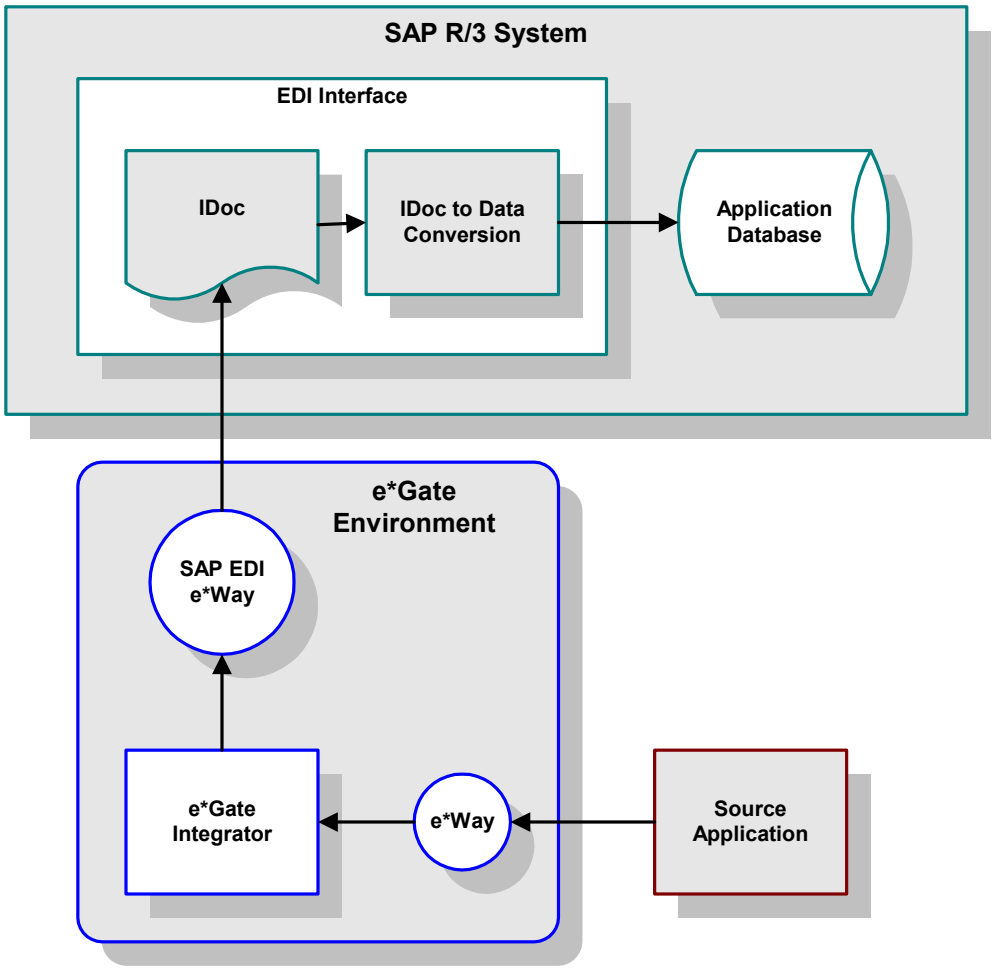
‘Processing successful’	14:33:57

1.3 The SAP EDI e*Way

1.3.1 Overview

e*Gate to SAP

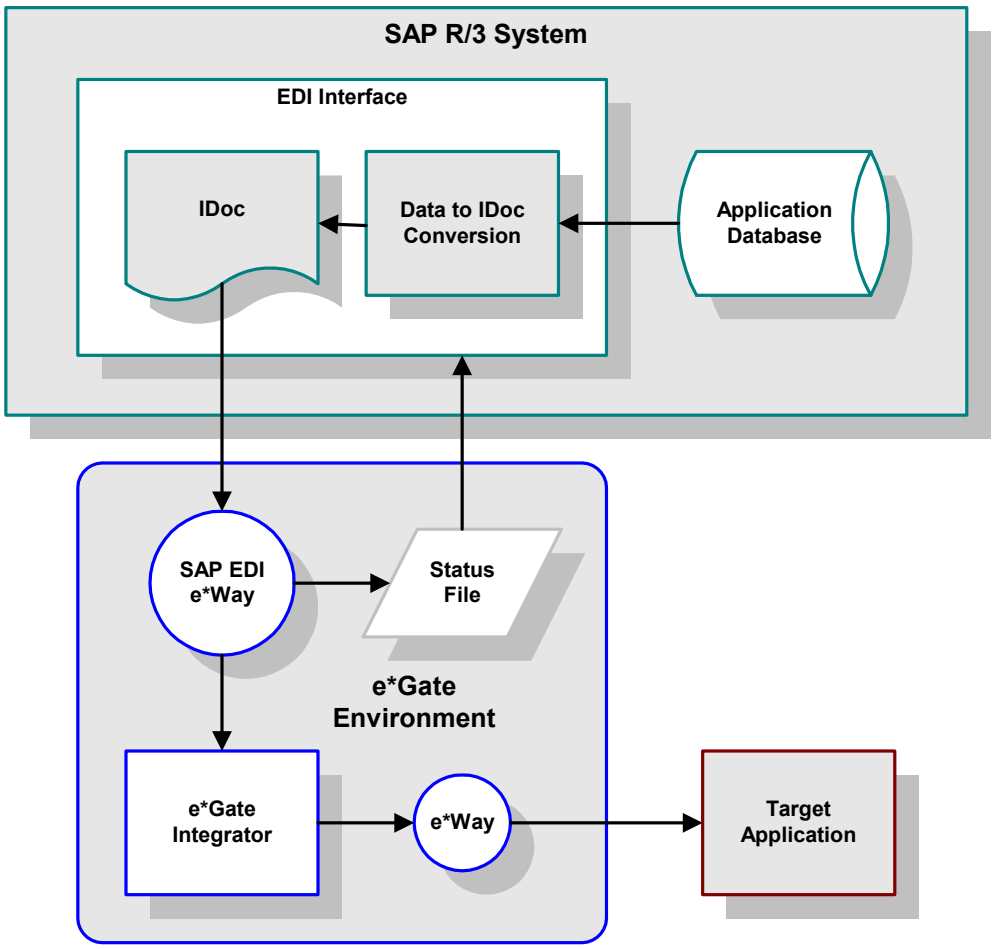
Figure 3 EDI SAP-inbound Interface Data Flow



In this mode of operation, the SAP EDI e*Way converts data into an IDoc, which is passed to the SAP EDI Interface. The EDI Interface, in turn, transforms the IDoc into SAP-format data which is passed to the appropriate SAP application.

SAP to e*Gate

Figure 4 EDI SAP-Outbound Interface Data Flow



In this mode of operation, the SAP EDI Interface converts data into an IDoc, which is stored as a file in its *Outbound Directory*. Two data retrieval methods are available for obtaining an IDoc from the SAP system. Details of these data retrieval methods are described in **Chapter 6**.

1.3.2 Components

The SAP EDI e*Way incorporates the following components:

- An executable file, **stcewsapedi.exe**
- An accompanying dynamic load library, **librfc32.dll** (**librfccm.sl** for UNIX), containing RFC extensions
- Default configuration files, **stcewsapediin.def** and **stcewsapediout.def**
- SeeBeyond utilities **stcftp_get.exe**, **stcftp_put.exe**, and **stcjdump.exe**
- Monk and Java Wizards used to build Event Type Definitions

- Example schema, described in **Sample Schema** on page 44

For a list of installed files, see **Installing the e*Way** on page 17.

1.4 Supported Operating Systems

The e*Way Intelligent Adapter for SAP EDI currently supports the following combinations of operating systems and releases of SAP R/3.

Table 1 English-language Version

Operating System	4.0B	4.5B	4.6B	4.6C
Windows 2000 and Windows Server 2003	X	X	X	X
HP-UX 11.0 and 11i (PA-RISC)	X	X	X	X
IBM AIX 5.1L	X	X	X	X
Sun Solaris 8	X	X	X	X

Table 2 Japanese-language Version

Operating System	4.0B	4.5B	4.6B	4.6C
Windows 2000 and Windows Server 2003	-	-	-	X
HP-UX 11.0 and 11i (PA-RISC)	-	-	-	X
Sun Solaris 8	-	-	-	X

Table 3 Korean-language Version

Operating System	4.0B	4.5B	4.6B	4.6C
HP-UX 11.0 and 11i (PA-RISC)	-	-	-	X
IBM AIX 5.1L	-	-	-	X

Chapter 2

Installation

This chapter describes the requirements and procedures for installing the e*Way software. Procedures for implementing a working system, incorporating instances of the e*Way, are described in **Chapter 3**.

Note: *Please read the readme.txt file located in the addons\ewsapedi directory on the installation CD-ROM for important information regarding this installation.*

2.1 System Requirements

To use the e*Way Intelligent Adapter for SAP EDI, you need the following:

1. An e*Gate Participating Host.
2. A TCP/IP network connection.
3. Sufficient free disk space to accommodate e*Way files:
   - Approximately 3 MB on Windows systems
   - Approximately 9.3 MB on Solaris systems
   - Approximately 8.4 MB on HP-UX systems
   - Approximately 10.8 MB on AIX systems

Note: *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.*

2.1.1 Environment Configuration

No changes are required to the Participating Host's operating environment to support this e*Way.

2.2 External System Requirements

The e*Way Intelligent Adapter for SAP EDI supports the following applications (see also **Supported Operating Systems** on page 14).

English

- SAP R/3 release 4.0B, 4.5B, 4.6B, and 4.6C

2.2.1 External Configuration Requirements

Your SAP R/3 system must be configured to communicate with the SAP EDI e*Way (see **SAP R/3 Configuration** on page 50).

2.3 Installing the e*Way

2.3.1 Windows Systems

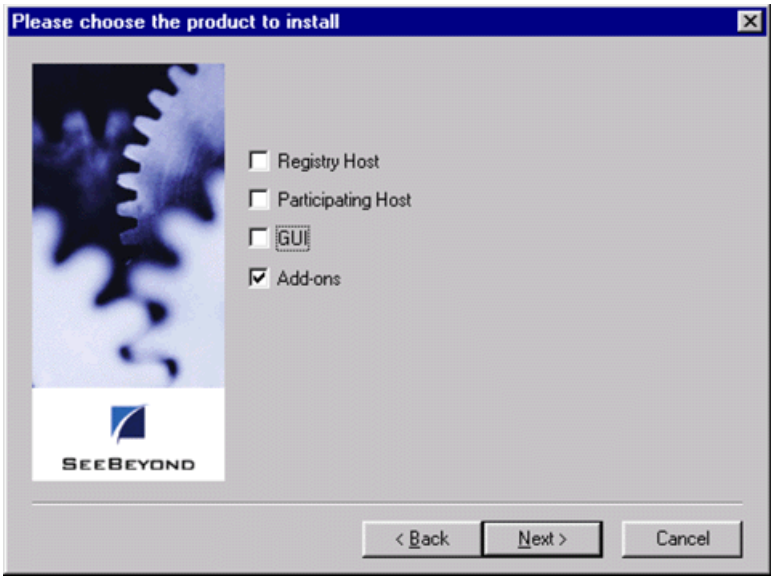
Installation Procedure

Note: *The installation utility detects and suggests the appropriate installation directory. Use this directory unless advised otherwise by SeeBeyond.*

To install the e*Way on a Microsoft Windows system

1. Log in as an Administrator on the workstation on which you want to install the e*Way (*you must have Administrator privileges to install this e*Way).*
2. Exit all Windows programs and disable any anti-virus applications before running the setup program.
3. Insert the e*Way installation CD-ROM into the CD-ROM drive.
4. Launch the setup program.
   - A If the CD-ROM drive's Autorun feature is enabled, the setup program should launch automatically. Follow the on-screen instructions until the **Choose Product** dialog box appears (see Figure 5). Check **Add-ons**, then click **Next**.

Figure 5 Choose Product Dialog Box

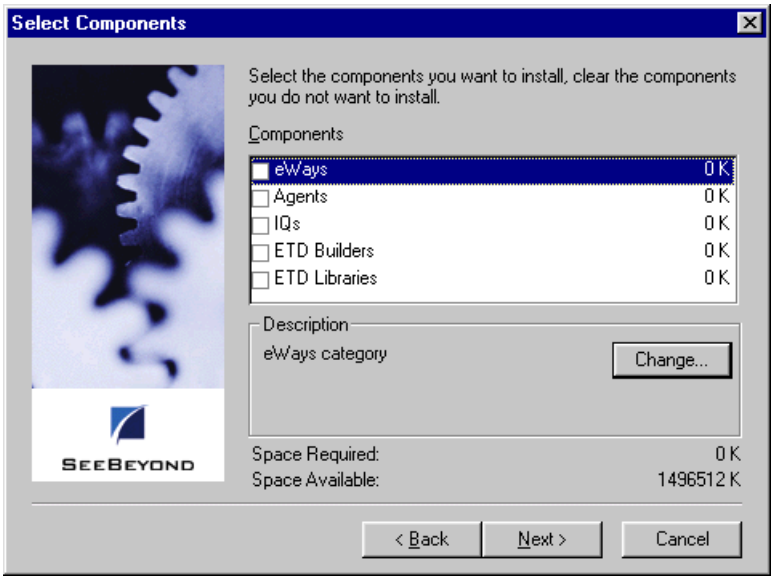


   - B If the setup program does not launch automatically, use the Windows Explorer or the Control Panel's **Add/Remove Applications** feature to launch the following file on the CD-ROM drive (this bypasses the **Choose Product** dialog):

```
setup\addons\setup.exe
```

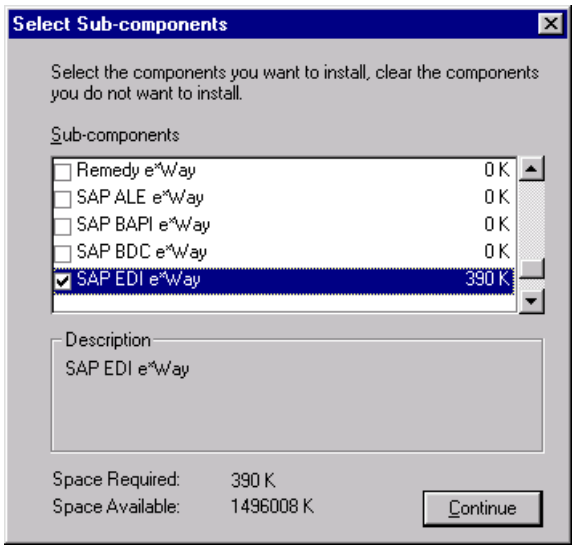
5 Follow the on-screen instructions until the **Select Components** dialog box appears (see Figure 6). Highlight—*but do not check*—**eWays** and then click **Change**.

Figure 6 Select Components Dialog Box



6 When the **Select Sub-components** dialog box appears (see Figure 7), check the **SAP EDI e*Way**.

Figure 7 Select e*Way Dialog



7 Click **Continue**, and the **Select Components** dialog box reappears.

8 Click **Next** and continue with the installation.

Subdirectories and Files

By default, the InstallShield installer creates the following subdirectories and installs the following files within the **\eGate\client** tree on the Participating Host, and the **\eGate\Server\registry\repository\default** tree on the Registry Host.

Table 4 Participating Host & Registry Host

Subdirectories	Files
\	stcewsapedi.ctl_retrieve
\bin\	get_idocs_from_r3.bat get_idocs_from_r3.sh librfc32.dll rfcexec.exe send_idocs_to_r3.bat send_idocs_to_r3.sh send_status_records_to_r3.bat send_status_records_to_r3.sh startrfc.exe stcewsapedi.ctl stcewsapedi.exe stcftp_get.exe stcftp_put.exe stcjdump.exe stcpxcmd.exe stcsapconvert.exe
\configs\stcewsapedi\	sapEdiIn3.6To4.1Rule.txt sapEdiOut3.6To4.1Rule.txt stcewsapediin.def stecwsapediout.def
\stcgui\ctls\	guisapedi.ctl

By default, the InstallShield installer also installs the following file within the **\eGate\Server\registry\repository\default** tree on the Registry Host.

Table 5 Registry Host Only

Subdirectories	Files
\	sapedi.ctl

2.3.2 UNIX Systems

Installation Procedure

Note: *You are not required to have root privileges to install this e*Way. Log on 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.*

To install the e*Way on a UNIX system

1. Log onto the workstation containing the CD-ROM drive and, if necessary, mount the drive.
2. Insert the CD-ROM into the drive.
3. At the shell prompt, type

   cd /cdrom
4. Start the installation script by typing:

   setup.sh
5. A menu appears, with several options. Select the **install e*Way** option, and follow any additional on-screen instructions.

Note: *The installation utility detects and suggests the appropriate installation directory. Use this directory unless advised otherwise by SeeBeyond. Note also that* ***no spaces*** *should appear in the installation path name.*

Subdirectories and Files

The preceding installation procedure creates the following subdirectories and installs the following files within the **/eGate/client** tree on the Participating Host, and the **/eGate/Server/registry/repository/default** tree on the Registry Host.

Table 6 Participating Host & Registry Host

Subdirectories	Files
/	stcewsapedi.ctl_retrieve
/bin/	lbrfccm.sl (HP-UX only) get_idocs_from_r3.bat get_idocs_from_r3.sh rfcexec send_idocs_to_r3.bat send_idocs_to_r3.sh send_status_records_to_r3.bat send_status_records_to_r3.sh startrfc stcewsapedi.ctl stcewsapedi stcftp_get stcftp_put stcjdump stcpxcmd
/configs/stcewsapedi/	sapEdiIn3.6To4.1Rule.txt sapEdiOut3.6To4.1Rule.txt stcewsapediin.def stcewsapediout.def
/stcgui/ctls/	guisapedi.ctl

The preceding installation procedure also installs the following file only within the **/eGate/Server/registry/repository/default** tree on the Registry Host.

Table 7 Registry Host Only

Subdirectories	Files
\	sapedi.ctl

2.4 Optional Example Files

The installation CD-ROM contains a sample schema, **SapEdi**, located in the **samples\ewsapedi** directory. To use a schema, you must load it onto your system using the following procedure. See **Sample Schema** on page 44 for descriptions of the sample schema and instructions regarding its use.

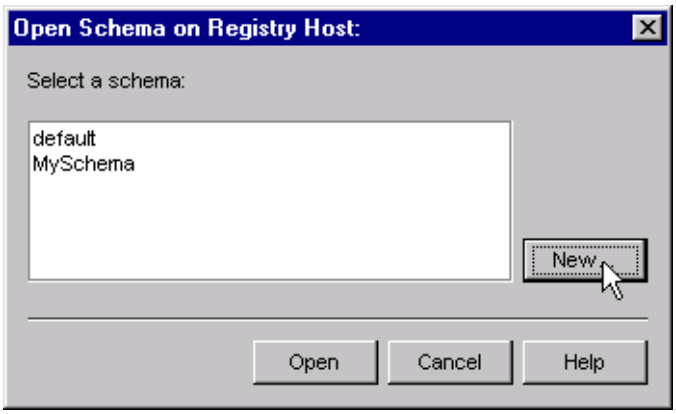
Note: *The SAP EDI e*Way must be properly installed on your system before you can run the sample schema.*

2.4.1 Installation Procedure

To load a sample schema

1 Invoke the **Open Schema** dialog box and select **New** (see Figure 8).

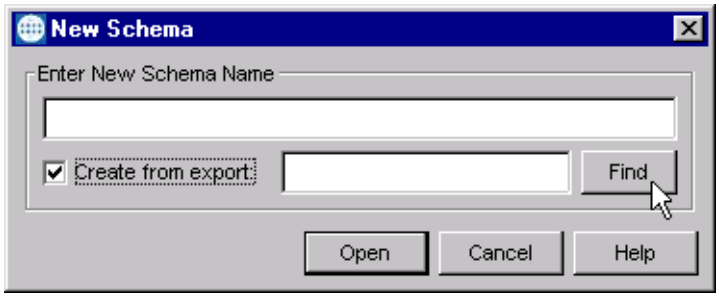
Figure 8 Open Schema Dialog



2 Type the name you want to give to the schema (for example, **SapEdi.Sample**)

3 Select **Create from export** and navigate to the directory containing the sample schema by clicking the **Find** button (see Figure 9).

Figure 9 New Schema Dialog



4 Select **SapEdi.zip** and click **Open**.

Note: *The schema installs with the host name* **localhost** *and control broker name* **localhost_cb**. *If you want to assign your own names, copy the file* **SapEdi.zip** *to a local directory and extract the files. Using a text editor, edit the file* **SapEdi.exp**, *replacing all instances of the name* **localhost** *with your desired name. Add the edited* **.exp** *file back into the* **.zip** *file.*

2.4.2 Subdirectories and Files

The preceding procedure creates the following subdirectories and installs the following files within the **\eGate\Server\registry\repository\<SchemaName>** tree on the Registry Host, where **<SchemaName>** is the name you have assigned to the schema in step 2.

Table 8 Subdirectories and Files - SapEdi

Subdirectories	Files
\	SapEdi.ctl
\runtime\bin	get_idocs_from_r3.bat get_idocs_from_r3.sh send_idocs_to_r3.bat send_idocs_to_r3.sh
\runtime\configs\stcewsapedi\	EDI_TO_SAP_OUT.cfg EDI_TO_SAP_OUT.sc SAP_TO_EDI_IN.cfg SAP_TO_EDI_IN.sc
\runtime\data\edi_src\	matmas02.idocs
\runtime\monk_scripts\common\	root.ssc
\runtime\monk_scripts\templates\Sap_r3_4.0b\	IT4_MATMAS02.ssc

Chapter 3

System Implementation

In this chapter we take a more detailed look at the information presented in the Introduction, and describe the steps required for setting up a working system. Please refer to the *e*Gate Integrator User's Guide* for additional information.

3.1 Overview

This e*Way provides a specialized transport component for incorporation in an operational schema. The schema also contains Collaborations, linking different data or Event types, and Intelligent Queues. Typically, other e*Way types also are used as components of the schema.

One or more sample schema, included in the software package, are described at the end of this chapter. These can be used to test your system following installation and, if appropriate, as a template that you can modify to produce your own schema.

3.1.1 Pre-Implementation Tasks

- Installation of SeeBeyond software

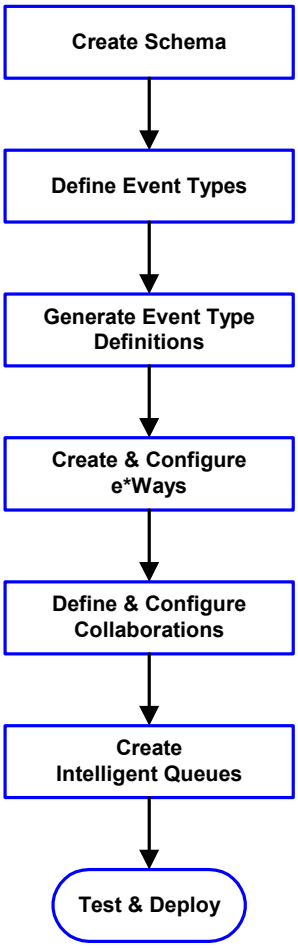
  The first task, of course, is to install the SeeBeyond software as described in **Chapter 2**.

- Modification of SAP R/3 environment

  After you have installed the SeeBeyond software, your existing SAP R/3 system must be configured to interact properly with the SAP ALE e*Way. See **Chapter 4**.

Topics covered in this chapter are:

3.1.2 Implementation Sequence



1 The first step is to create a new Schema—the subsequent steps apply only to this Schema (see **Creating a Schema** on page 26).

2 The second step is to define the Event Types you are transporting and processing within the Schema (see **Creating Event Types** on page 27).

3 Third, you need to associate the Event Types created in the previous step with Event Type Definitions (ETDs) derived from the applicable Business Rules (see **Creating Event Type Definitions** on page 28).

4 The fourth step is to create and configure the required e*Ways (see **Chapter 5**).

5 Next is to define and configure the Collaborations linking the Event Types from step 2 (see **Defining Collaborations** on page 43).

6 Now you need to create Intelligent Queues to hold published Events (see **Creating Intelligent Queues** on page 44

7 Finally, you must test your Schema. Once you have verified that it is working correctly, you may deploy it to your production environment.

3.1.3 Viewing e*Gate Components

Use the Navigator and Editor panes of the e*Gate Schema Designer to view the various e*Gate components. Note that you may only view components of a single schema at one time, and that all operations apply only to the current schema. All procedures in this chapter should be performed while displaying the **Components** Navigator pane. See the *e*Gate Integrator User's Guide* for a detailed description of the features and use of the Schema Designer.

3.2 Creating a Schema

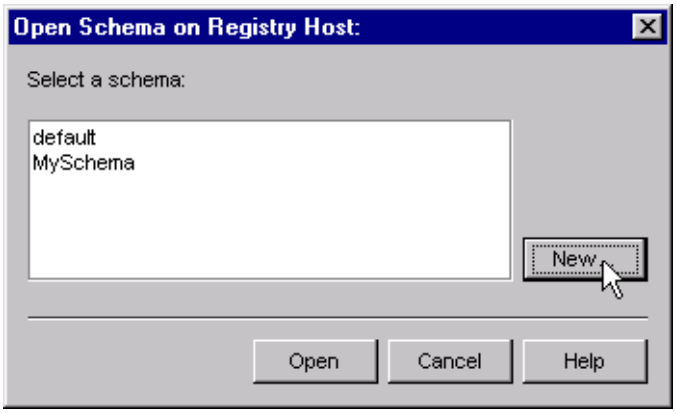
A schema is the structure that defines e*Gate system parameters and the relationships between components within the e*Gate system. Schemas can span multiple hosts.

Because all setup and configuration operations take place within an e*Gate schema, a new schema must be created, or an existing one must be started before using the system. Schemas store all their configuration parameters in the e*Gate Registry.

To select or create a schema

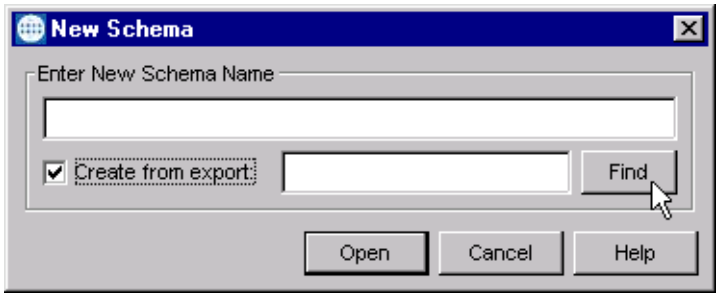
1 Invoke the **Open Schema** dialog box (Figure 10) and **Open** an existing schema, or click **New** to create a new schema.

Figure 10 Open Schema Dialog



2 Clicking **New** invokes the **New Schema** dialog box (Figure 11).

Figure 11 New Schema Dialog



3 Enter a new schema name and click **Open**.

4 The e*Gate Schema Designer then opens under your new schema name.

5 From the **Options** menu, click on **Default Editor** and select **Monk**.

6 Select the **Components** tab, found at the bottom of the Navigator pane of the e*Gate Schema Designer window.

7 You are now ready to begin creating the necessary components for this new schema.

3.3 Creating Event Types

Within e*Gate, messages and/or packages of data are defined as Events. Each Event must be categorized into a specific Event Type within the schema.

To define the Event Types

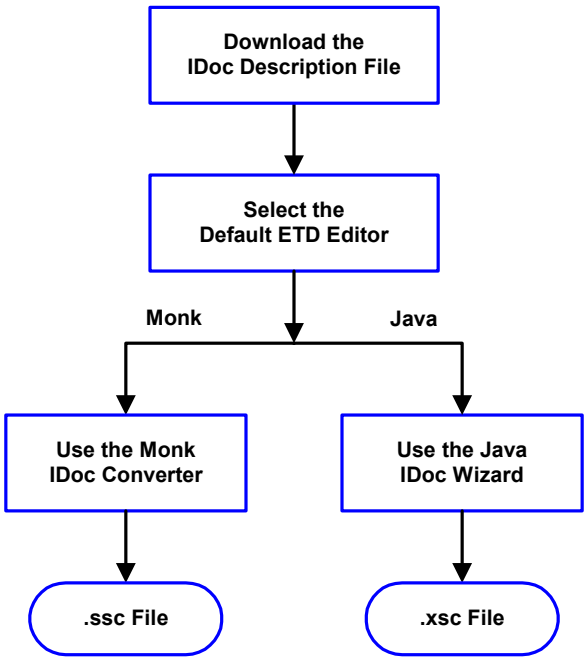
1. In the e*Gate Schema Designer's Navigator pane, select the **Event Types** folder.
2. On the Palette, click the **New Event Type** button  .
3. In the **New Event Type Component** box, enter the name for the input Event Type and click **Apply**. Use this method to create all required Event Types, for example:
   - **InboundEvent**
   - **ValidEvent**
   - **InvalidEvent**
4. After you have created the final Event Type, click **OK**.

3.4 Creating Event Type Definitions

Before e*Gate can process any data to or from an SAP system, you must create an Event Type Definition to package and route that data within the e*Gate system. The ETD Editor automatically creates an Event Type Definition file based upon structural metadata, using a Wizard-driven build tool.

With e*Gate two options are available: a Monk IDoc Converter, and a Java IDoc Wizard. You choose the method by selecting the default ETD editor (either **Monk** or **Java**) from the Schema Designer's **Options** menu.

Figure 12 ETD Creation Process



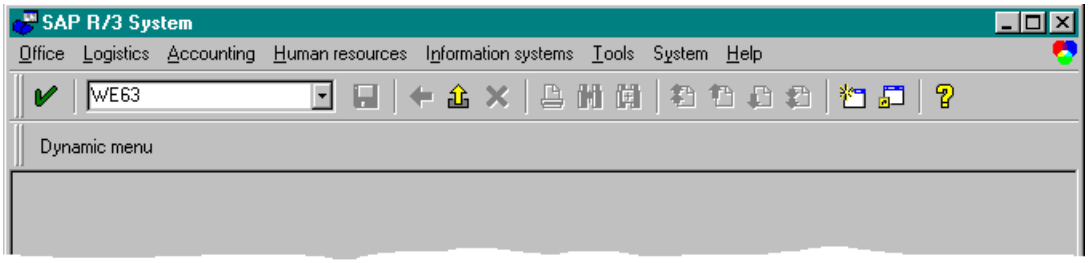
See the *e*Gate Integrator User's Guide* for additional information about Event Type Definitions and the e*Gate ETD Editor.

3.4.1 Downloading the IDoc Description File

You must download the IDoc Description file to the host where theSAP EDI e*Way is being run from within the SAP application. To download the IDoc Description File from SAP to the e*Way:

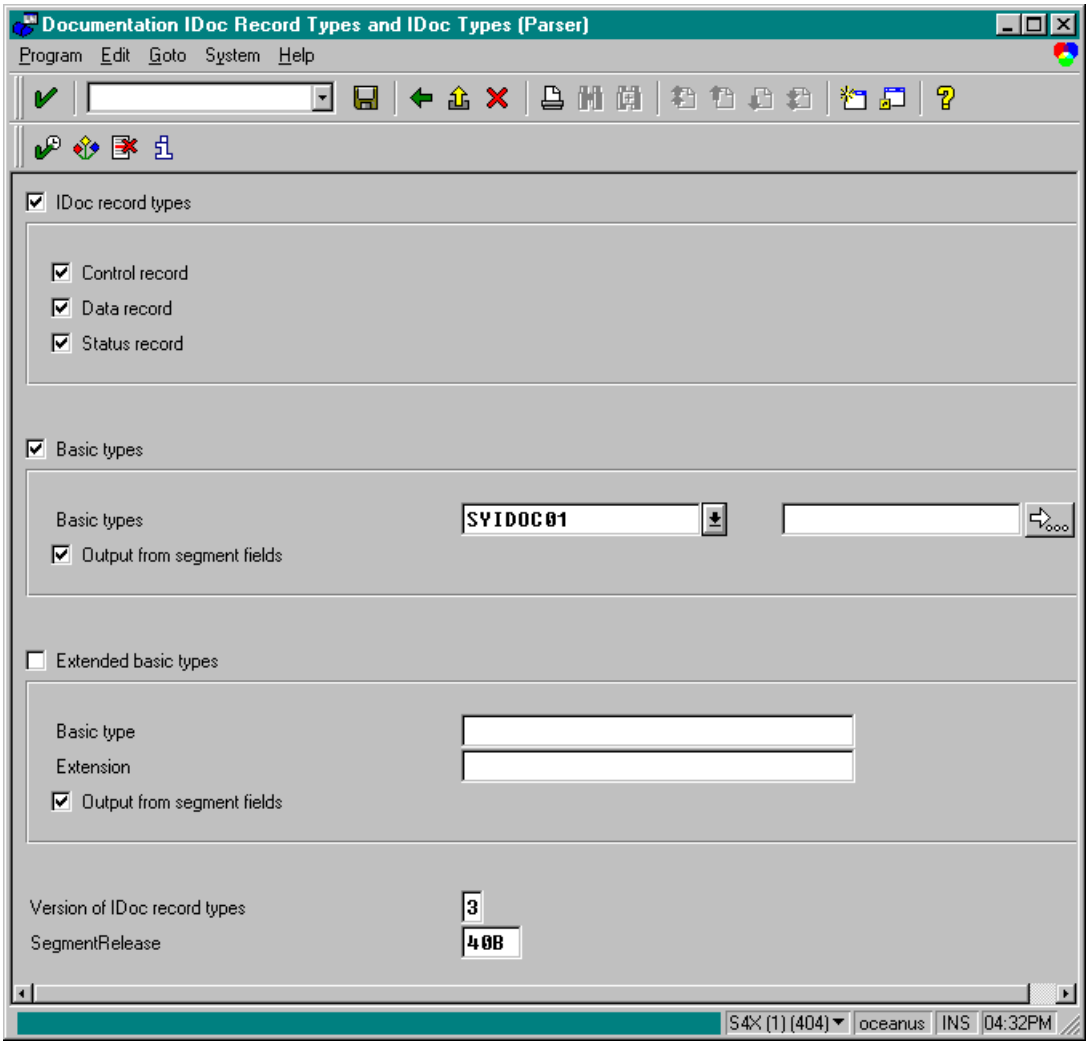
Transaction: WE63

Figure 13 SAP R/3 System Window



1 Type **WE63** into the command field of the *SAP R/3 System* window and select  , which displays the *Documentation IDoc Record Types and IDoc Types (Parser)* window (alternatively, you can execute RSE IDOC3 in **SE38**)

Figure 14 Documentation IDoc Record Types and IDoc Types (Parser) Window (1)

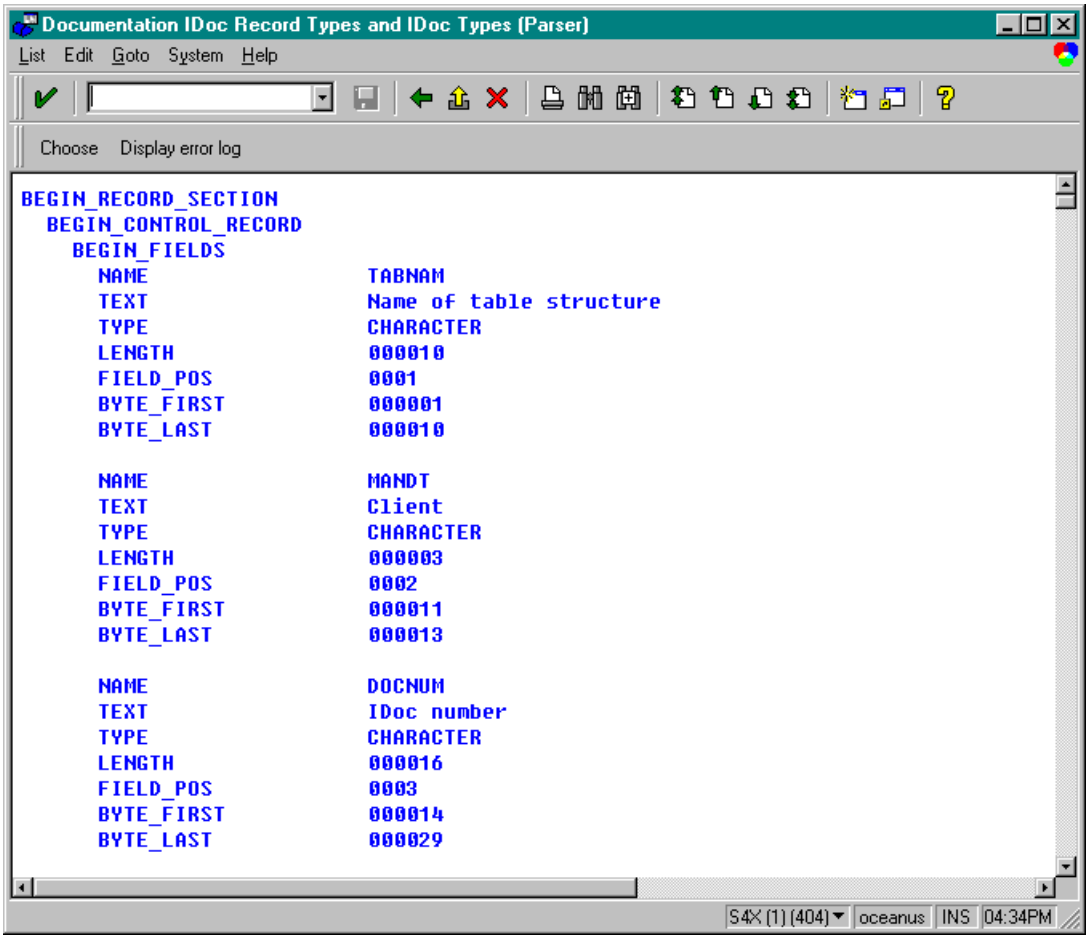


2 Type the **IDoc name** for which you wish to generate a Description File into the command field, or locate by invoking the drop-down menu for existing **IDoc types**.

3 Make sure the *Control record*, *Data record*, and *Status record* boxes are all checked.

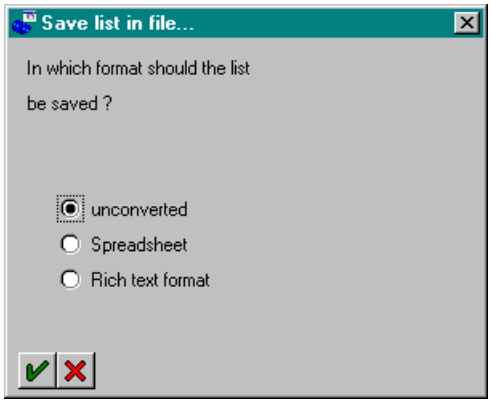
4 Select  to begin parsing the IDoc message.

Figure 15 Documentation IDoc Record Types and IDoc Types (Parser) Window (2)



5 After the Description File appears, select **List** > **Download** from the **Menu** bar, which displays the *Save list in file...* dialog box.

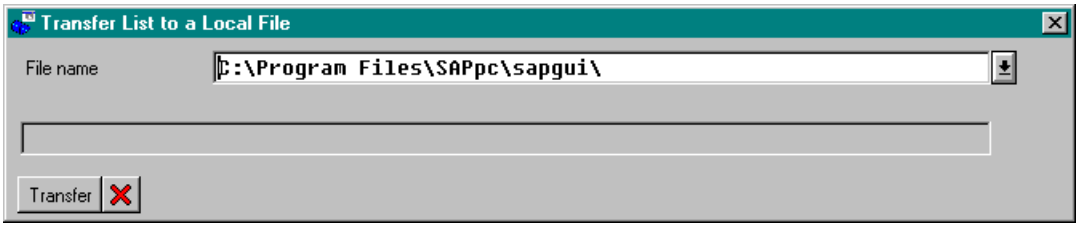
Figure 16 Save List in File... Dialog Box



6 Select the **unconverted** option, since no format conversion is required.

7 Select  , which displays the *Transfer List to a Local File* window.

Figure 17 Transfer List to a Local File Window



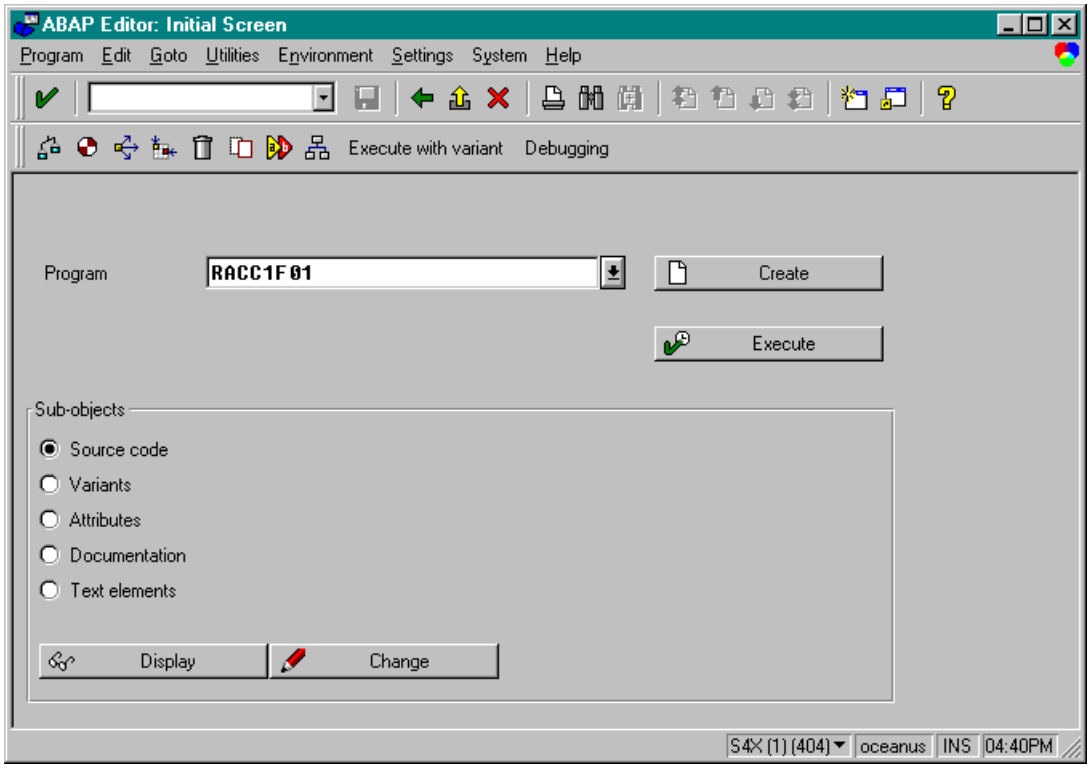
8 Enter the name and path of the local file to receive the IDoc Description File.

9 Selecting **Transfer** downloads the file and returns you to the *SAP R/3 System* window.

Transaction: SE38

1 Type **SE38** into the command field of the *SAP R/3 System* window and **Enter**, which displays the *ABAP Editor Initial Screen* window.

Figure 18 ABAP Editor Initial Screen Window



2 Enter the name of the Program to run into the text box or select from the drop-down menu.

3 Execute the program by selecting Execute.

4 **Enter** and select **Back** repeatedly to return to the *SAP R/3 System* window.

3.4.2 Using the ETD Editor's Build Tool

The Event Type Definition Editor's Build tool automatically creates an Event Type Definition file representing an IDoc defined in SAP, using the IDoc Description File downloaded from SAP.

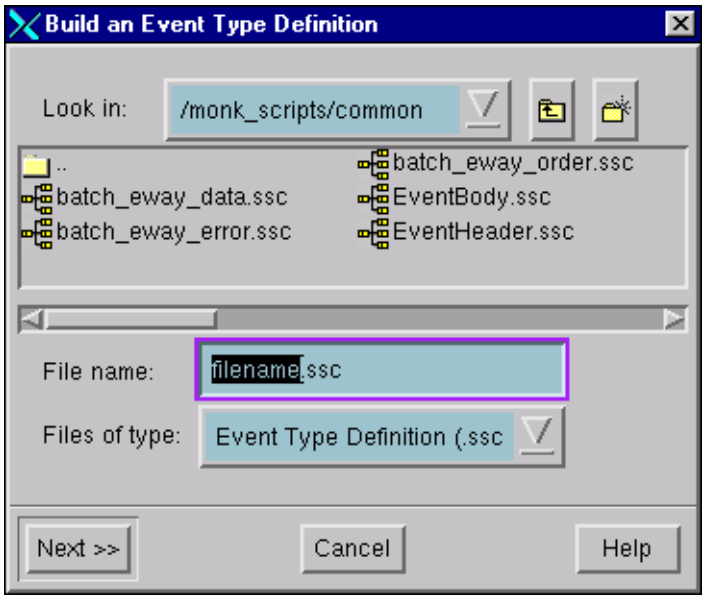
Note: *Be sure to set the Default Editor to* **Monk**, *from the* **Options** *menu in the e*Gate Schema Designer.*

To create an Event Type Definition using the Build tool

1. Launch the ETD Editor by clicking  in the e*Gate Schema Designer tool bar.
2. On the ETD Editor's tool bar, click **Build**.

   The *Build an Event Type Definition* dialog box opens.

Figure 19 Build Event Type Definition Dialog

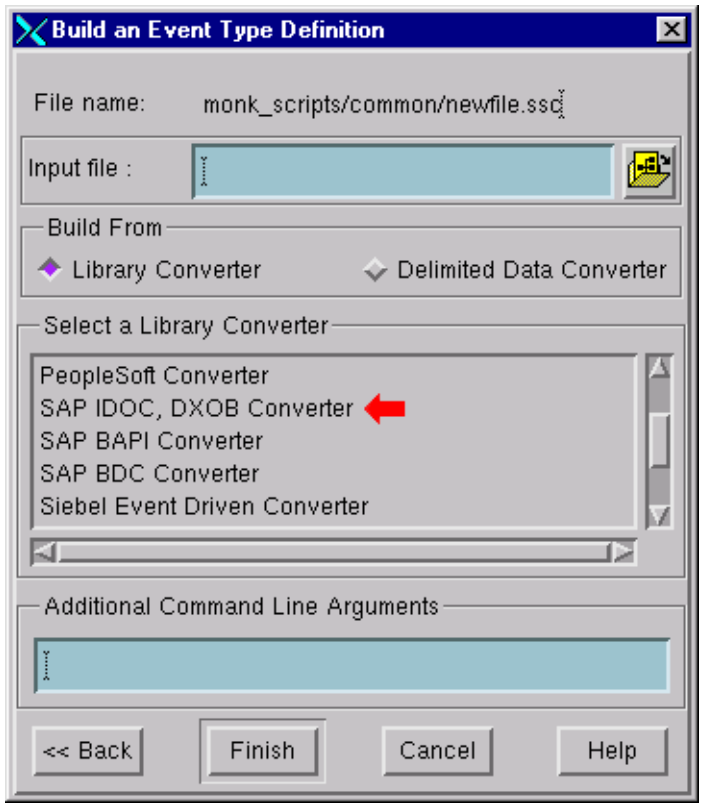


3. In the *File name* box, type the name of the ETD file you want to build.

Note: *The Editor automatically supplies the* **.ssc** *extension.*

4. Click **Next**. A new dialog box appears, as shown in Figure 20.

Figure 20 Building the ETD



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

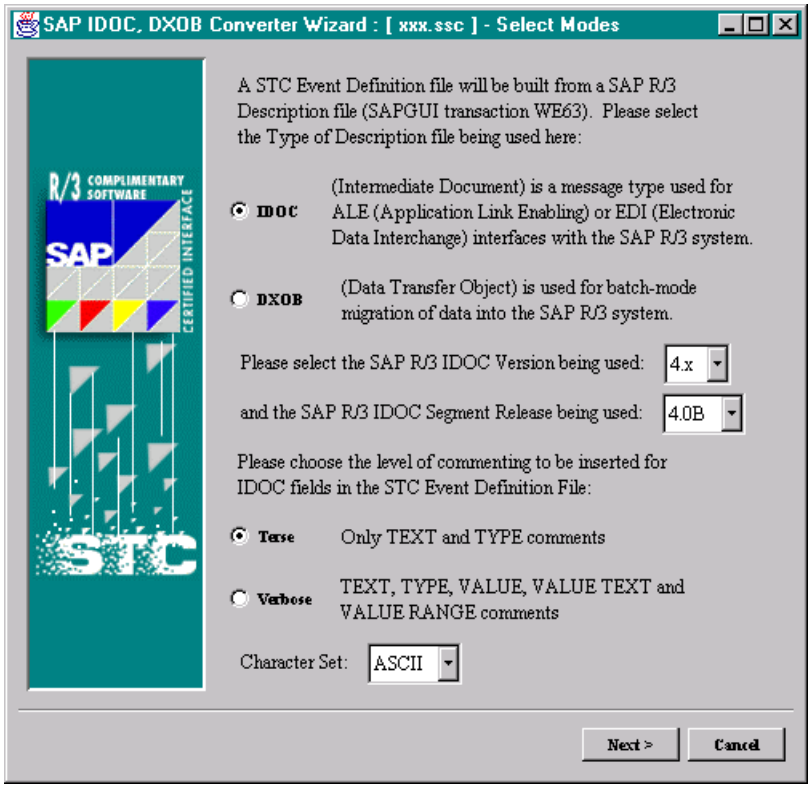
6 Under *Select a Library Converter*, select **IDOC, DXOB Converter**.

7 In the *Additional Command Line Arguments* box, type any additional arguments, if desired.

8 Click **Finish**, and the IDOC, DXOB Converter Wizard appears.

9 Follow the Wizard's instructions to finish building the ETD file.

Figure 21 SAP Converter Wizard (1)



10 Select the **IDoc** option.

11 Select or enter the IDoc Version and the SAP R/3 IDoc Release.

Note: *An **IDoc Release** refers to the corresponding SAP R/3 release, whereas an **IDoc Version** specifies whether it's the older, shorter format or the newer, longer format (see the following table).*

SAP R/3 Release	IDoc Version	Control Record Length (EDI_DC)	Data Record Length (EDI_DD)
3.0C to 3.0H	2	464	1055
4.0A and above	3	524	1063

12 Select **Terse** or **Verbose** commenting.

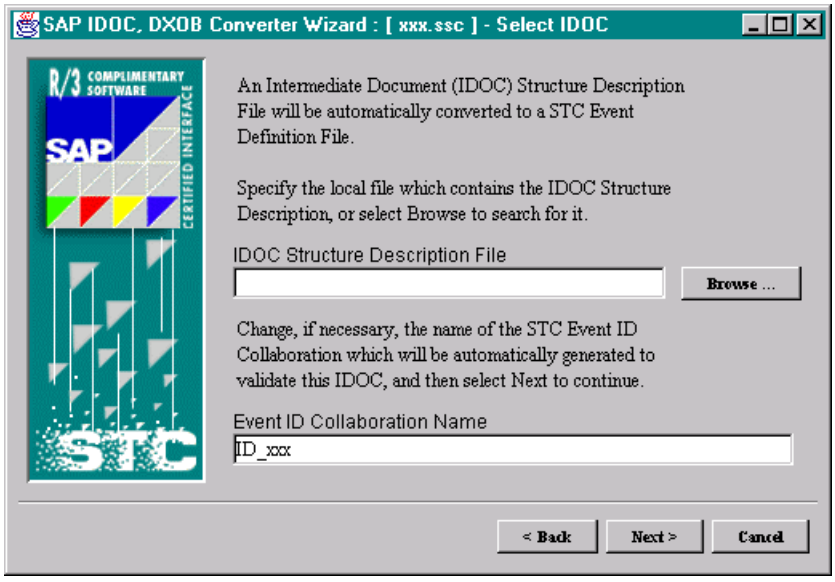
Note: *Do not select the* **Verbose** *option while attempting to use the SAP IDoc/DXOB Converter in a meaningful way, since it leads to copious debugging (this is a result of the STC_TRACEMASK variable setting).*

13 Select the **Character Set**, if applicable.

Note: *If you use the English version of Microsoft Windows, ASCII is the only Character Set option. If you use the Japanese version of Microsoft Windows, you also have the option of using SJIS.*

14 Selecting **Next** displays the following screen.

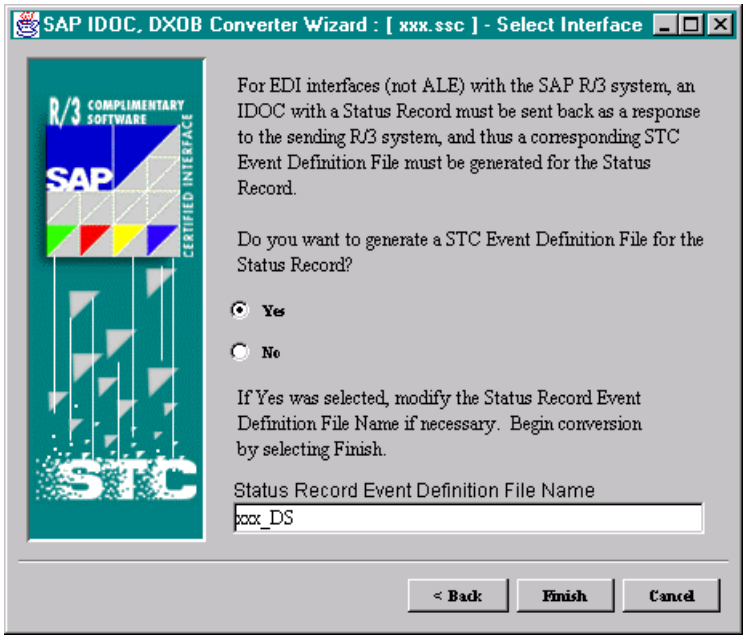
Figure 22 SAP Converter Wizard (2)



15 Enter the path of the **IDoc Description File** you have downloaded to the e*Gate host into the text box, or locate using **Browse**. (Remember that the file must be located on the local file system.)

16 The **Validation Collaboration ID Rule Name** is generated automatically. If you need to modify it for some reason, you may do so. Then select **Next**, which displays the following information:

Figure 23 SAP Converter Wizard (3)

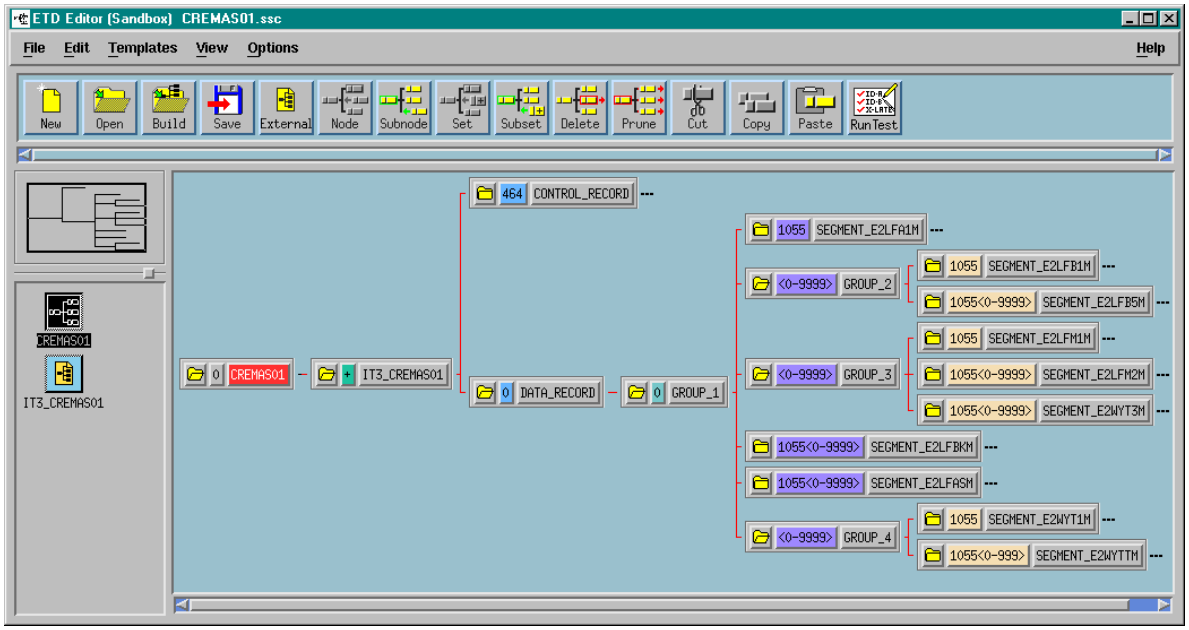


17 Since you are configuring for EDI-mode operation, select **Yes**.

18 Click **Finish** to invoke the conversion process.

19 The *Event Type Definition Editor* window is then displayed, showing the Event Type Definition for the selected file. Refer to the SeeBeyond guide *Creating an End-to-End Scenario with e*Gate Integrator* for instructions on defining Collaboration Rules.

Figure 24 Event Type Definition Editor Window



3.4.3 Using the Java ETD Editor's IDoc Wizard

The Java ETD Editor contains a Java-based IDoc Wizard, which takes an IDoc and converts it to a **.xsc** file.

Note: *When using the SAP EDI e*Way, the IDoc data blob coming in to the Java Collaboration service does not contain Carriage Return-Linefeed (CR-LF) pairs to delimit the data records in the IDoc; instead, each data record is padded out to its full size. Conversely, when sending an IDoc data blob from the Collaboration service side back to the EDI e*Way, the e*Way expects the data records to be fully padded. As a result, it truncates them at the last defined field position using a CR-LF pair before placing the data in a file to be sent to SAP via FTP.*

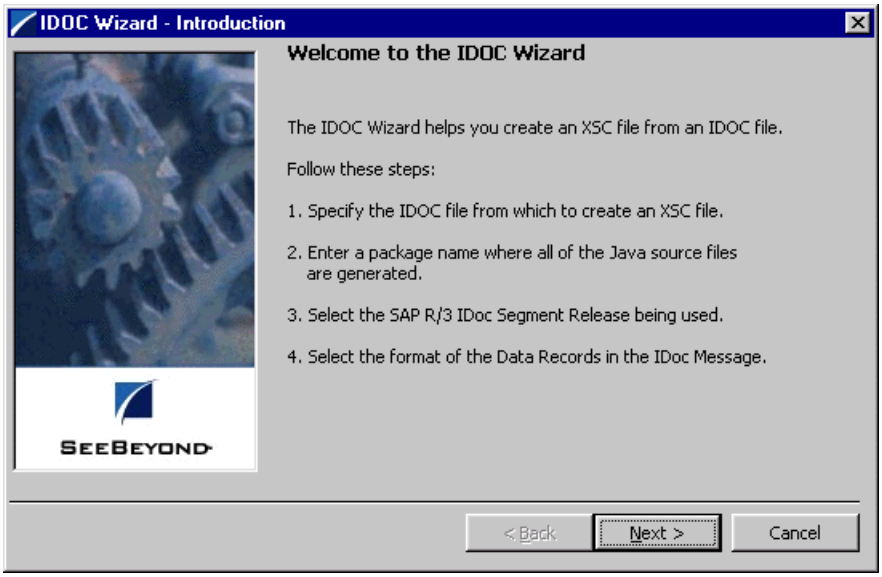
To create an Event Type Definition using the IDoc Wizard

1. Start the e*Gate Schema Designer, and open the schema for which you want to create an IDoc ETD.
2. From the **Options** menu, select **Default Editor** > **Java**.
3. Launch the ETD Editor.
4. Select **New** on the Java ETD Editor's toolbar. The New Event Type Definitions window appears, displaying all installed ETD Wizards.
5. Invoke the IDoc Wizard by selecting its icon (Figure 25).

Figure 25 IDoc Wizard Icon

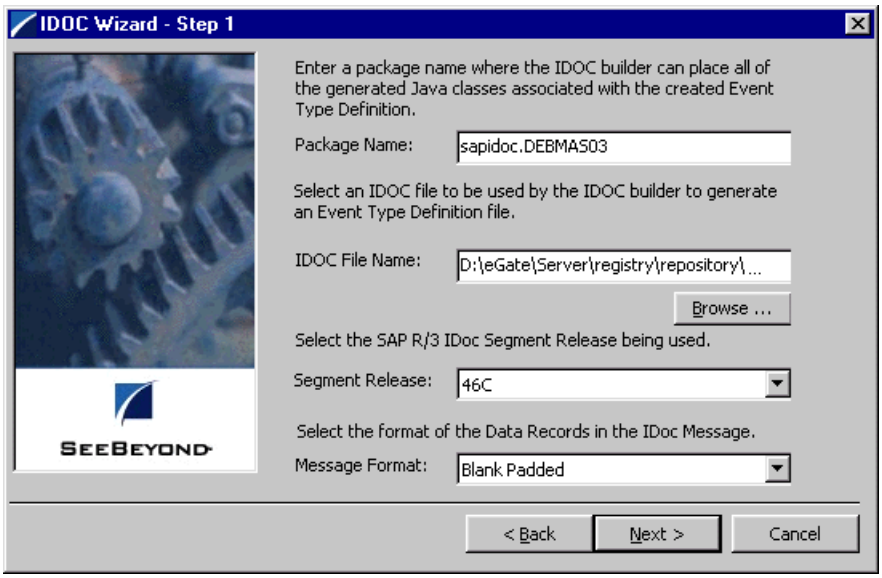
The initial Wizard dialog window now appears (see Figure 26).

Figure 26 IDoc Wizard - Introduction



6 Select **Next**.

Figure 27 IDoc Wizard - Step 1



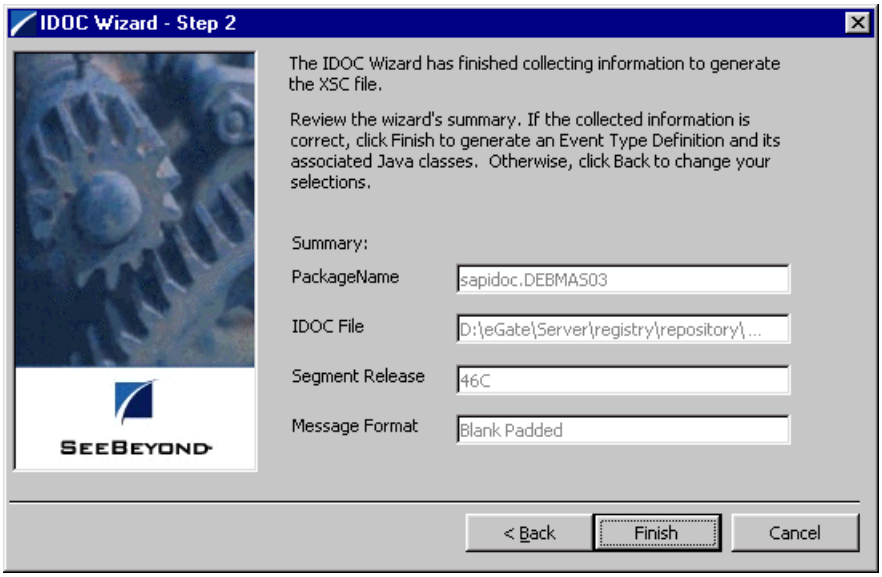
7 Enter a Package Name for the container in which the Wizard places the generated Java classes.

8 Enter an IDoc File Name for the IDoc description file to be used to create this IDoc.

9 Select the SAP segment release version to be used.

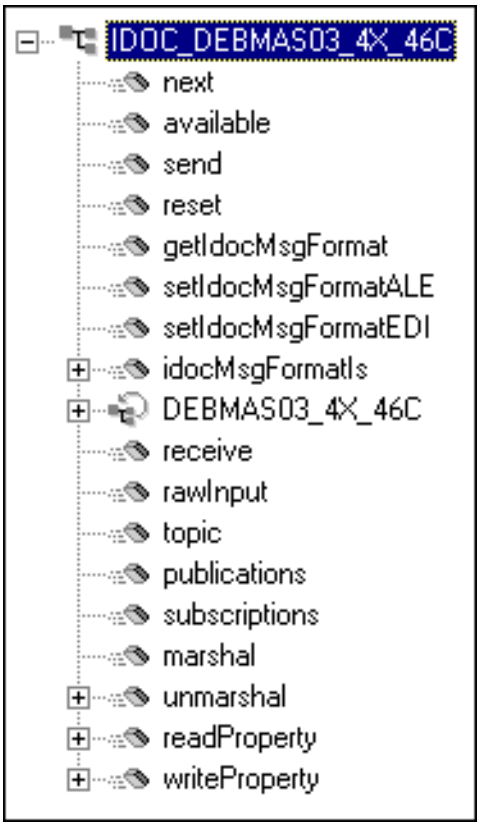
10 Click **Next** to view the next Wizard dialog box.

Figure 28 IDoc Wizard - Step 2



11 Click **Finish** to view the ETD.

Figure 29 Event Type Definition - IDoc Example (DEBMAS03)



The message format for the IDoc is stored in the resulting Java file, shown in this example as **IDOC_DEBMAS03_4X_46C**, in the field **idocMsgFormat**. This field

determines how data is marshaled or unmarshaled. If you want to change the type of data that is being processed in the collaboration, you can use one of the **set** methods; however, you must make sure these **set** methods are called *before* calling the **marshal** and **unmarshal** methods.

Note: *If data is to be sent back to SAP via the SAP EDI e*Way, you also must generate a Monk version of the IDoc ETD (select* **Monk** *as the Default Editor, start the Monk ETD editor, click the* **Build** *button and select* **SAP IDOC, DXOB Converter***) in order for the e*Way to determine the Data Record Segment Dictionary for truncation purpose.*

See **Appendix B** for descriptions of the Java methods associated with the SAP IDoc.

3.5 Assigning ETDs to Event Types

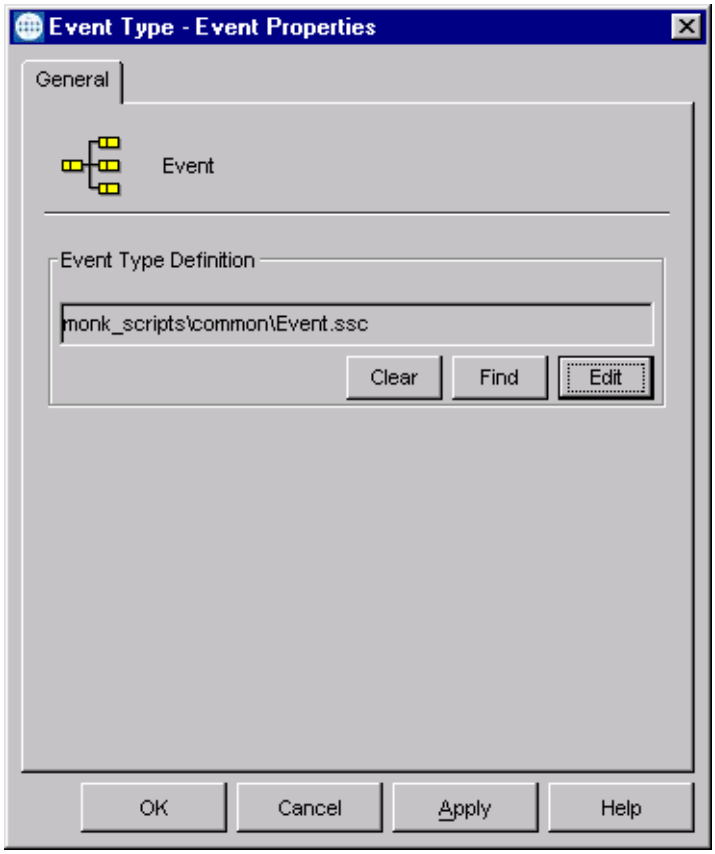
After you have created the e*Gate system's ETD files, you can assign them to Event Types you have already created.

To assign ETDs to Event Types

1. In the Schema Designer window, select the **Event Types** folder in the Navigator/Components pane.
2. In the Editor pane, select one of the Event Types you created.
3. Right-click on the Event Type and select **Properties** (or click  in the toolbar).

   The Event Type Properties dialog box appears. See Figure 30.

Figure 30 Event Type Properties Dialog Box



4. Under Event Type Definition, click **Find**, and the Event Type Definition Selection dialog box appears (it is similar to the Windows Open dialog box).
5. Open the **monk_scripts\common** folder, then select the desired file name (***.ssc**).
6. Click **Select**. The file populates the Event Type Definition field.

7 To save any work in the properties dialog box, click **Apply** to enter it into the system.

8 When finished assigning ETDs to Event Types, click **OK** to close the properties dialog box and apply all the properties.

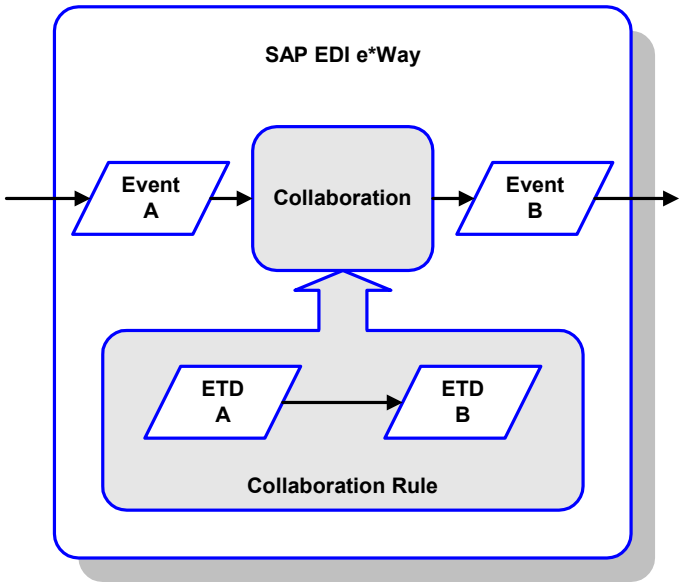
Each Event Type is now associated with the specified Event Type Definition.

3.6 Defining Collaborations

After you have created the required Event Type Definitions, you must define a Collaboration to transform the incoming Event into the desired outgoing Event.

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 or 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.

Figure 31 Collaborations



The Collaboration is driven by a Collaboration Rule script, which defines the relationship between the incoming and outgoing ETDs. You can use an existing Collaboration Rule script, or use the Monk programming language to write a new Collaboration Rule script. Once you have written and successfully tested a script, you can then add it to the system's run-time operation.

Collaborations are defined using the e*Gate Monk Collaboration Rules Editor. See the *e*Gate Integrator User's Guide* for instructions on using this Editor. The file extension for Monk Collaboration Rules is **.tsc**.

3.7 Creating Intelligent Queues

The final step is to create and associate an IQ for the SAP EDI e*Way. 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). See the *e*Gate Integrator User's Guide* for complete information on queuing options and procedures for creating IQs.

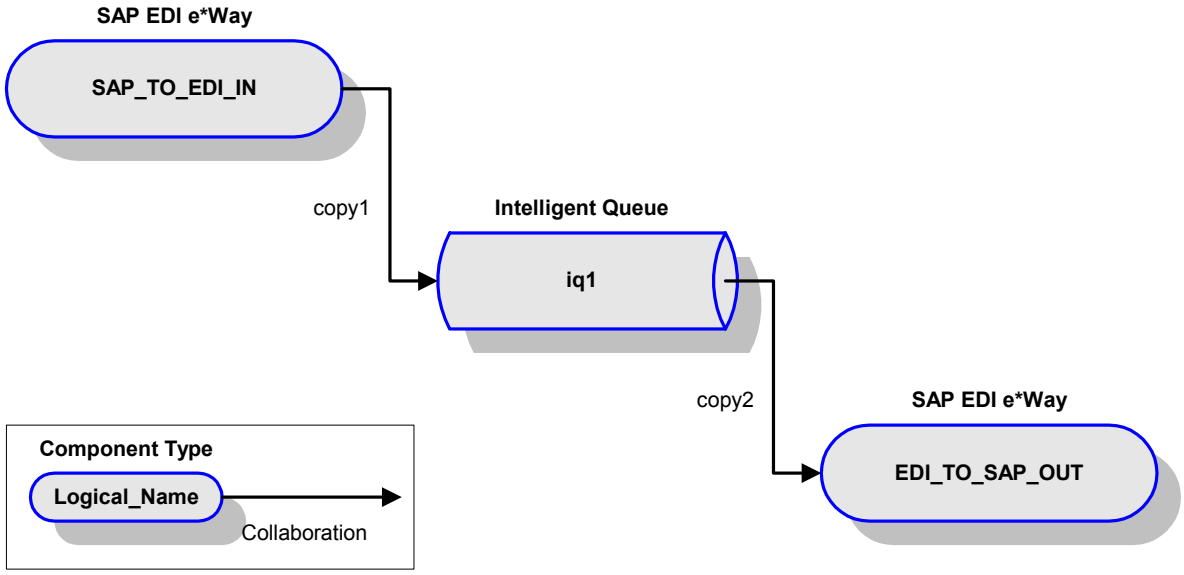
3.8 Sample Schema

3.8.1 Overview

Components

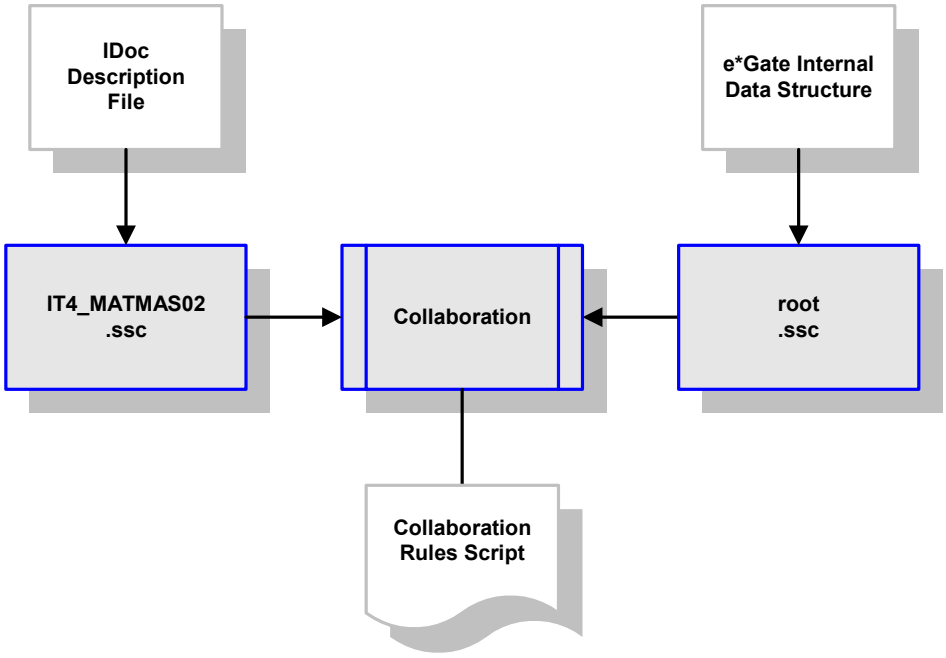
The schema **SAP_EDI** sets up two SAP EDI e*Ways having the logical names (LN) **SAP_TO_EDI_IN** and **EDI_TO_SAP_OUT**, and an Intelligent Queue having the LN **iq1**. These components are shown in Figure 32.

Figure 32 SAP_EDI Schema



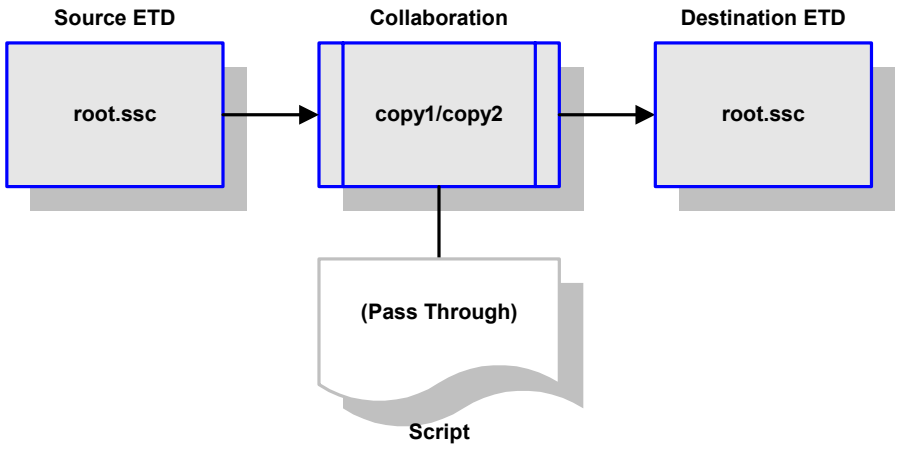
The sample schema also contains two ETDs, representing the SAP IDoc structure and the e*Gate internal structure, that are used to form a Collaboration between SAP and the e*Way as shown in Figure 33.

Figure 33 SAP-e*Gate Collaboration



The internal Collaborations **copy1** and **copy2** are both pass-through Collaborations, which provide a one-to-one copy function between the e*Ways and the IQ (see Figure 34).

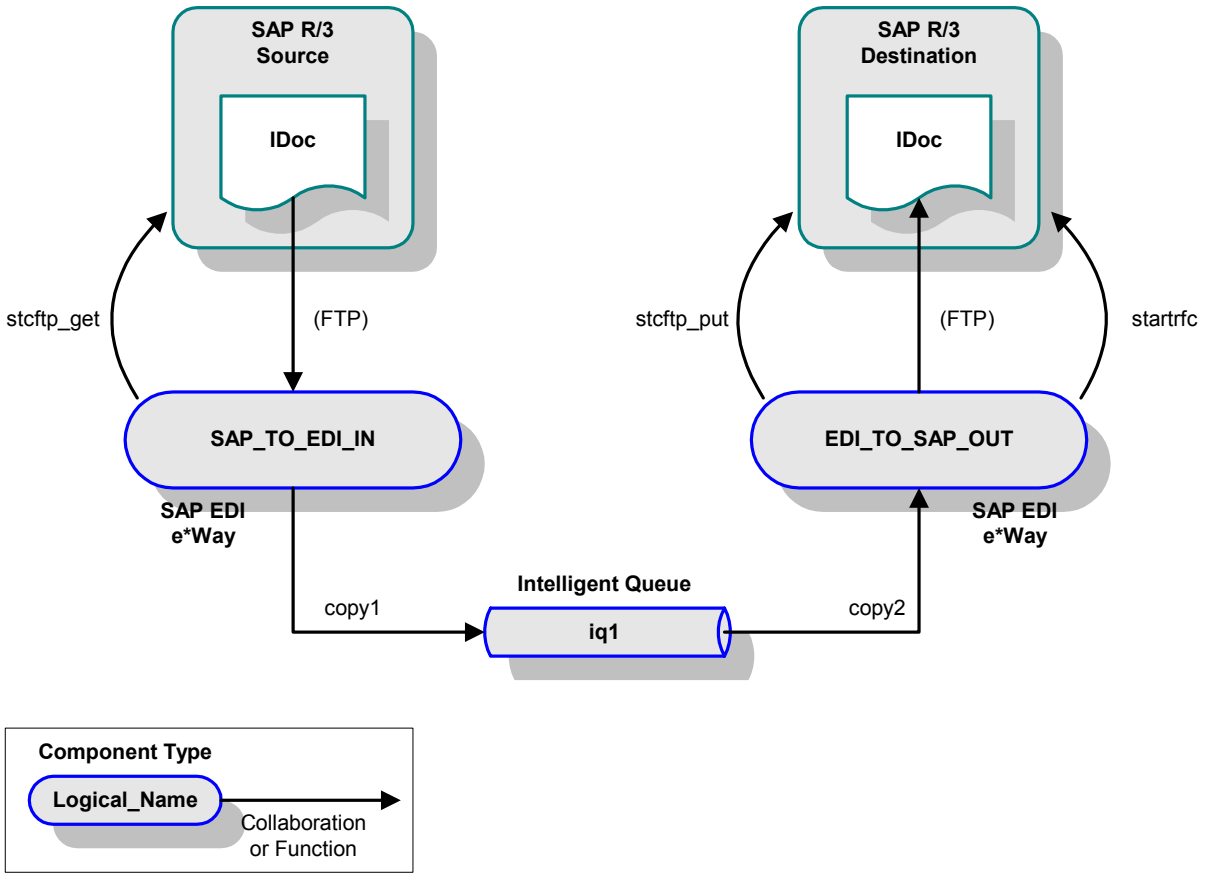
Figure 34 Internal Collaborations



Operation

SAP_TO_EDI_IN uses the script **get_idocs_from_r3** to extract (via FTP) an IDoc file (**matmas02.idocs**) from a SAP R/3 source system. It converts the IDoc format to an e*Gate-internal format using the Collaboration shown in Figure 33, and sends the Event to the IQ by means of the internal Collaboration **copy1**.

Figure 35 Sample Schema Operation

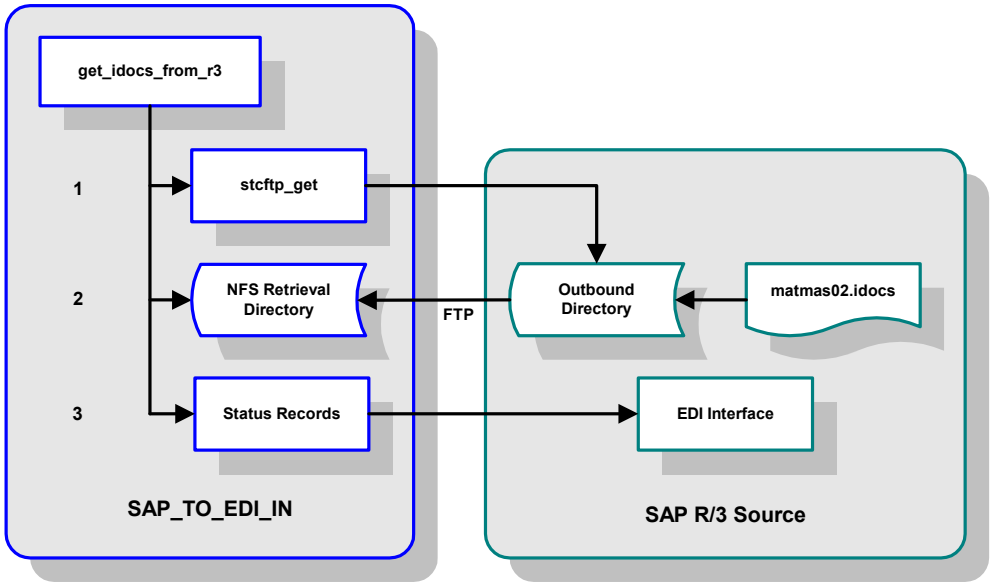


EDI_TO_SAP_OUT uses the script **send_idocs_to_r3** to extract the Event from the IQ by means of the internal Collaboration **copy2**. It then converts it to IDoc format, and sends it to the SAP R/3 destination system (via FTP), again using the Collaboration shown in Figure 33 (now in the e*Gate-to-SAP direction). It then executes **startrfc.exe** which, in turn, notifies the SAP host that there are messages ready to pick up.

3.8.2 Inbound Script: get_idocs_from_r3

Using the script **get_idocs_from_r3**, an e*Gate-inbound EDI e*Way, **SAP_TO_EDI_IN**, polls for and retrieves a data file in IDoc format from the remote host (a sample file, **matmas02.idocs**, is provided). This e*Way picks up the IDoc file from a designated location via FTP and puts the IDoc messages contained in the file into a queue.

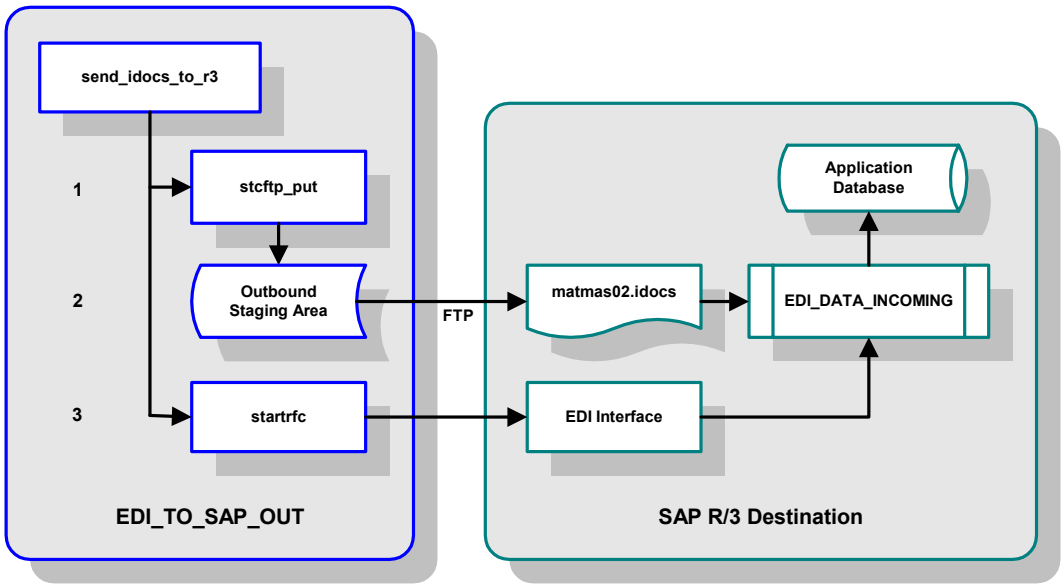
Figure 36 get_idocs_from_r3 Script



3.8.3 Outbound Script: send_idocs_to_r3

Using the script **send_idocs_to_r3**, an e*Gate-outbound EDI e*Way, **EDI_TO_SAP_OUT**, retrieves the messages from the queue and puts them in a designated output location. The outbound EDI e*Way then FTPs an IDoc file that contains the output messages to the remote host and executes **startrfc.exe**. This, in turn, notifies the SAP host that there are messages ready to pick up.

Figure 37 send_idocs_to_r3 Script



3.8.4 Configuring the Sample Schema

1 Follow the installation procedure given in **"Optional Example Files" on page 22**, or perform the following tasks in the e*Gate Schema Designer GUI:

- Create a new schema named **SAP_EDI**
- Add an e*Way named **SAP_TO_EDI_IN** to your CB, selecting the configuration file **SAP_TO_EDI_IN.cfg**
- Add an e*Way named **EDI_TO_SAP_OUT** to your CB, selecting the configuration file **EDI_TO_SAP_OUT.cfg**

2 Make a copy of the file **get_idocs_from_r3.bat** (for Windows) or **get_idocs_from_r3.sh** (for UNIX), and specify the following fields:

- **r3Home**
- **r3Outbound**
- **remoteDestHost**
- **remoteType**
- **localDestHost**

3 Make a copy of the file **send_idocs_to_r3.bat** (for Windows) or **send_idocs_to_r3.sh** (for UNIX), and specify the following fields:

- **r3Home**
- **r3Inbound**
- **remoteDestHost**
- **remoteType**
- **localDestHost**
- **localDestFname**
- **localArchDir**
- **SID**
- **sapClient**
- **sapSysId**
- **sapGwService**
- **sapLogicalPort**

Note: *In UNIX, make sure that shell scripts are executable and do not have* **^M** *at the end.*

4 Make a copy of the file **SAP_TO_EDI_IN.cfg** and specify the following parameters (the specified parameters should match the values used in steps 3 and 4):

- Auxiliary Polling Script (e.g. **bin/get_idocs_from_r3.sh**)
- File Transport User Name
- File Transport User Password

- SAP User Name
- SAP User Password
- Status Record Trigger Script (e.g. **bin/send_status_records_to_r3.sh**)

5 Make a copy of the file **EDI_TO_SAP_OUT.cfg** and specify the following parameters (the specified parameters should match the values used in steps 3 and 4):

- File Transport User Name
- File Transport User Password
- SAP User Name
- SAP User Password

3.8.5 Running the Sample Schema

1 From the control panel, double-click **Services**

2 Locate and select **E*Gate Control Broker SAP_EDI**

3 Click **Start**

4 Log in to SAP R/3

5 Go to transaction **BALE**

6 Select the menu path **Monitoring > IDoc Overview**

7 Specify the date range and **Execute**.

8 You should see **MATMASxx** and the number should increment by 1 (i.e., if you started with **MATMAS02**, the file name should become **MATMAS03**)

Chapter 4

SAP R/3 Configuration

This chapter describes procedures for preparing your SAP R/3 system to interact with e*Gate.

4.1 Overview

Before you have an operational SAP-e*Gate system, your existing SAP R/3 system must be configured to interact properly with the SAP EDI e*Way. This configuration is non-intrusive, and does not interfere with other SAP R/3 operations.

The topics described in this chapter include the following:

Note: *The screen captures shown in this chapter correspond to SAP Frontend Server release 4.5B, and R/3 Kernel release 4.0B. They are shown simply to illustrate the general nature of the procedures, and contain only example values. They are not meant to replace the SAP documentation supplied with your system. The procedures for releases 4.0, 4.5 and 4.6 vary in detail, but are similar in a logical sense. Please refer to your SAP documentation to determine the exact procedures and corresponding appearance of the GUI for your installation.*

4.2 SAP Hierarchies

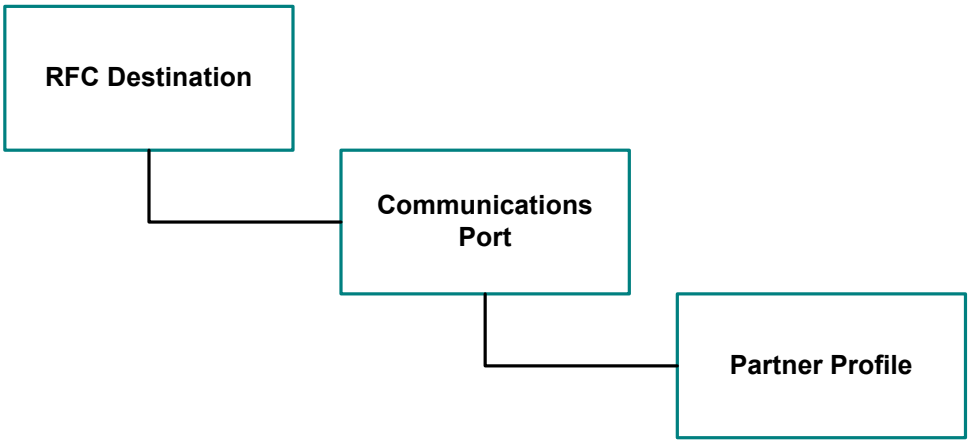
EDI and ALE share many common features, including the IDoc interface, types, and structures. EDI processing also uses the same R/3 function modules used by ALE. As a result, building an EDI interface is very similar to building an ALE interface, and message types available for EDI are a subset of those used for ALE.

EDI is used primarily for exchanging transactional data and documents, however, and not master data or control data. This leads to a number of differences, including the following:

- Partner Profiles do not need to be based on a Logical System
- EDI does not use the Distribution Model concept used by ALE
- EDI Ports are typically file-based

As a result, you only need to define Communications parameters in SAP to specify the correct routing of IDocs (either inbound to or outbound from SAP). The hierarchy of this Communication system is shown in Figure 38.

Figure 38 Communications Hierarchy



The RFC Destination defines the entity to which Remote Function Calls (RFCs) can be made. The Communications Port defines a channel for communication of IDocs, and the Partner Profile acts as an identifier for the e*Gate system.

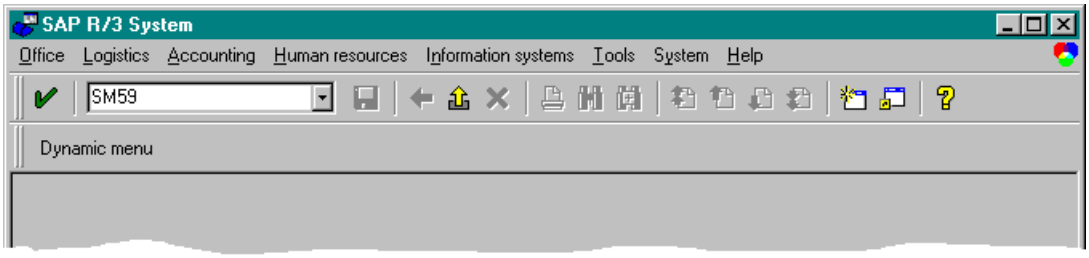
4.3 Communications Configuration

4.3.1 Defining the RFC Destination

Transaction: SM59

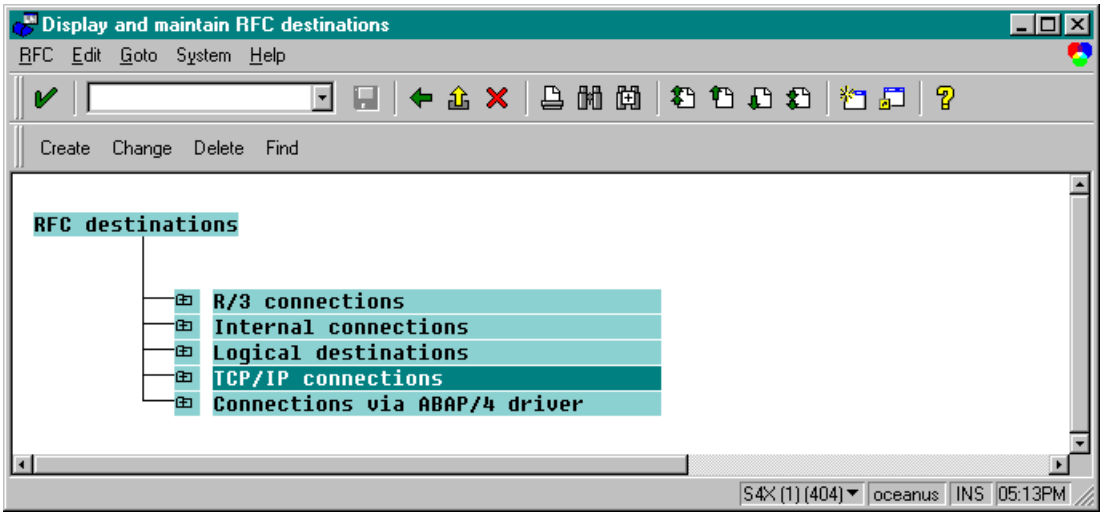
In this step you create a logical RFC destination for use in the Command File of the Port definition.

Figure 39 SAP R/3 System Window



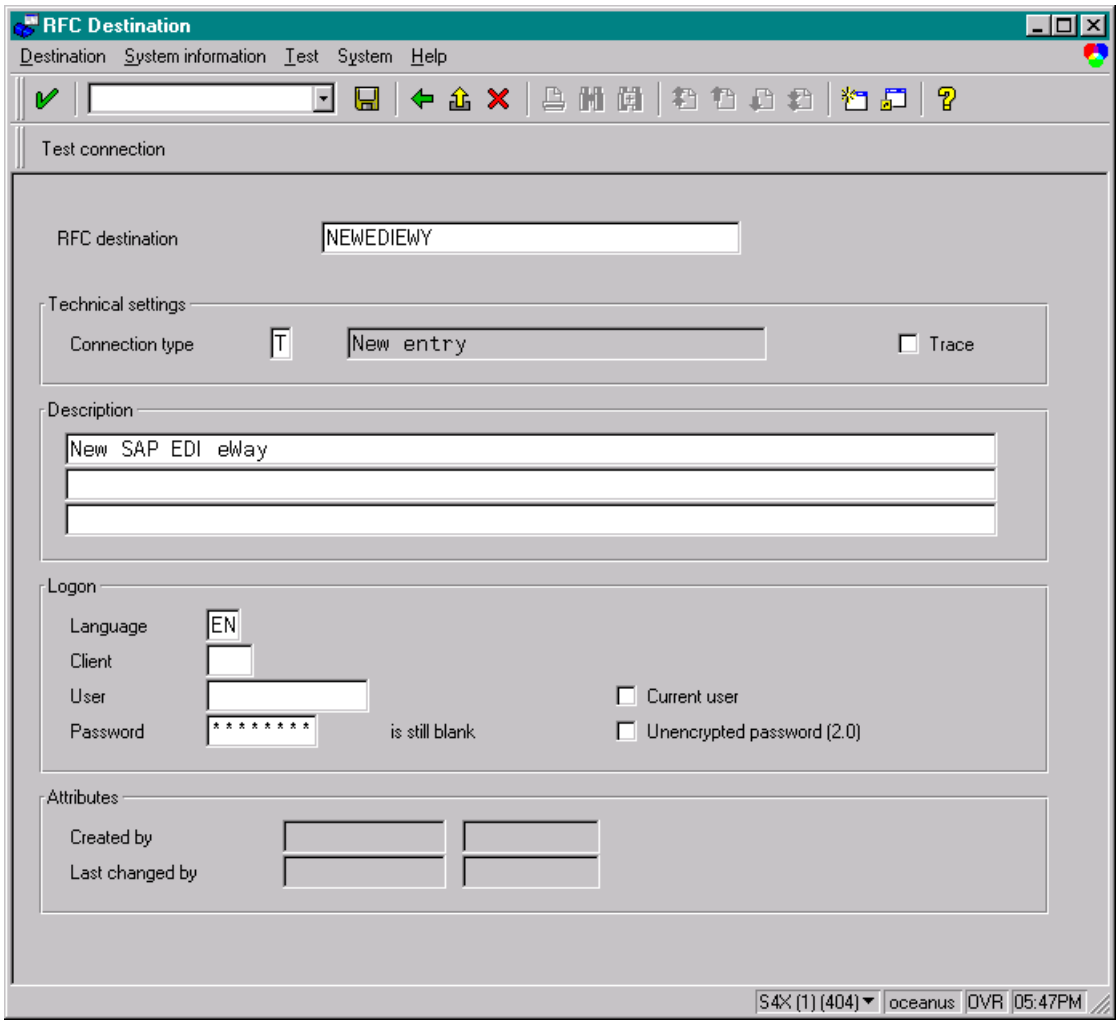
1 In the *SAP R/3 System* home window, type **SM59** into the command field and **Enter** to display the *RFC Destination Maintenance* window.

Figure 40 RFC Destination Maintenance Window



2 Select the **TCP/IP connections** option and then **Create**, which displays the *RFC Destination* entry window.

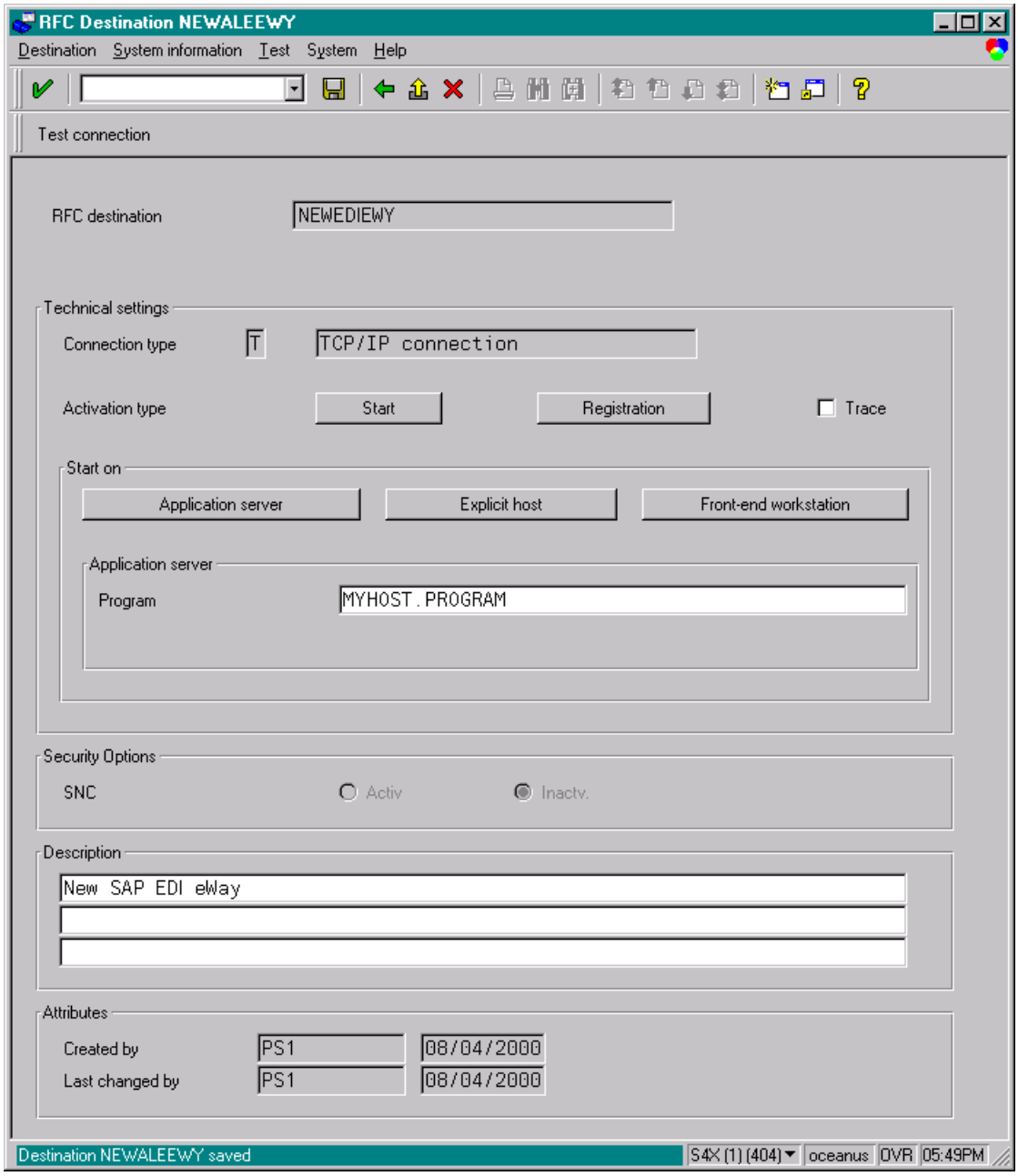
Figure 41 RFC Destination Entry Window



3 Type in the name of the **RFC Destination** you wish to use, an accompanying **Description**, and enter <**T**> for the Connection Type (TCP/IP).

4 **Save** , which returns a confirmation message and displays the *RFC Destination* window corresponding to your entry.

Figure 42 RFC Destination Window



5 Select **Registration** for the **Activation type**, type in a **Program ID** of the form <**hostname**>.<**program name**>, and a **Description**.

6 Select **Test connection**, which tests the connection for logon speed and message transfer speed. When the e*Way is running, the results are displayed in a table; otherwise, return code 3 is displayed.

Figure 43 Connection Test Results

Connection test STCDGW	
Connection type:	TCP/IP connection
Logon:	255 msec
0 KB:	496 msec
10 KB:	491 msec
20 KB:	504 msec
30 KB:	505 msec

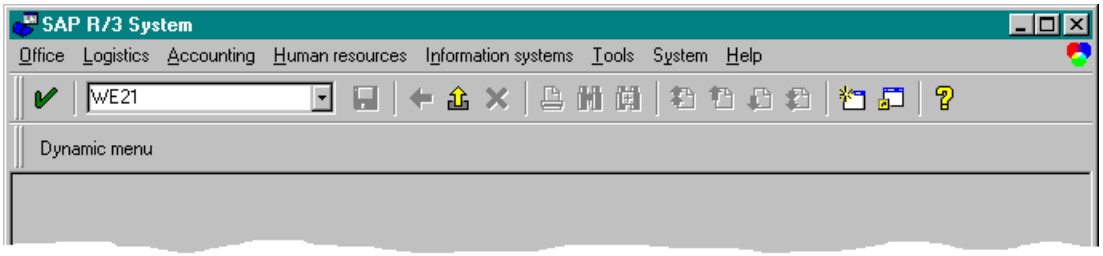
7 Save and select repeatedly to return to the *SAP R/3 System* window.

4.3.2 Defining the Communications Port

Transaction: WE21

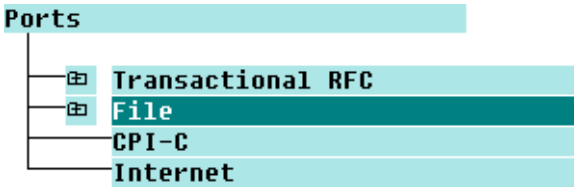
In this step you specify the outbound file name, directory path, and any associated function modules.

Figure 44 SAP R/3 System Window



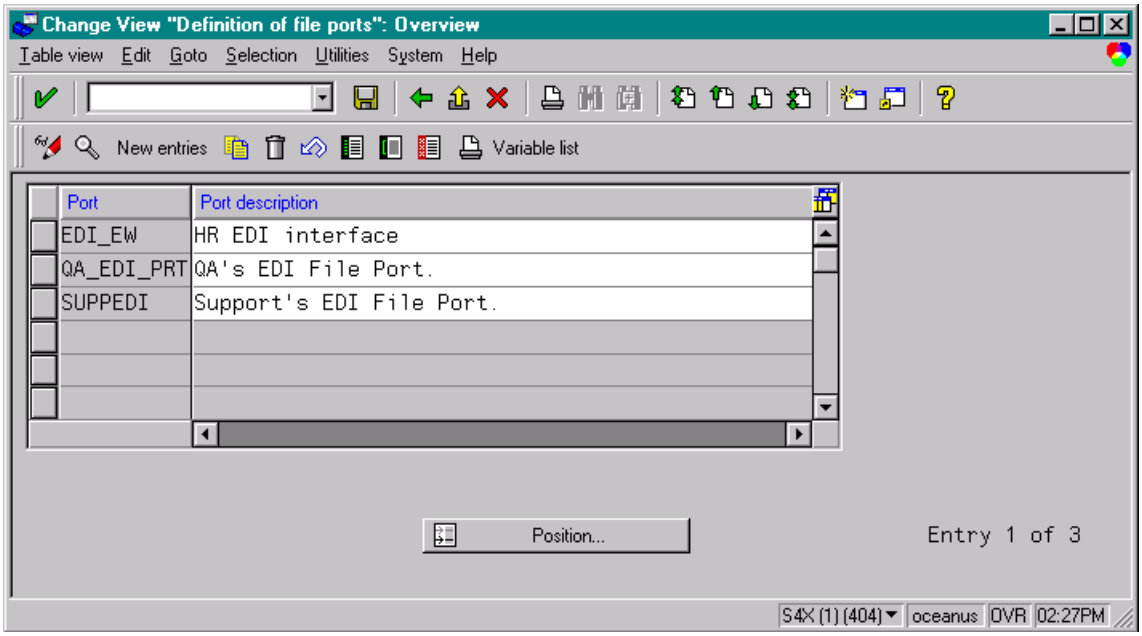
1 In the *SAP R/3 System* home window, type **WE21** into the command field and enter to display the *WF-EDI Port Definition* window.

Figure 45 WF-EDI Port Definition Tree



2 A new Port can be added, or an existing Port changed, by highlighting **File** and then selecting **Change** , which displays the *Definition of File Ports Overview* window. (Note that existing Port Names cannot be changed.)

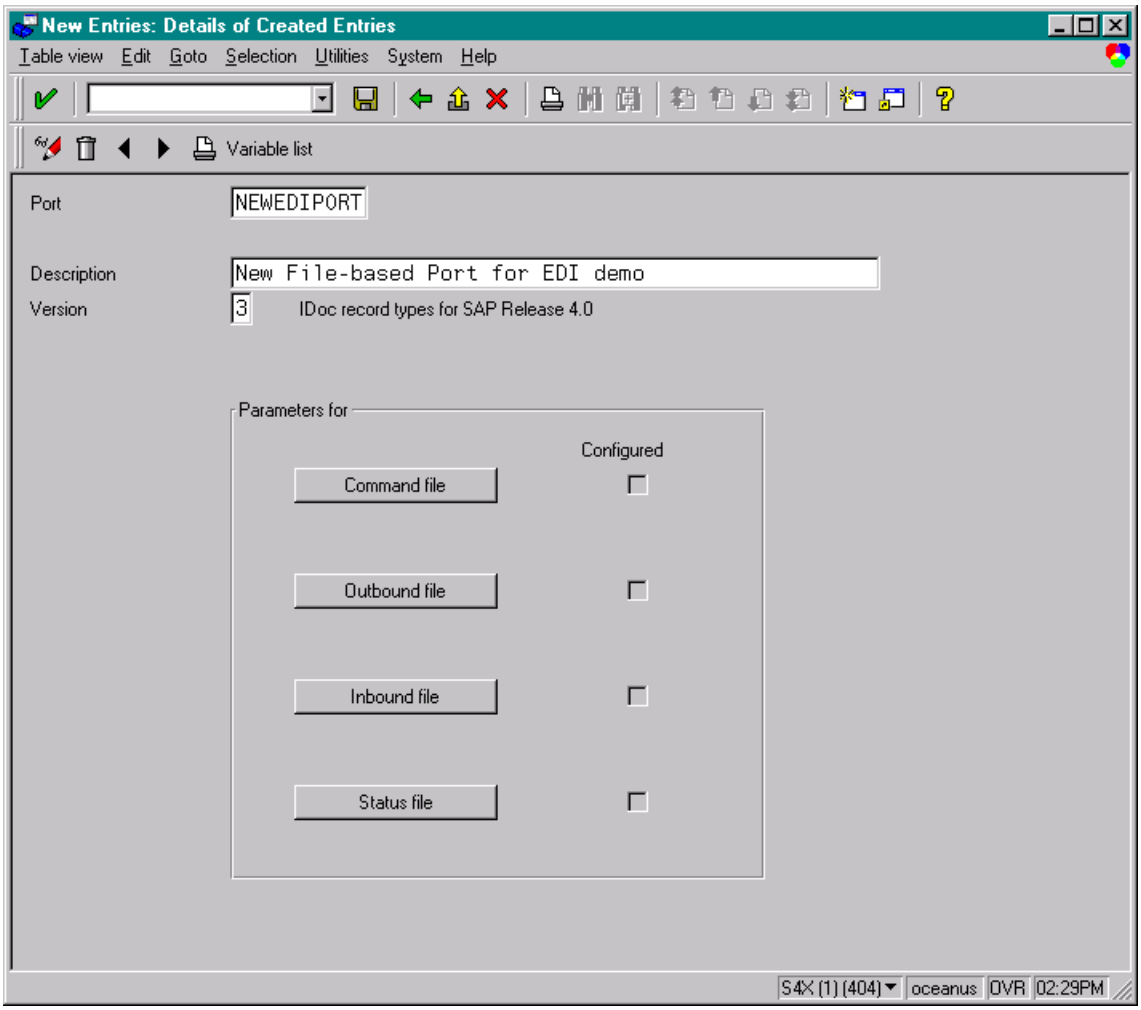
Figure 46 Definition of File Ports Overview Window



3 Changes can be made to existing entries by selecting the Change/Display toggle button  , which opens allowed fields that may be grayed out initially.

4 A new entry can be created by selecting **New Entries**, which displays the *New Entries* detail window.

Figure 47 New Entries (Ports) Window



5 Enter the new Port name, Description, and Version, then **Save**.

6 Selecting the appropriate button displays the parameters for the Input, Output, or Status files. For example, selecting **Outbound file** displays the *Parameters for Outbound File* window.

Figure 48 Parameters for Outbound File

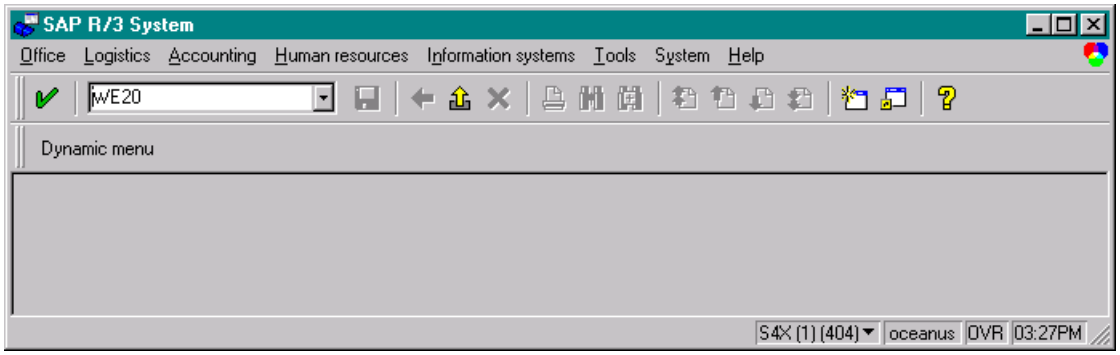
7 Make the necessary entries and **Enter** .

8 **Save** any changes you have made and select repeatedly to return to the *SAP R/3 System* window.

4.3.3 Creating a Partner Profile

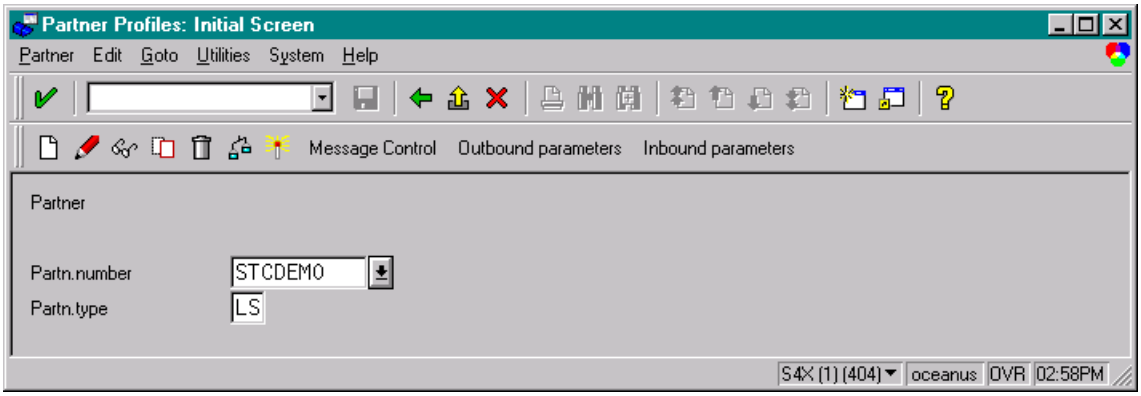
Transaction: WE20

Figure 49 SAP R/3 System Window



1 In the SAP R/3 System home window, type **WE20** into the command field and **Enter** to display the *Partner Profile: Initial Screen* window.

Figure 50 Partner Profile: Initial Screen Window

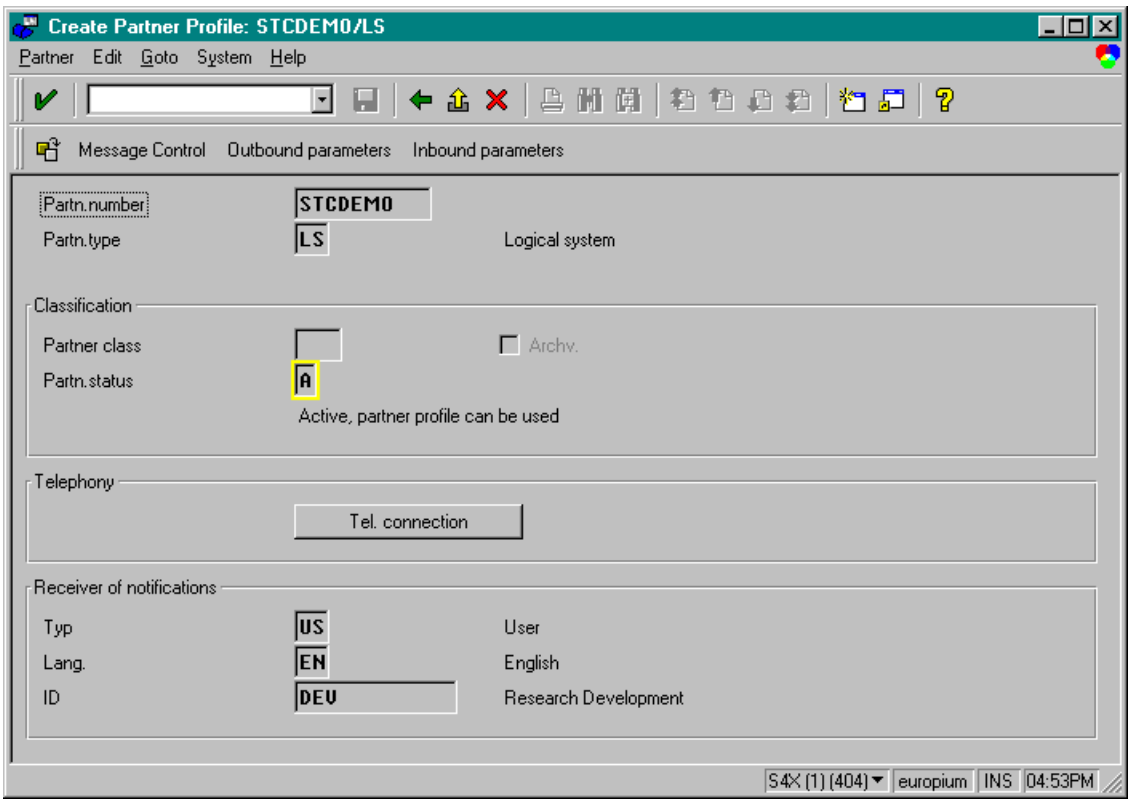


2 Type the desired name (e.g., **STCDEMO**) into the Partner number field, select the appropriate Partner type, and select . This creates the Partner, and displays the *Create Partner Profile <Partner Number>* window.

Note that the Partner Types supported by EDI are:

- **B** (Bank)
- **KU** (Customer)
- **LI** (Vendor)
- **LS** (Logical system)

Figure 51 Create Partner Profile Window



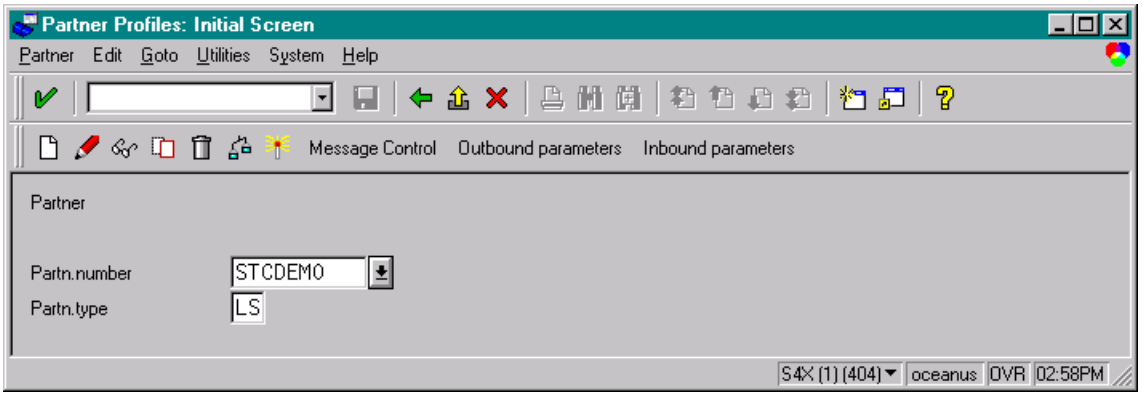
3 Select **A** for the Partner status (Active) and **Save** . You now have created the Partner, and need to continue to the next section to configure the Partner Profile.

4.3.4 Configuring a Partner Profile

Transaction: WE20

In this step you configure the Partner Profiles for the SAP Client and the e*Way for the Partner type.

Figure 52 Partner Profile: Initial Screen



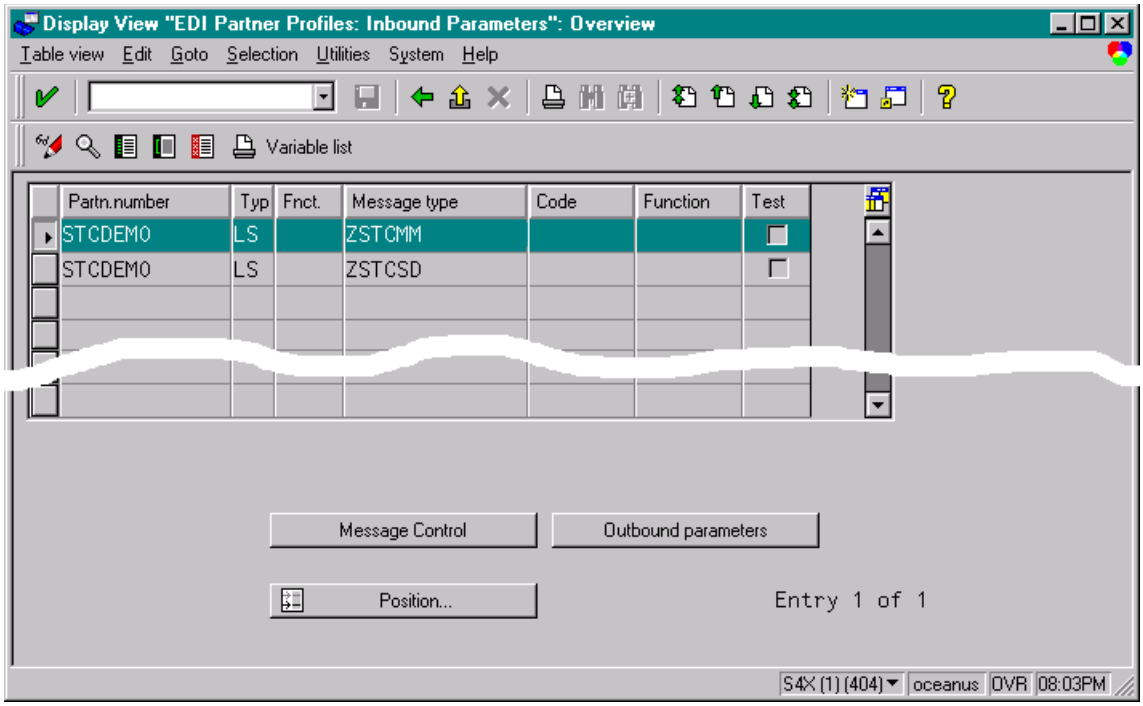
4 In the *Partner Profile: Initial Screen* window, select the desired Partner Number, for example **STCDEMO**.

5 Selecting Inbound parameters then displays either the

C *EDI Partner Profile: Inbound Parameters Overview* window, if there are multiple entries for **STCDEMO**.

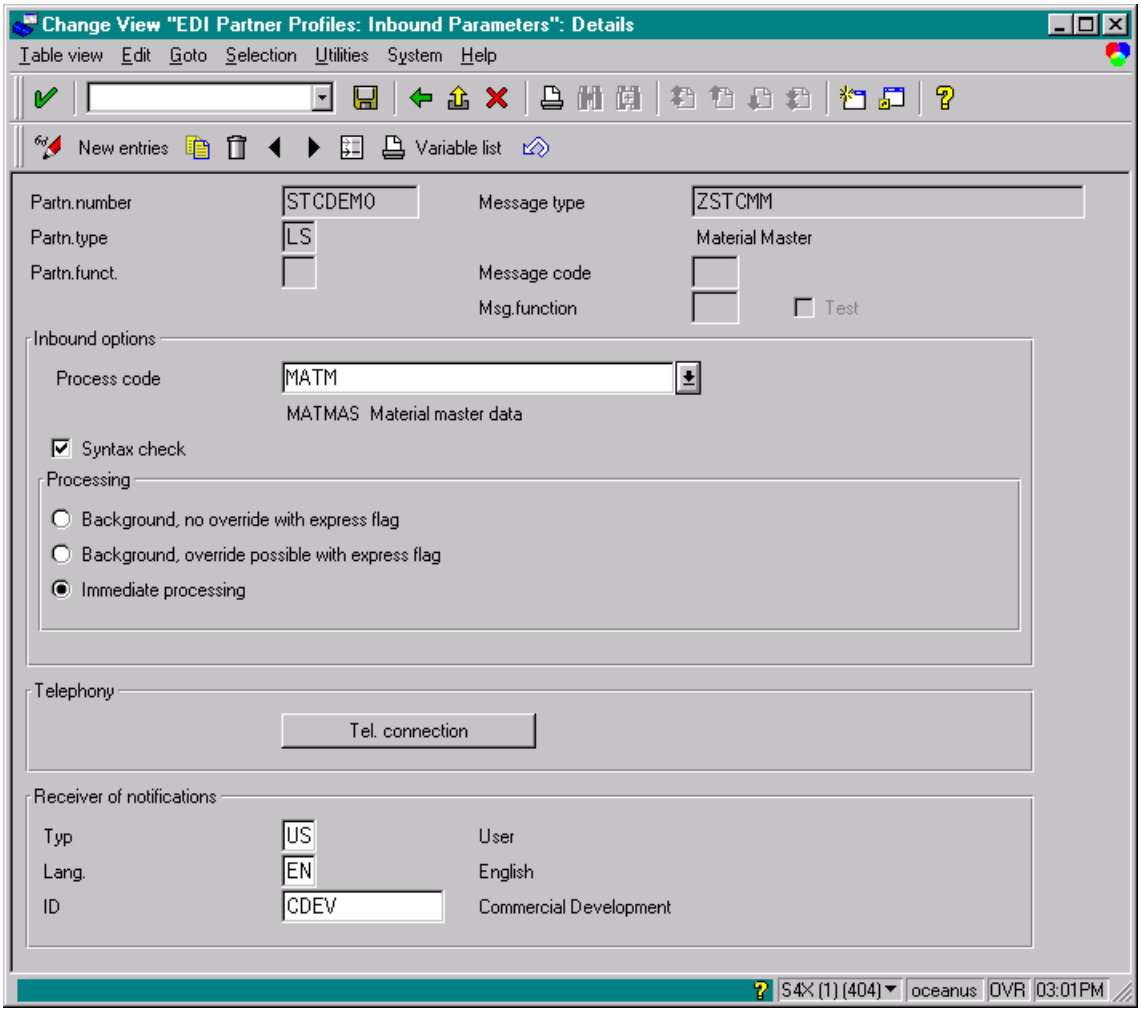
D *EDI Partner Profile: Inbound Parameters Detail* window, if there is a single entry for **STCDEMO**.

Figure 53 EDI Partner Profile: Inbound Parameters Overview Window



6 If there are multiple entries, select the desired line in the *Overview* window and click **Details** . This displays the *Details* window for that selection.

Figure 54 EDI Partner Profile: Inbound Parameters Details Window



7 Here you set the Inbound options to the desired Process code (e.g., **MATM**).

8 After making your entries, **Save** and **Back** to the main *SAP R/3 System* window.

Chapter 5

Setup Procedures

This chapter describes the procedures for customizing the SAP EDI e*Way to operate with your SAP system.

5.1 Overview

After creating a schema, you must instantiate and configure the SAP EDI e*Way to operate within the schema. A wide range of setup options allow the e*Way to conform to your system's operational characteristics and your facility's operating procedures.

The topics discussed in this chapter include the following:

Setting Up the e*Way

Troubleshooting the e*Way

5.2 Setting Up the e*Way

Note: *The e*Gate Schema Designer GUI runs only on the Windows operating system.*

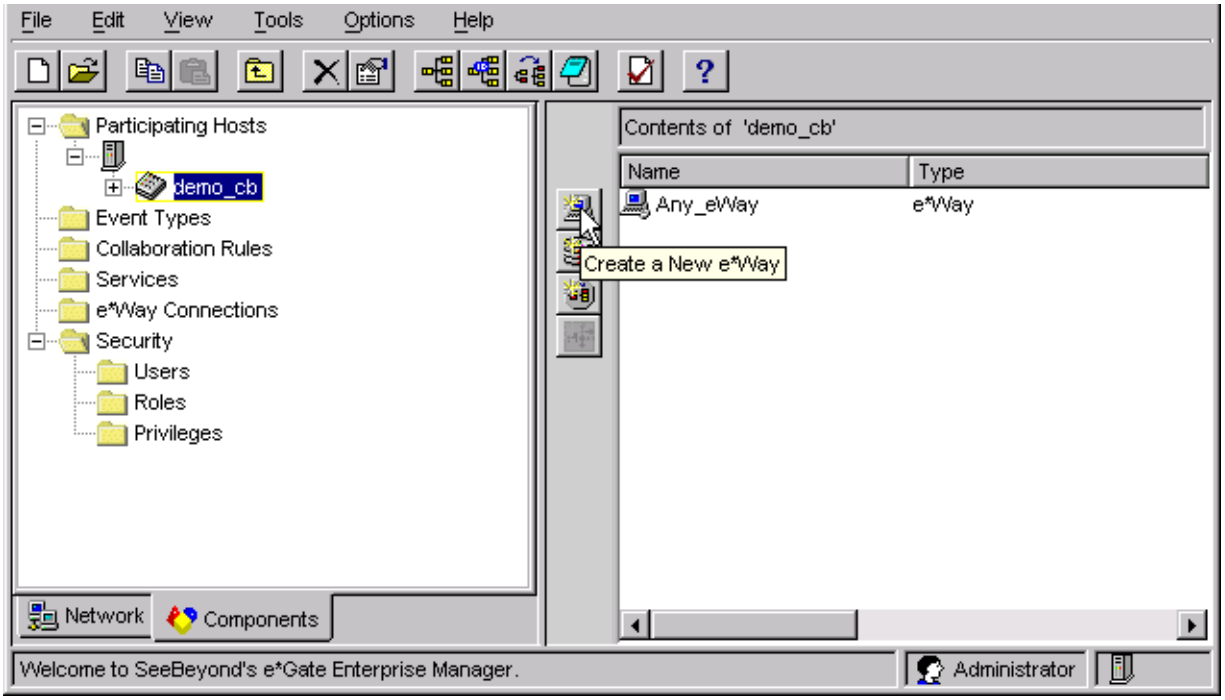
5.2.1 Creating the e*Way

The first step in implementing an e*Way is to define the e*Way component using the e*Gate Schema Designer.

To create an e*Way

1. Open the schema in which the e*Way is to operate.
2. Select the e*Gate Schema Designer Navigator's **Components** tab.
3. Open the host on which you want to create the e*Way.
4. Select the Control Broker you want to manage the new e*Way.

Figure 55 e*Gate Schema Designer Window (Components View)



5. On the Palette, click **Create a New e*Way**.
6. Enter the name of the new e*Way, then click **OK**.
7. All further actions are performed in the e*Gate Schema Designer Navigator's **Components** tab.

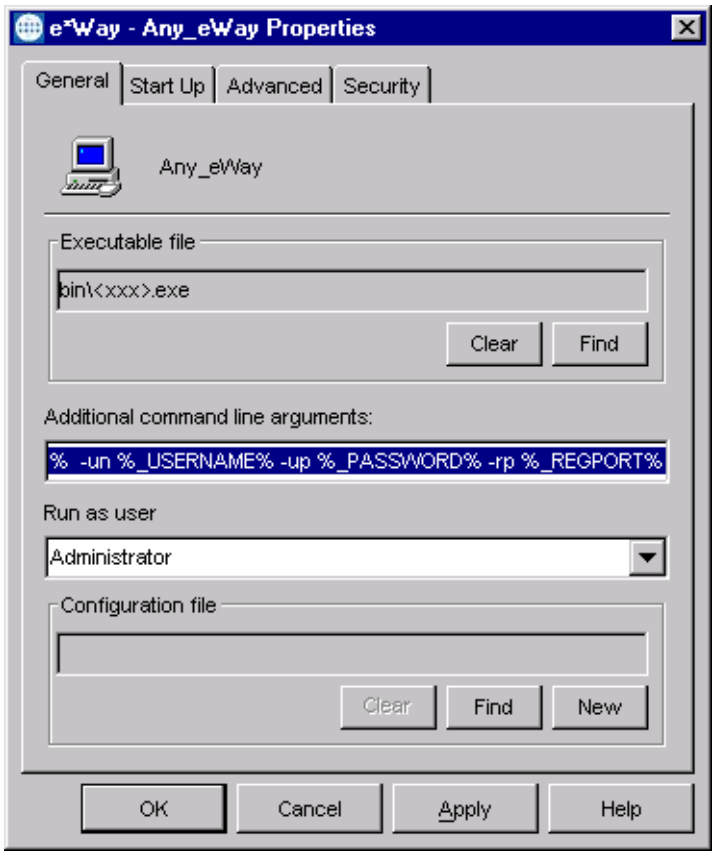
5.2.2 Modifying e*Way Properties

To modify any e*Way properties

1 Right-click on the desired e*Way and select **Properties** to edit the e*Way's properties. The properties dialog opens to the **General** tab (shown in Figure 56).

Note: *The executable file is* **stcewsapedi.exe**.

Figure 56 e*Way Properties (General Tab)



2 Make the desired modifications, then click **OK**.

5.2.3 Configuring the e*Way

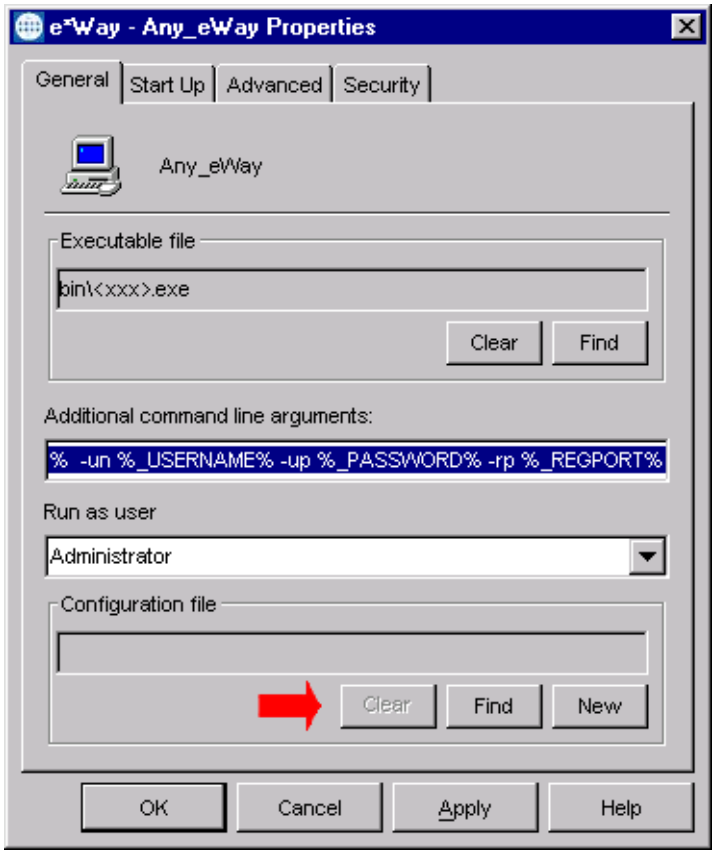
The e*Way's default configuration parameters are stored in an ASCII text file with a **.def** extension. The e*Way Editor provides a simple graphical interface for viewing and changing those parameters to create a working configuration (**.cfg**) file.

To change e*Way configuration parameters

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

Note: *The default configuration files are* **stcewsapediin.def** *and* **stcewsapediout.def**.

Figure 57 e*Way Properties - General Tab

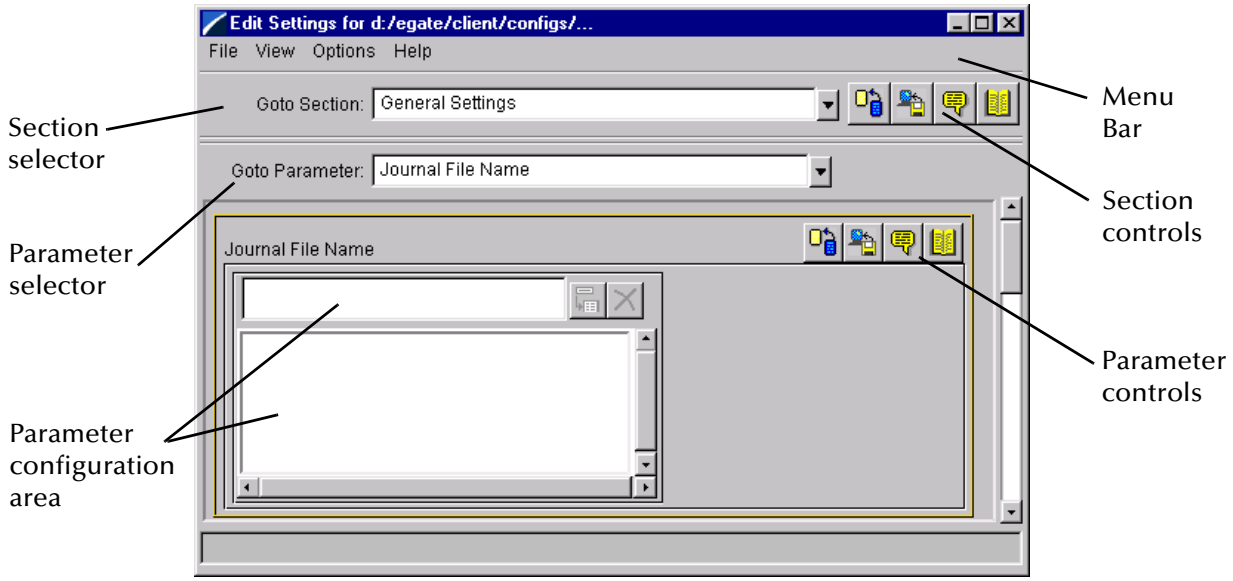


2 Under **Configuration File**, click **New** to create a new file or **Find** to select an existing configuration file. If you select an existing file, an **Edit** button appears. Click this button to edit the currently selected file.

3 You are now in the e*Way Configuration Editor.

Using the e*Way Editor

Figure 58 The e*Way Configuration Editor



The e*Way Editor controls fall into one of six categories:

- The **Menu bar** allows access to basic operations (e.g., saving the configuration file, viewing a summary of all parameter settings, and launching the Help system)
- The **Section selector** at the top of the Editor window enables you to select the category of the parameters you wish to edit
- **Section controls** enable you to restore the default settings, restore the last saved settings, display tips, or enter comments for the currently selected section
- The **Parameter selector** allows you to jump to a specific parameter within the section, rather than scrolling
- **Parameter controls** enable you to restore the default settings, restore the last saved settings, display tips, or enter comments for the currently selected parameter
- **Parameter configuration controls** enable you to set the e*Way's various operating parameters

Section and Parameter Controls

The section and parameter controls are shown in Table 9 below.

Table 9 Parameter and Section Controls

Button	Name	Function
	**Restore Default**	Restores default values
	**Restore Value**	Restores saved values
	**Tips**	Displays tips
	**User Notes**	Enters user notes

Note: *The **section controls** affect **all** parameters in the selected section, whereas the **parameter controls** affect only the **selected** parameter.*

Parameter Configuration Controls

Parameter configuration controls fall into one of two categories:

- Option buttons
- Selection lists, which have controls as described in Table 10

Table 10 Selection List Controls

Button	Name	Function
	**Add to List**	Adds the value in the text box to the list of available values.
	**Delete Items**	Displays a “delete items” dialog box, used to delete items from the list.

Command-line Configuration

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.

Getting Help

To launch the e*Way Editor's Help system

From the **Help** menu, select **Help topics.**

To display tips regarding the general operation of the e*Way

From the **File** menu, select **Tips.**

To display tips regarding the selected Configuration Section

In the **Section** Control group, click  .

To display tips regarding the selected Configuration Parameter

In the **Parameter** Control group, click  .

Note: *"Tips" are displayed and managed separately from the Help system that launches from the Toolbar's Help menu. You cannot search for Tips within the Help system, or view Help system topics by requesting Tips.*

For detailed descriptions and procedures for using the e*Way Configuration Editor, see the *e*Gate Integrator User's Guide*.

5.2.4 Changing the User Name

Like all e*Gate executable components, e*Ways run under an e*Gate user name. By default, all e*Ways run under the **Administrator** user name. You can change this if your site's security procedures so require.

To change the user name

1. Display the e*Way's properties dialog.
2. On the **General** tab, use the **Run as user** list to select the e*Gate user under whose name this component is to run.

See the *e*Gate Integrator System Administration and Operations Guide* for more information on the e*Gate security system.

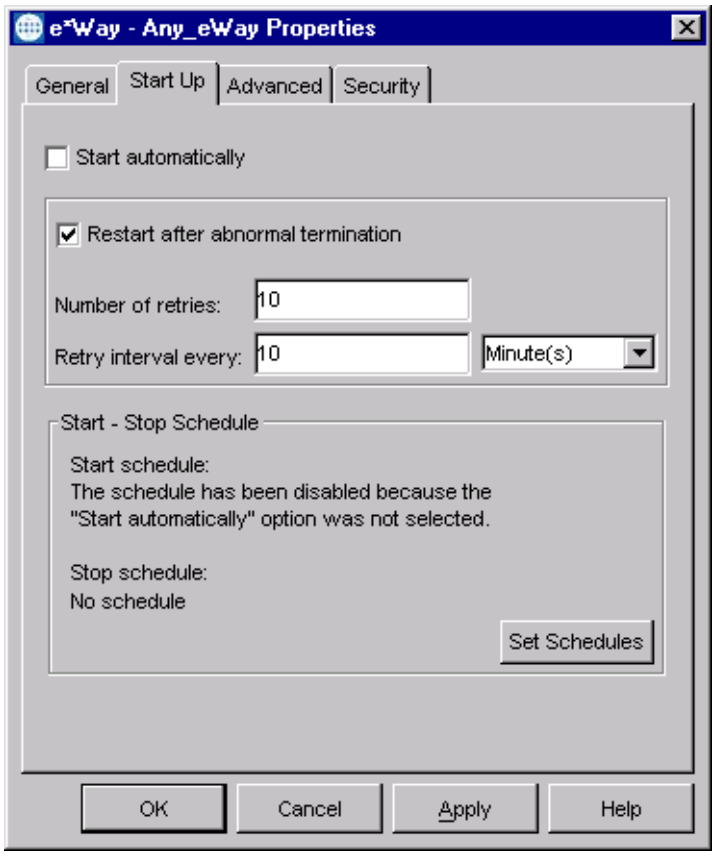
5.2.5 Setting Startup Options or Schedules

SeeBeyond e*Ways can be started or stopped by any of the following methods:

- The Control Broker can start the e*Way automatically whenever the Control Broker starts.
- The Control Broker can start the e*Way automatically whenever it detects that the e*Way terminated execution abnormally.
- The Control Broker can start or stop the e*Way on a schedule that you specify.
- Users can start or stop the e*Way manually using an interactive monitor.

You determine how the Control Broker starts or shuts down an e*Way using options on the e*Way properties **Start Up** tab (see Figure 59). See the *e*Gate Integrator System Administration and Operations Guide* for more information about how interactive monitors can start or shut down components.

Figure 59 e*Way Properties (Start-Up Tab)



To set the e*Way's startup properties

1. Display the e*Way's properties dialog.
2. Select the **Start Up** tab.
3. To have the e*Way start automatically when the Control Broker starts, select the **Start automatically** check box.
4. To have the e*Way start manually, clear the **Start automatically** check box.
5. To have the e*Way restart automatically after an abnormal termination:
   - A Select **Restart after abnormal termination.**
   - B Set the desired number of retries and retry interval.
6. To prevent the e*Way from restarting automatically after an abnormal termination, clear the **Restart after abnormal termination** check box.
7. Click **OK**.

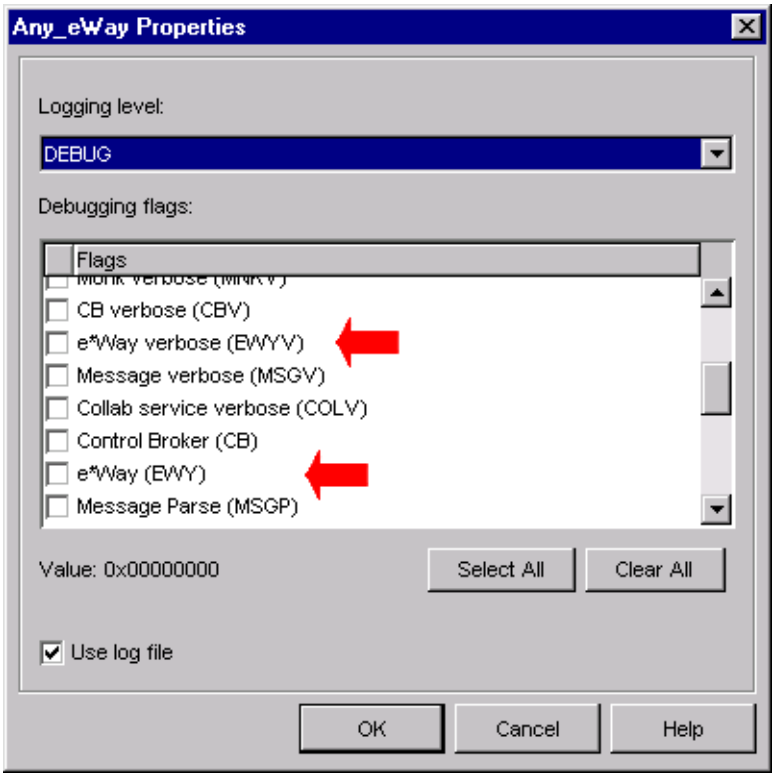
5.2.6 Activating or Modifying Logging Options

Logging options enable you to troubleshoot problems with the e*Way and other e*Gate components.

To set the e*Way debug level and flag

1. Display the e*Way's Properties dialog.
2. Select the **Advanced** tab.
3. Click **Log**. The dialog window appears (see Figure 60).

Figure 60 e*Way Properties (Advanced Tab - Log Option)



4. Select **DEBUG** for the **Logging level**.
5. Select either **e*Way (EWY)** or **e*Way Verbose (EWYV)** for the **Debugging flag.** Note that the latter has a significant negative impact on system performance.
6. Click **OK**.

The other options apply to other e*Gate components and are activated in the same manner. See the *e*Gate Integrator Alert and Log File Reference* for additional information concerning log files, logging options, logging levels, and debug flags.

5.2.7 Activating or Modifying Monitoring Thresholds

Monitoring thresholds enable you to monitor the throughput of the e*Way. When the monitoring thresholds are exceeded, the e*Way sends a Monitoring Event to the Control Broker, which routes it to the e*Gate Schema Manager and any other configured destinations.

1. Display the e*Way's properties dialog.
2. Select the **Advanced** tab.
3. Click **Thresholds**.
4. Select the desired threshold options and click **OK**.

See the *e*Gate Integrator Alert and Log File Reference* for more information concerning threshold monitoring, routing specific notifications to specific recipients, or for general information about e*Gate's monitoring and notification system.

5.3 Troubleshooting the e*Way

In the initial stages of developing your e*Gate Integrator system administration system, most problems with e*Ways can be traced to configuration.

5.3.1 Configuration Problems

In the Schema Designer

- Does the e*Way have the correct Collaborations assigned?
- Do those Collaborations use the correct Collaboration Services?
- Is the logic correct within any Collaboration Rules script employed by this e*Way's Collaborations?
- Do those Collaborations subscribe to and publish Events appropriately?
- Are all the components that *feed* this e*Way properly configured, and are they sending the appropriate Events correctly?
- Are all the components that this e*Way *feeds* properly configured, and are they subscribing to the appropriate Events correctly?

In the e*Way Editor

- Check that all configuration options are set appropriately.
- Check that all settings you changed are set correctly.
- Check all required changes to ensure they have not been overlooked.
- Check the defaults to ensure they are acceptable for your installation.

On the e*Way's Participating Host

- Check that the Participating Host is operating properly, and that it has sufficient disk space to hold the IQ data that this e*Way's Collaborations publish.
- Check that your *path* environment variable includes the location of the XXX dynamically-loaded libraries. The name of this variable on the different operating systems is:
  - PATH (Windows)
  - LD_LIBRARY_PATH (Solaris)
  - LIBPATH (AIX)
  - SHLIB_PATH (HP-UX)

In the SAP Application

- Check that the application is configured correctly, is operating properly, and is sending or receiving the correct data appropriately.

5.3.2 System-related Problems

- Check that the connection between the external application and the e*Way is functioning appropriately.
- Once the e*Way is up and running properly, operational problems can be due to:
  - External influences (network or other connectivity problems).
  - Problems in the operating environment (low disk space or system errors)
  - Problems or changes in the data the e*Way is processing.
  - Corrections required to Collaboration Rules scripts that become evident in the course of normal operations.

One of the most important tools in the troubleshooter's arsenal is the e*Way log file. See the *e*Gate Integrator Alert and Log File Reference Guide* for an extensive explanation of log files, debugging options, and using the e*Gate Schema Manager system to monitor operations and performance.

Chapter 6

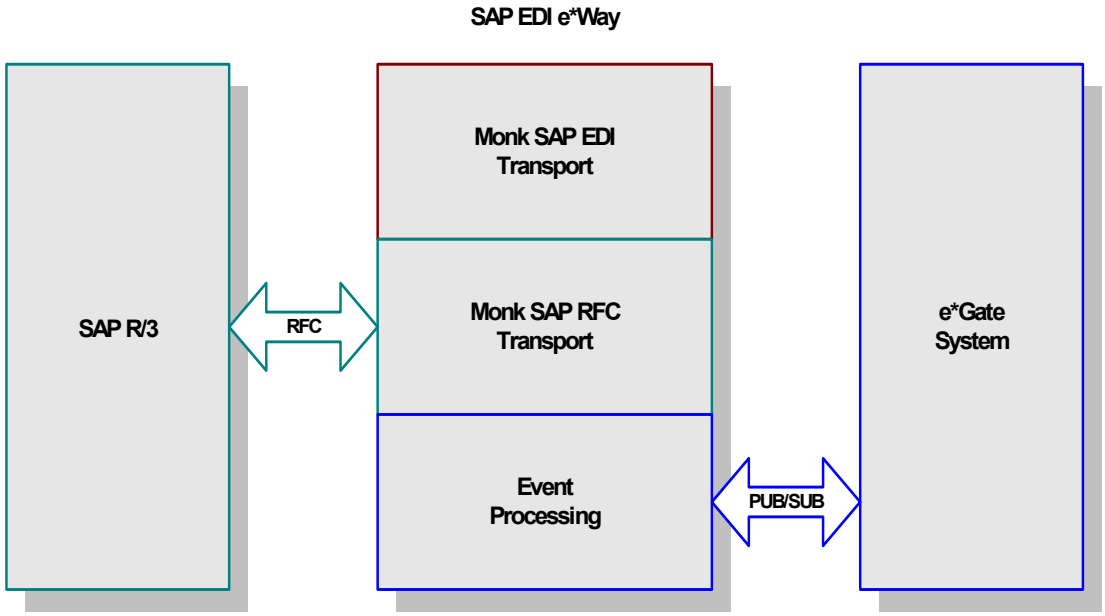
Operational Overview

6.1 e*Way Architecture

Conceptually, the SAP EDI e*Way can be viewed as a multi-layered structure (see Figure 61), consisting of:

- Monk SAP EDI Transport layer, which manages EDI communication logic
- Monk SAP RFC Transport layer, which manages RFC communications with the SAP R/3 system
- Event Processing layer, which manages the processing of data and subscribing or publishing to other e*Gate components

Figure 61 SAP EDI e*Way Architecture



Each layer contains Monk scripts and/or functions, and makes use of lower-level Monk functions residing in the layer beneath. You, as user, primarily use the highest-level functions, which reside in the upper layers.

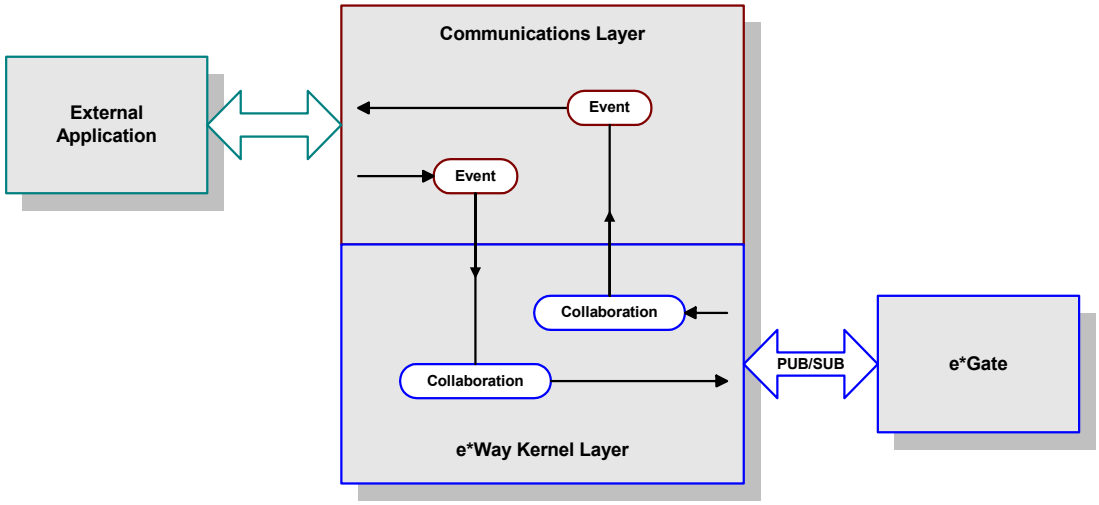
6.1.1 RFC Communications

Messages can be sent to the SAP R/3 host via Transactional RFC (tRFC) or regular RFC. With tRFC, the receiving SAP system relies on an unique Transactional ID (TID) sent with the message to ascertain whether or not a transaction has ever been processed by it before. Rather than comparing every message sent out by this e*Way against a database to determine if it has been processed previously, however, the SAP EDI e*Way assumes that all messages handled are new and assigns a new TID to each message (the counter is persistently stored by the e*Way).

6.1.2 Events and Collaborations

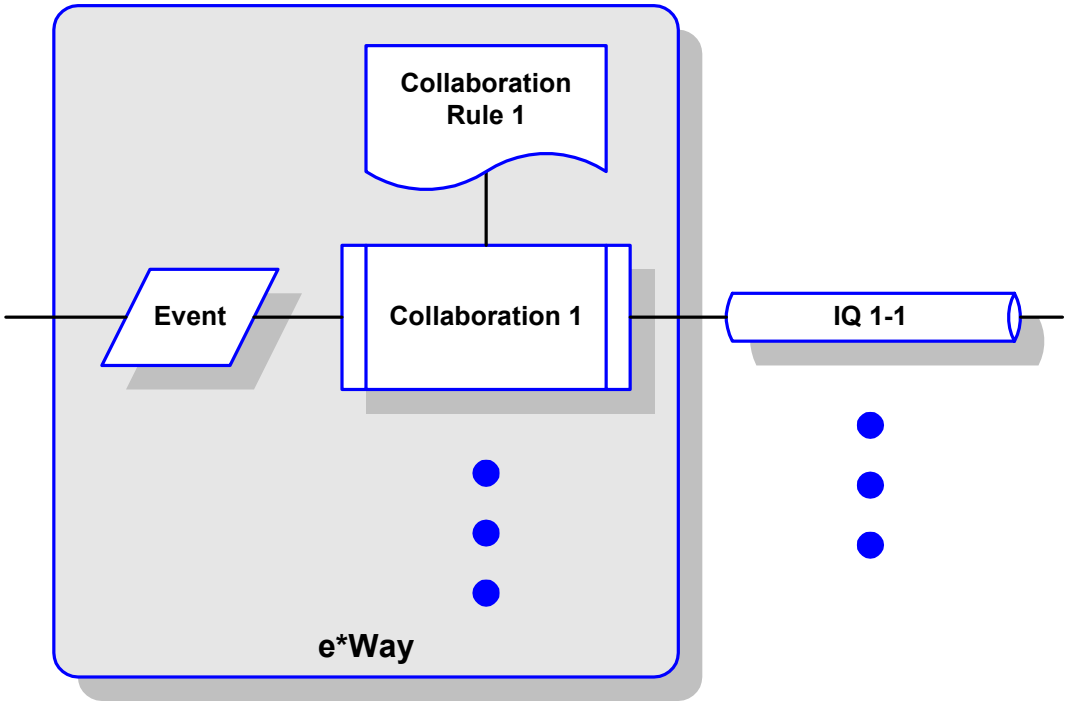
The upper layers of the e*Way use Monk functions to start and stop scheduled operations, exchange data with the external system, package data as e*Gate "Events," send those Events to Collaborations, and manage the connection between the e*Way and the external system (see Figure 62).

Figure 62 Basic e*Way Operations



Collaborations execute the business logic that enable the e*Way to do its intended work. In turn, each Collaboration executes a Collaboration Rules script, containing the actual instructions to execute the business logic. Each e*Way requires at least one Collaboration (more, if necessary to perform the required tasks), and each Collaboration requires one or more IQs (or Java e*Way Connections) to which its processed Events are published (see Figure 63).

Figure 63 Collaborations and IQs



The communication layers of the e*Way are single-threaded. Functions run serially, and only one function can be executed at a time. Processing layers are multi-threaded, with one executable thread for each Collaboration. Each thread maintains its own Monk environment; therefore, information such as variables, functions, path information, and so on cannot be shared between threads.

Configuration options that control the Monk environment and define the Monk functions used to perform various e*Way operations are discussed in **Chapter 7**. You can create and modify these functions using the SeeBeyond Collaboration Rules Editor or a text editor (such as *Microsoft Word* or *Notepad*, or UNIX *vi*).

Note: *The Collaboration Rules Editor currently is limited to reading files containing less than 6500 (collapsed) rules. For file sizes larger than this, or if a Collaboration causes abnormal termination of the Collaboration Rules Editor, an external text editor should be used.*

For more information on defining Collaborations, defining IQs, assigning Collaborations to e*Ways, or configuring Collaborations to publish Events, see the e*Gate Schema Designer's online Help system.

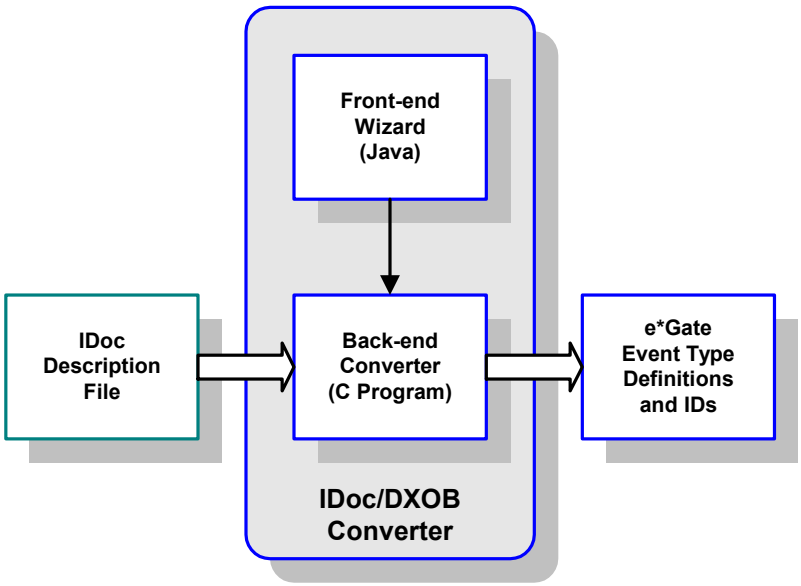
6.2 The IDoc/DXOB Converter

The SAP EDI e*Way includes a front-end SAP Converter Wizard, integrated with the e*Gate Event Type Definition (ETD) Editor Build Library Converter. The Converter parses the SAP IDoc Description Files and builds the corresponding Event Type Definitions and IDs automatically.

The SAP Converter Wizard GUI is implemented in Java, and calls an ANSI-C program that performs the actual conversion between IDoc and e*Gate Event Type Definitions and ID files.

When invoked by the **ETD Editor** > **Build Library** > **Converter** menu selection, the Java application is supplied with the specified Event Type Definition file path name and the appropriate environment variables.

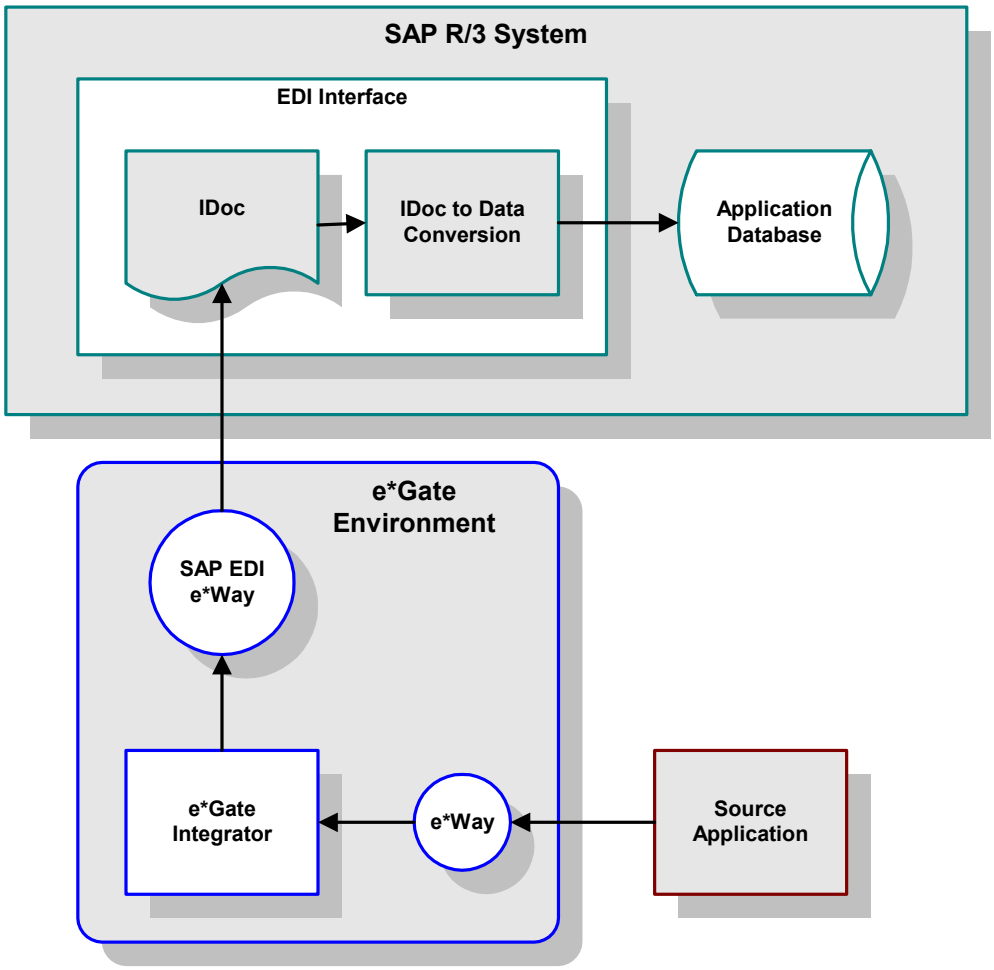
Figure 64 IDoc/DXOB Converter



6.3 Data Flow and Event Processing

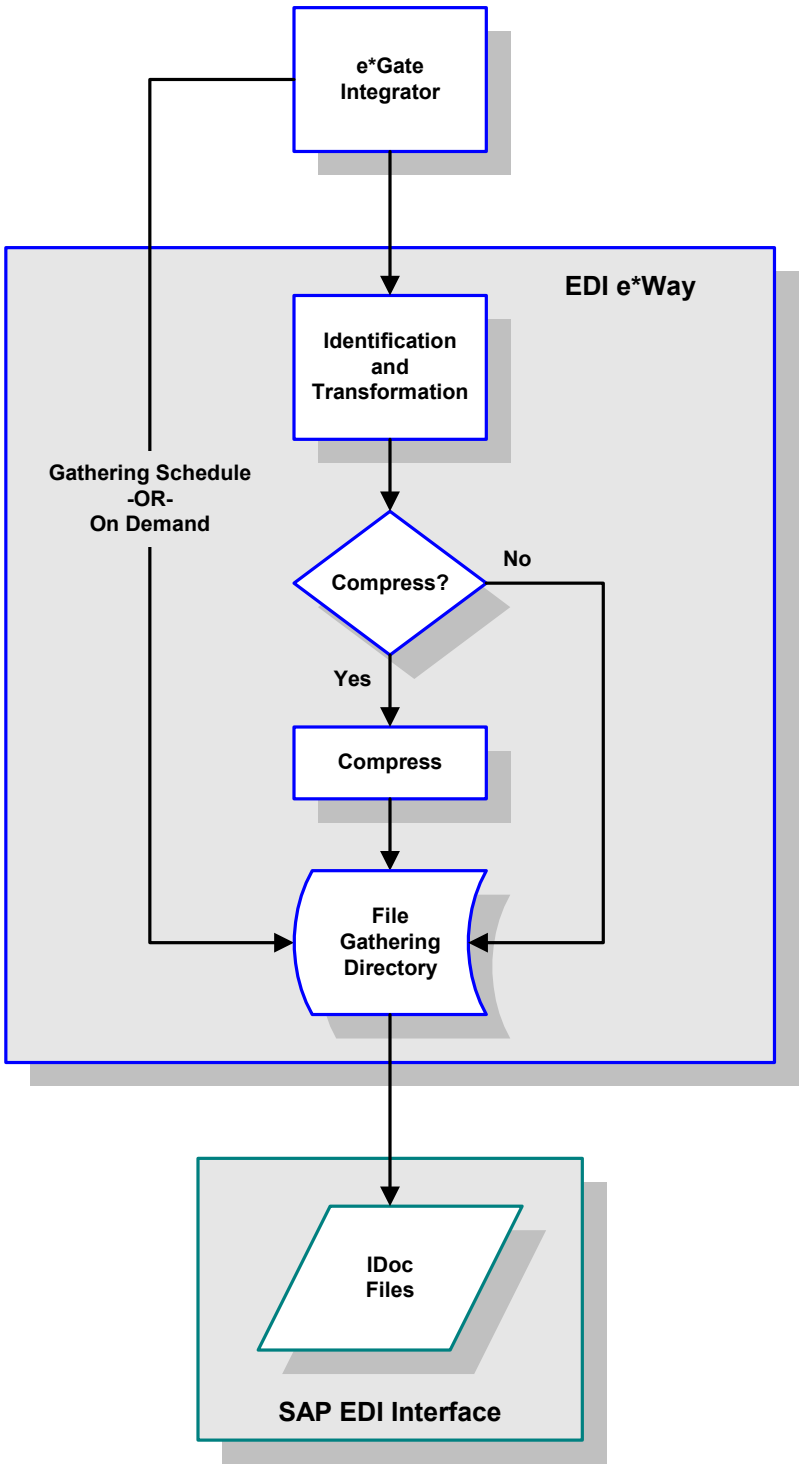
6.3.1 e*Gate to SAP

Figure 65 e*Gate-to-SAP Data Flow



In this mode of operation, the SAP EDI e*Way converts data into an IDoc, which is passed to the SAP EDI Interface. The EDI Interface, in turn, translates it into SAP internal-format data which is passed to the appropriate SAP application.

Figure 66 e*Gate-to-SAP Event Processing (1)



An Event sent from e*Gate to the SAP EDI e*Way is processed as follows:

1 Each Event from e*Gate is tested against a set of Event ID/Collaboration pairs, and the first successful ID match runs the corresponding transformation to IDoc format.

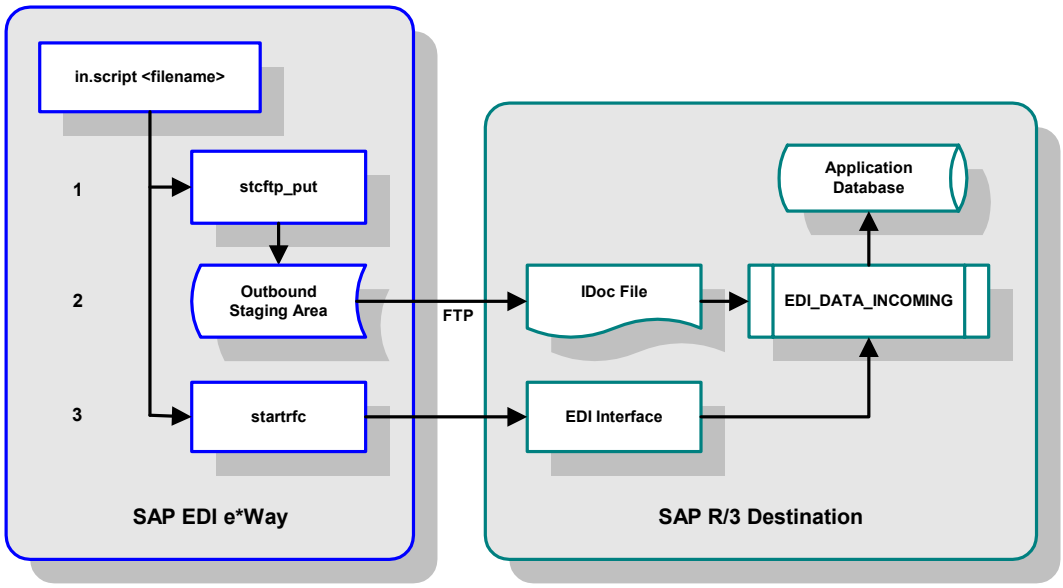
2 The resulting IDoc is compressed if necessary and placed in a temporary file the e*Way's File Gathering Directory.

Note: ***Compression*** *involves truncating the IDoc Data Record Segment and terminating it with a* ***cr-nl*** *character pair immediately following the last defined field of the segment.*

3 Contents of the temporary file are sent to the SAP system either according to a gathering schedule, or on demand. The details of this operation are described next.

Note: *If desired, the beginning and ending of the file can be defined by using* **SOF|MONK_FileBracket** *and* **EOF|MONK_FileBracket**. *An end-of-file indicator (***EOF***) indicates to the recipient that the file can be closed.*

Figure 67 e*Gate-to-SAP Event Processing (2)



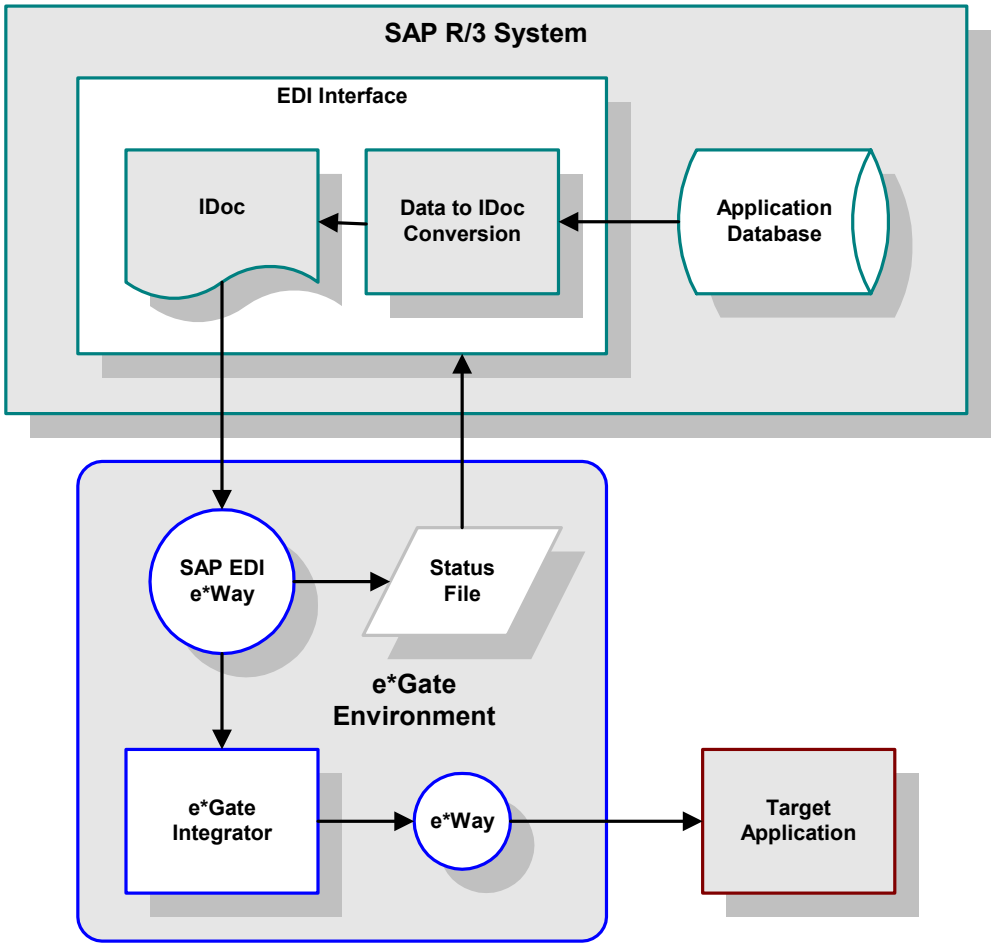
1 When the temporary file is gathered up, the shell script **in.script** is executed on the file name, which:

- triggers the utility **stcftp_put**, sending the designated file via FTP to the SAP EDI Interface
- sends to the SAP EDI Interface a **startrfc** command

2 In response to **startrfc**, the SAP EDI Interface launches the ABAP/4 program **EDI_DATA_INCOMING** to process the IDoc file.

6.3.2 SAP to e*Gate

Figure 68 SAP-to-e*Gate Data Flow



In this mode of operation, the SAP EDI Interface converts data into an IDoc, which is stored as a file in its *Outbound Directory*. Two data retrieval methods are available for obtaining an IDoc from the SAP system:

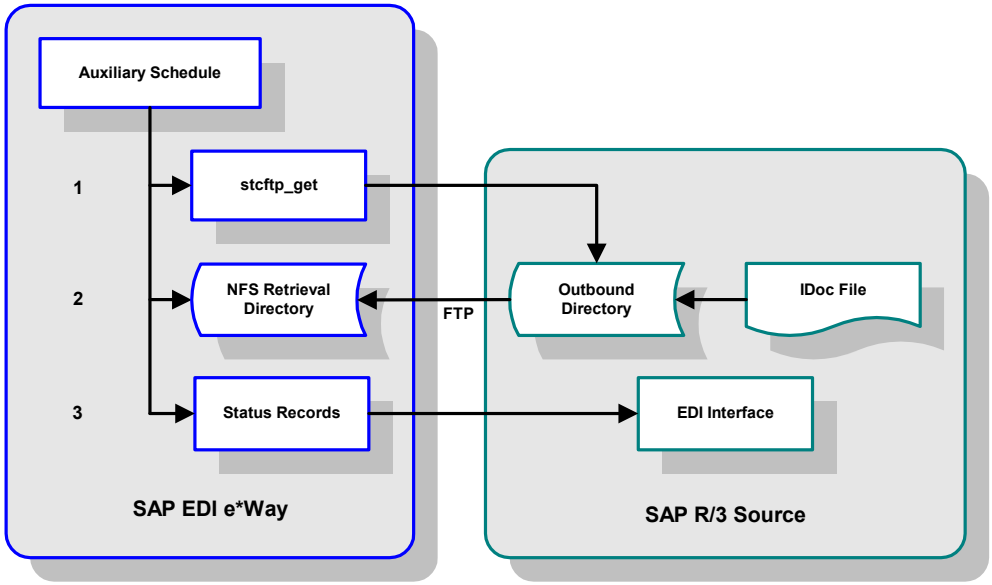
- **Active Polling**, where the e*Way initiates the file transport
- **Passive Polling**, where the SAP system initiates the file transport

Following retrieval of the file, the e*Way transforms the IDoc(s) into data which it then sends to e*Gate. The status of the Collaboration process is monitored by the e*Way and reported to the SAP system as a Status Record.

Active Polling Sequence

This method makes use of an *Auxiliary Schedule*, which is set up within the e*Way to periodically check the SAP host's *Outbound Directory* for new IDoc files. This schedule can be configured as a specific time-of-day or as a repeating time period (e.g., every 10 minutes).

Figure 69 Event Retrieval: Active Polling

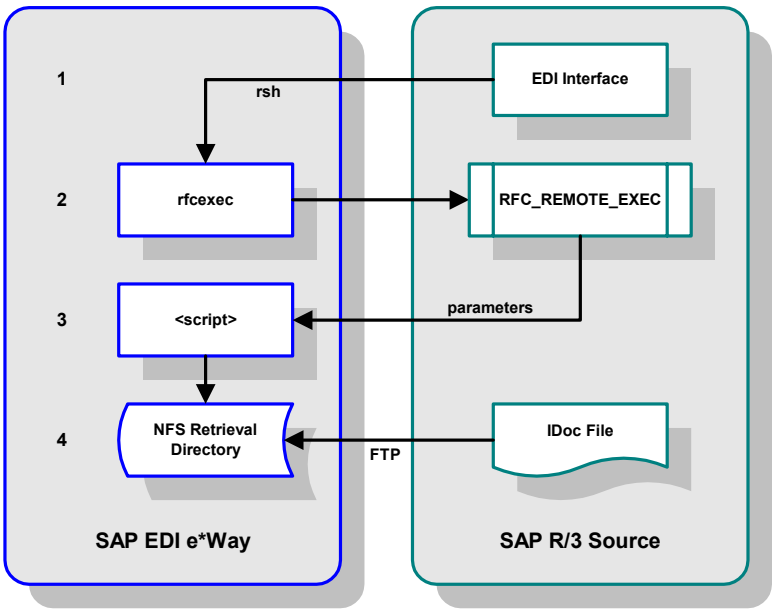


1 The *Auxiliary Schedule* periodically causes the e*Way to invoke the utility **stcftp_get** to query the SAP *Outbound Directory* for the existence of an IDoc file (see the *SAP EDI e*Way User's Guide* for a full description of **stcftp_get**).

2 If an unprocessed IDoc file is found, **stcftp_get** retrieves the file from the SAP *Outbound Directory* and stores it in the e*Way's *NFS Retrieval Directory*.

Passive Polling Sequence

Figure 70 Event Retrieval: Passive Polling



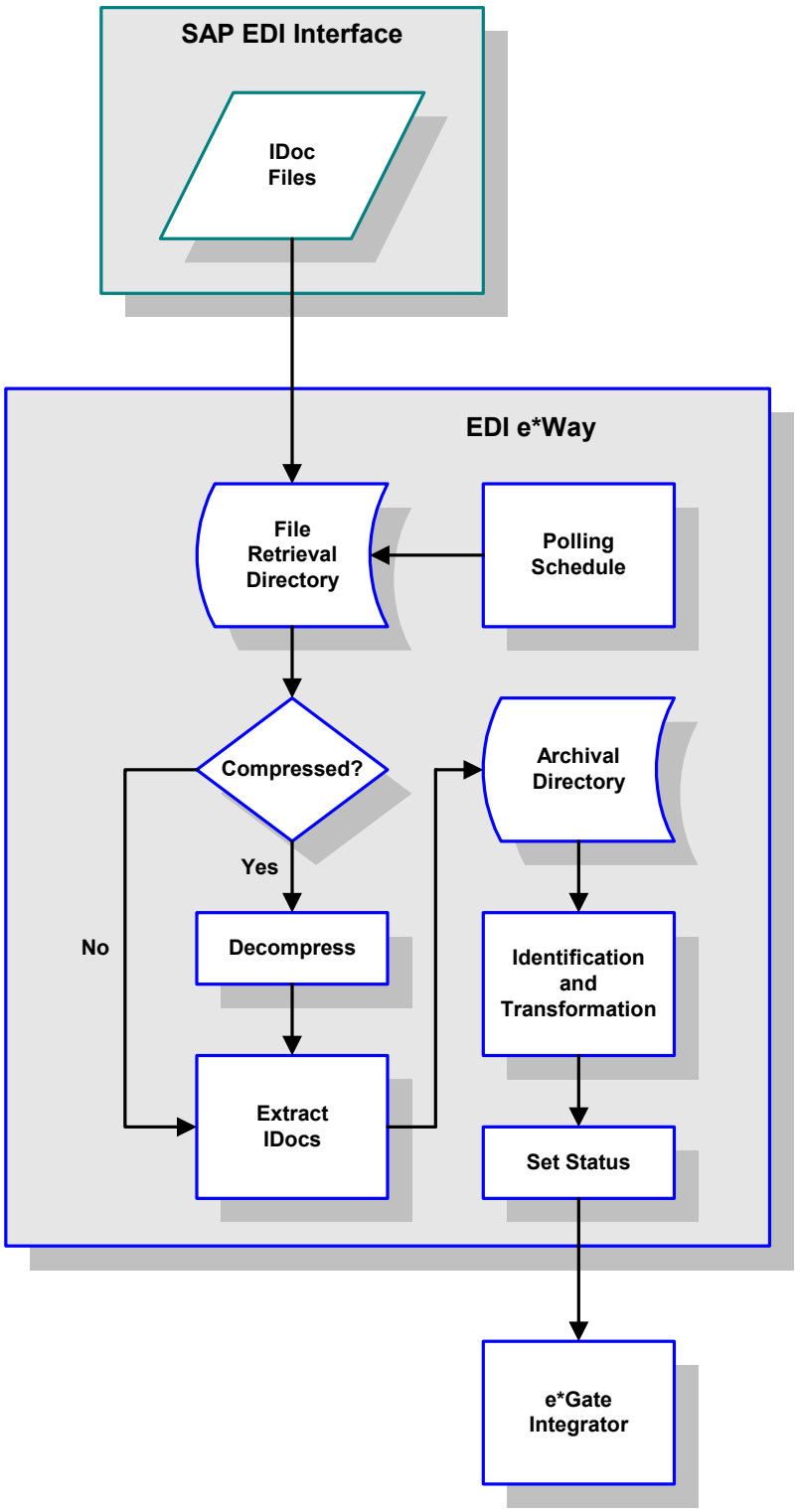
1. The SAP EDI Interface converts data from an SAP application to an IDoc, which it stores in a specific location in the EDI Interface (the location is configured via the transaction **WEDI** in the SAP GUI).
2. The SAP EDI Interface then sends an **rsh** command to the e*Way host, invoking the executable **rfcexec**.

Note: *Using the remote shell,* **rsh**, *requires that the SAP and e*Way hosts share a trusted relationship.*

3. The executable **rfcexec** then registers itself on the SAP host as an ABAP/4 program, **RFC_REMOTE_EXEC**.
4. The SAP host sends the fully-qualified name of the IDoc file and the name of a script file to be run on the e*Way host.
5. The e*Way host runs the script, which
   - retrieves the specified file from the SAP EDI Interface via FTP
   - stores the file in the *NFS Retrieval Directory* for subsequent processing by the e*Way
   - removes the file from the SAP host

Event Processing

Figure 71 SAP-to-e*Gate Event Processing



1. A polling schedule is set up within the e*Way to periodically check the *NFS Retrieval Directory* for new IDoc files. As with the *Auxiliary Schedule*, this schedule can be configured as a specific time-of-day or as a repeating time period (e.g., every 10 minutes).
2. Any IDoc file found in the *NFS Retrieval Directory* is moved to a configurable *Archive Directory*. As each file is read, the individual IDocs contained within are extracted, one at a time, and their data records decompressed (if necessary).

Note: *Since the Monk engine expects fixed-length data record segments, any* **cr** *or* **nl** *terminators used to truncate trailing blank fields are replaced with blanks to restore each record segment back to its full length, so that Collaborations function properly. This is referred to as* ***decompression*** *of the IDoc Data Record.*

3. Each IDoc extracted from the archive file is tested against a set of Event ID/ Collaboration Rule pairs, and the first successful ID match runs the corresponding Collaboration. If no match is found, a Status Record is generated to notify the SAP system of that fact.
4. During processing of the IDoc records, the e*Way creates *IDoc Status Records*, which are collected in the *Status Record File* for subsequent transmittal to the SAP host. These Status Records provide information regarding the processing status of the IDoc. Any number of Status Records may be generated for a single IDoc.
5. After the e*Way's Monk engine has performed the required transformations, it can, for example:
   - set the **MONK_X_Status** Monk variable to **OK**, and send the transformed Event to e*Gate immediately
   - set the **MONK_X_Status** Monk variable to **HOLD**, and gather them up for subsequent transmission
6. Eventually, the transformed Events are sent to e*Gate.

Chapter 7

Configuration Parameters

This chapter describes the configuration parameters for the SeeBeyond e*Way Intelligent Adapter for SAP (EDI).

7.1 Overview

The e*Way's configuration parameters are set using the e*Way Editor; see **Configuring the e*Way** on page 66 for procedural information. The SAP EDI e*Way's configuration parameters are organized into the following sections. The default configurations are provided in **stcewsapediin.def** and **stcewsapediout.def.**

7.2 General Settings

Data Direction

Description

Specifies the direction of Event or data flow, either to or from this e*Way.

Required Values

One of the following:

- **Inbound To e*Gate**

  When configured as **Inbound To e*Gate**, this e*Way receives Events from the SAP system and forwards them to e*Gate.

- **Outbound From e*Gate**

  When configured as **Outbound From e*Gate**, this e*Way receives Events from e*Gate and forwards them to the SAP system.

Interface Name

Description

Specifies a more verbose name to the **status** commands that originate from **stccmd.exe** or the e*Gate Schema Manager.

Required Values

A string; the defaults are **SAP_EDI_Inbound_Ewy** or **SAP_EDI_Outbound_Ewy**.

Additional Information

Any imbedded spaces present in the value are converted automatically to underscores (_).

Internal Buffer Initial Size

Description

Specifies an amount of memory to allocate for incoming or outgoing Events. The parameter helps the e*Way set an initial size as well as for subsequent increments.

Required Values

A number between **1024** and **65536**, inclusive. The default is **16384**.

The units are selectable as **Bytes**, **Kilobytes**, or **Megabytes**. The default is **Bytes**.

Additional Information

If unsure as to what value to enter, then use the default.

Temporary Working Directory

Description

Specifies the directory in which the temporary files are stored. Although these files are temporary in nature, they should not be summarily deleted without full knowledge of their purpose.

Required Values

A valid directory. Either an absolute path or a directory in the e*Gate client directory.

Additional Information

Any imbedded spaces present in the value are converted automatically to underscores (_).

7.3 EDI Setup

This section sets up parameters pertinent to accessing the EDI data on the SAP R/3 host.

SAP System Type

Description

Specifies the SAP System type.

Note: *Only SAP R/3 (3 Tier) Systems are currently supported by this e*Way.*

Required Values

A string; the default is **R/3.**

SAP IDoc Default Release

Description

Specifies the SAP R/3 Release.

Required Values

A string; the default is **4.0B**.

Additional Information

Select the default IDoc Release for Inbound (to e*Gate) or Outbound (from e*Gate) IDocs, unless the release is specified in the Control Record. Do not use periods (or dots) if entering a Release that's not on the displayed list. Only Release 3.0C and higher are supported.

Data File Layout

Description

Specifies the data layout of the records within the file.

Required Values

IDoc

Additional Information

Only SAP R/3 EDI IDoc records are currently supported. These can be compressed (i.e., trailing blanks within the Data Segments, **EDI_DD**, can be truncated with **Carriage Return** and/or **Newline**).

Data File Ready Sensor (SAP to e*Gate Only)

Description

Enables the detection of the presence of an EDI Data File on the SAP R/3 System for processing by this e*Way.

Required Values

File Polling

This e*Way polls an NFS directory (can be local or remote) for the presence of certain EDI Data files, according to a preset schedule.

Data File Compression Method (e*Gate to SAP Only)

Description

Defines the compression method to use upon the data within the gathered data file.

Required Values

One of the following:

- **IDoc Data Record Truncation**

  Trailing blanks up to the defined length of each IDoc Data Record are truncated and terminated with an End-of-Line terminator (**Newline** for Unix).

- **None**

  Data not compressed.

File Polling Schedule (SAP to e*Gate Only)

Description

Specifies a schedule of when to poll a designated area for EDI Data Files deposited by the SAP R/3 Host for processing by this e*Way.

Required Values

A time interval in **Seconds**; the default is **30**.

Auxiliary Polling Script (SAP to e*Gate Only)

Description

Specifies the path name to a script that periodically retrieves (e.g., via FTP) files from a remote server into the local file system, where they can be accessed by this e*Way. No Auxiliary Polling is performed if this parameter is not supplied.

Required Values

A file name, with an optional absolute path.

If you provide a file name that is not an absolute path, then the value of **SystemData** from **.egate.store** is prefixed automatically to the value of the parameter. The default is **bin/get_idocs_from_r3.bat**.

This script receives no parameters, but has environmental variables defined as:

EGSYSTYEM	e*Gate System Data Directory, (e.g., /home/eGate/)
FTUSER	file Transport User Name (if defined below)
FTPASSWORD	decrypted File Transport User Password
SAPUSER	SAP R/3 User Name (if defined below)
SAPPASSWORD	decrypted SAP R/3 User Password

Typically, for SAP EDI applications, the SAP R/3 host **rsh** (remote shell) executes on an EDI subsystem host (which this e*Way runs on) an **RFCEXEC** program, which it then communicates with (via RFC) and executes a script on the subsystem host to retrieve a certain EDI file and trigger its processing.

On operating systems which do not provide **rsh** server services, however, this methodology is not viable. For these operating systems, the Auxiliary Polling Script provides a proactive way to retrieve the EDI Data files from the SAP R/3 host when they are available.

Additional Information

Use **stcftp_get.exe** for file retrieval from a remote R/3 system:

```
stcftp_get.exe -h<host name> [-t<host type>] [-l<local host>]
[-n<net timeout (s)>] [-s<svr timeout (s)>]
[-f<file size stable timeout (s)>] [-g|-G (turn ON debugging)]
[-b (binary transfer mode)] -u<user name> -p<user password>
[-d{<Shared Data directory> | <FTP Heuristics File>}]
[-a<archive to remote directory> | -r (remove remote file)]
{<remote filename> <local {filename |dirname}>|
 <remote direname> <local dirname>}
```

Note: *The dirname specified must end with a directory delimiter native to the respective operating system. Available host types are:*
UNIX
HCLFTPD 5.1
HCLFTPD 6.0.1.3
VMS
MSFTPD 2.0
MVS

Debug logging appears in **stcftp_get.log** where:

-g	overwrites
-G	appends
-ver	shows version
-h	this screen

Auxiliary Polling Schedule (SAP to e*Gate Only)

Description

Specifies the schedule of when to run an Auxiliary Polling Script.

Required Values

A time interval in **Seconds**. The default is **20**.

This parameter must be set if a value is entered for the Auxiliary Polling Script.

See also:

Auxiliary Polling Script (SAP to e*Gate Only) on page 92

File Transport User Name

Description

Specifies the user name to allow access for the transport mechanism used, such as FTP, when the Auxiliary Polling Script (SAP to e*Gate) or the optional Notification Shell Script (e*Gate to SAP) is used.

Value

A string.

Additional Information

This parameter must be entered *before* the **File Transport User Password**, in order for the password encryption to function properly.

When the Auxiliary Polling Script, Status Record Trigger Script, or **Notification Shell Script** is run, the specified value is available through the environmental variable **FTUSER**.

See also:

Auxiliary Polling Script (SAP to e*Gate Only) on page 92

Status Record Trigger Script (SAP to e*Gate Only) on page 117

File Transport User Password

Description

Specifies the corresponding password for the user specified in the **File Transport User Name**.

Value

A string.

Additional Information

This parameter must be entered *after* the File Transport User Name.

When the Auxiliary Polling Script, Status Record Trigger Script, or **Notification Shell Script** is run, the specified value is available through the environmental variable **FTPASSWORD**.

See also:

Auxiliary Polling Script (SAP to e*Gate Only) on page 92

Status Record Trigger Script (SAP to e*Gate Only) on page 117

SAP User Name

Description

Specifies the user name to allow access to the SAP R/3 system.

Value

A string.

Additional Information

This parameter must be entered *before* the **SAP User Password**, in order for the password encryption to function properly.

When the Status Record Trigger Script or Notification of Shell Script is run, the specified value is available through the environmental variable **SAPUSER**.

SAP User Password

Description

Specifies the corresponding password for the user specified.

Value

A string.

7.4 NFS File Polling Setup (SAP to e*Gate Only)

This section sets up parameters pertinent to NFS File Polling for input data files from the SAP R/3 host.

NFS Retrieval Directory

Description

Specifies the NFS directory accessible to the local system from which files are to be retrieved by this e*Way.

Required Values

A valid directory name, with an optional absolute path.

If you provide a directory that is not an absolute path, then the value of **SystemData** from **.egate.store** is prefixed to the value of the parameter automatically.

Additional Information

If NFS is used and is "soft" mounted (file operations on the directory are not blocked if the NFS server is unavailable), you must specify a Retrieval Directory that is *not* the mount point of the file system but rather a subdirectory of the mount point; otherwise when the NFS server is unavailable, data is retrieved from the local file system instead.

If NFS is "hard" mounted (file operations on the directory are blocked until the NFS server is available), the e*Way *can* be at times non-responsive to e*Gate when it's blocked in a file operation. If this is not desirable, use "soft" mounted NFS file systems.

Files Retrieval Filter Type

Description

Specifies the Filter Type to be used in retrieving files, during automatic file retrieval.

Required Values

One of the following:

- **Regular Expression (RE)**
- **Constant**
- **Counter**
- **Chronometer**

See **File Filter Pattern** for additional information.

File Filter Pattern

Description

Specifies the appropriate Filter Pattern for the **Files Retrieval Filter Type**.

Required Values

A file name-like expression as exemplified in the following table.

Filter Type	Filter Patterns	
Regular Expression (RE)	Some basic regular expression (RE) syntax:	
	^	Beginning of field
	$	End of field
	[ ]	A character within ASCII range
	[^]	A character not within ASCII range
	.	Any character
	*	Zero or more occurrences
	+	One or more occurrences
	?	Zero or one occurrence
	{ }	Exact number of occurrences
	{m,}	Minimum number of occurrences
	{,n}	Maximum number of occurrences
	{m,n}	Min Max number of occurrences
	|	Or
	\	The escaped RE character
	[[:space:]]	Any white space character
	[[:alnum:]]	Any alphanumeric character
	[[:digit:]]	Any digital character
	[[:punct:]]	Any punctuation character
	[[:lower:]]	Any lowercase character
	[[:upper:]]	Any uppercase character
	[[:xdigit:]]	Any hexadecimal digit
	For example, to specify files beginning with ORD, followed by 5 digits, and ending with a DAT extension, enter: **^ORD[0-9]{5}\.DAT**.	
Constant	Only files with the exact name specified, for example **ORDERS.DAT**.	
Counter	C printf()-like format specifier for numeric display, such as: **ORD%05d.DAT** (e.g., for ORD00051.DAT) **ORD%d.DAT** (e.g., for ORD1.DAT or ORD123.DAT)	

<table>
<tr><th>Filter Type</th><th colspan="2">Filter Patterns</th></tr>
<tr><td rowspan="3">Chronometer</td><td colspan="2">C strftime()-like format specifier for epoch display such as: ORD%d%b%Y.DAT (e.g., for ORD30Apr1998.DAT).
The only supported specifiers are:</td></tr>
<tr><td>%H
%M
%S
%d
%b
%m
%y
%Y</td><td>hour: 00 - 24
min: 00 - 59
sec: 00 - 59
day: 01 - 31
month: Jan - Dec
month: 01 - 12
yy: 70 - 36
ccyy: 1970 - 2036</td></tr>
<tr><td colspan="2">Note: The current date and time are used for the format specifiers. You must provide enough information to specify the UTC time, i.e., sec, min, hour, day, month, year. Any component missing from the sequence is defaulted to the corresponding current time component.</td></tr>
</table>

Counter Start Mode

Description

Determines the starting counter mode. This parameter needs to be defined *only* for the **Counter** filter type (see **Files Retrieval Filter Type**), and is ignored by all other filter types.

Required Values

One of the following:

- **Resume**

  Indicates that the counter should resume from its previous value. If there is no previous value, it starts from its **Counter Initial Value.**

- **Initialize**

  The counter resets and uses its **Counter Initial Value**.

Counter Initial Value

Description

Determines the initial value for a numerical counter. This parameter needs to be defined only for the **Counter** filter type (see **Files Retrieval Filter Type**), and is ignored by all other filter types.

Required Values

An integer between **1** and **2147483647**, inclusive. The default is **1**.

Additional Information

Only **Counter** type format filenames with counter values greater than or equal to this value are used in the file transfer to and from the remote system.

Counter Maximum Width

Description

Determines the maximum width value (rollover value) for the counter. This parameter needs to be defined only for **Counter** filter type (see **Files Retrieval Filter Type**), and is ignored by all other filter types.

Required Values

An integer between **1** and **10**, inclusive. The default is **10**.

Additional Information

The Counter Maximum Width determines the rollover value for the counter, that is:

Counter Width	Rollover Value
1	9
2	99
3	999
4	9999
5	99999
6	999999
7	9999999
8	99999999
9	999999999
10	2147483647

Any attempt to increment the counter beyond the rollover value resets it to its initial value.

Chronometer Start Mode

Description

Determines the starting time value. This parameter *only* needs to be defined for the **Chronometer** filter type (see **Files Retrieval Filter Type**), and is ignored by all other filter types.

Required Values

One of the following:

- **Resume**

The chronometer resumes from its previous value. If there's no previous value, it starts from its **Chronometer Initial Value - Time**.

- **Initialize (Reset)**

  The chronometer resets and uses its **Chronometer Initial Value - Time**.

- **Current Time**

  The chronometer resets to the current local time.

Chronometer Initial Value - Date

Description

Determines the initial value for a **Chronometer** filter Date. This parameter needs to be defined only for the **Chronometer** filter type, and is ignored by all other filter types. Only filenames with Date values greater than or equal to this value is used in the file transfer to and from the remote system.

Required Values

A date within the range **JAN 01, 1970 - FEB 5, 2036**, inclusive.

Chronometer Initial Value - Time

Description

Determines the initial value for a **Chronometer** filter Time (local). This parameter needs to be defined only for the **Chronometer** filter type, and is ignored by all other file types. Only filenames with Time values greater than or equal to this value is used in the file transfer to and from the remote system.

Required Values

A time within the range **00:00:00 - 23:59:59**, inclusive.

Input File Start Position Mode

Description

Determines the offset position within a data file from which to begin reading, if processing was previously interrupted.

Required Values

One of the following:

- **Resume**

  Indicates the e*Way should resume reading an input data file from wherever it previously left off.

- **Initialize (Reset)**

Indicates the e*Way should start reading from the beginning of the current file to be processed.

Additional Information

Generally, the setting should be **Resume** (even for new files to be processed; where the file beginning is used), but if after an interruption and the entire file needs to be reprocessed, then **Initialize** (**Reset**) can be chosen.

NFS Access Verification Interval

Description

Determines the timeout interval between consecutive verification of the NFS Retrieval Directory to test for accessibility.

Required Values

A number between **1** and **2147483647**, inclusive. The default is **15**.

Units are specified in **Seconds** or **Minutes**. The default unit is **Seconds**.

File Size Stable Timeout

Description

Determines the timeout interval between consecutive file size checks to determine stability.

Required Values

A number between **1** and **2147483647**, inclusive. The default is **3**.

Units are specified in **Seconds** or **Minutes**. The default unit is **Seconds**.

Processed File Disposition

Description

Determines what is done with the local file after it has been successfully sent to the remote.

Required Values

One of the following:

- **Delete**

  The file is deleted.

- **Archive**

  The file is archived (or renamed) to another local directory

- **Archive and Compress**

  In addition to being archived, the local file also is compressed to save space.

File Archival Directory

Description

Determines the directory on the Local file system in which files are archived after they have been successfully sent.

Required Values

A valid directory, with an optional absolute path.

If you provide a directory that is not an absolute path, then the value of **SystemData** from **.egate.store** is prefixed to the value of the parameter automatically.

Additional Information

This parameter needs to be considered only if the **Archive** or **Archive and Compress** option is selected for the **Processed File Disposition** parameter.

File Compression Executable

Description

Specifies the fully-qualified path name to a program executable that compresses a file. Since various programs have different formats for the actual arguments, the format is defined next in the **Compression Executable Arguments Format** parameter.

Required Values

A file name, with an optional absolute path.

If you provide a file name that is not an absolute path, then the value of **SystemData** from **.egate.store** is prefixed to the value of the parameter automatically. (The defaults are **opt/gnu/bin/gzip, gzip.exe, bin/compress, pkzip.exe**.)

Additional Information

This parameter needs to be specified only if the **Archive and Compress** option is selected for Processed File Disposition.

Compression Executable Arguments Format

Description

Defines the format string representing all the arguments necessary to compress a file, whose name is designated by the **%s** token, to the same directory as the original.

Required Values

A string corresponding to the **File Compression Executable**, as shown in the following table:

Compression Executable	Argument Format String
gzip	%s
compress	%s
PKZIP.EXE	%.8s %s (see additional information below)

Additional Information

For DOS version of **PKZIP.EXE**, file names have to conform with the 8.3 notation so input filenames (given by **File Filter Pattern**), from which archive filenames are derived, have to comply. The first argument to **PKZIP** is the base name of the ZIP to create (a **.ZIP** extension is automatically appended) and the second is the actual file to compress.

Start of File Notification Event

Description

Specifies whether or not another e*Way needs to be notified of the **Start-Of-File** condition.

Required Values

SOF|%s

Additional Information

Before an input file is processed, it may be necessary to notify another e*Way of the **Start-Of-File** condition. If this parameter is defined, such an Event (with the **%s** replaced by the Archived Name of the file to be processed) is sent to e*Gate to be routed to the appropriate e*Way for processing. The receiving e*Way can use this information to preserve the file bracketing of the original data.

If this parameter is defined, it's corresponding **End Of File Notification Event** parameter also must be defined.

End Of File Notification Event

Description

Specifies whether or not another e*Way needs to be notified of the **End-Of-File** condition.

Required Values

EOF|%s

Additional Information

After an input file is processed, it is sometimes necessary to notify another e*Way which has been receiving the Events from this e*Way of the **End-Of-File** condition. If this parameter is defined, such as Event (with <**%s**> replaced by the Archived Name of

the processed file) is sent to e*Gate to be routed to the appropriate e*Way for processing. The receiving e*Way can use this information to preserve the file bracketing of the original data.

If this parameter is defined, it's corresponding **Start of File Notification Event** parameter also must be defined.

Rejected Event Journal

Description

Specifies a file name to which the system sends

- all Events sent in by the external system that could not be identified or translated successfully by this e*Way
- all Events translated by this e*Way which were rejected by e*Gate

Required Values

A file name, with an optional absolute path.

If you provide a file name that is *not* an absolute path, then the value of **SystemData** from **.egate.store** is prefixed to the value of the parameter automatically.

Additional Information

Journaling is enabled if this configuration parameter contains a value. Leaving this parameter blank disables the journaling in the e*Way.

The Events are appended to the Journal file (if the file does not exist, it is created). This Journal can be converted into a readable flat file by using the **stcjdump.exe** utility (see **Appendix A**).

7.5 NFS File Gathering Setup (e*Gate to SAP Only)

NFS Access Verification Interval

Description

Specifies the timeout interval between consecutive verification of the NFS File Gathering Directory to test for accessibility.

Required Values

A number between **1** and **2147483647**, inclusive. The default is **15**.

Units are specified in **Seconds** or **Minutes**. The default unit is **Seconds**.

NFS File Gathering Directory

Description

Specifies the NFS directory accessible to the local system where the files are to be saved by this e*Way.

Required Values

A file name, with an optional absolute path. If you provide a file name that is *not* an absolute path, then the value of **SystemData** from **.egate.store** is prefixed to the value of the parameter.

Additional Information

If NFS is used and is "soft" mounted (file operations on the directory are not blocked if the NFS server is unavailable), you must specify a Gather Directory that is *not* the mount point of the file system but rather a subdirectory of the mount point; otherwise when the NFS server is unavailable, data is retrieved from the local file system instead.

If NFS is "hard" mounted (file operations on the directory are blocked until the NFS server is available), the e*Way *can* be at times non-responsive to e*Gate when it's blocked in a file operation. If this is not desirable, use "soft" mounted NFS file systems.

Gathered data are placed into a temporary holding file, named **Port Name.tmp**, in this same directory. When it's time to wrap-up he file, the temporary file is automatically renamed to its filtered name specified later.

File Gathering Start Mode

Description

Specifies whether or not data should continue to be gathered into the temporary holding file, after shutdown.

Required Values

One of the following:

- **Resume**

  Indicates that file gathering should resume, appending to the current contents, if any, of the temporary holding file and also, when the e*Way is shutdown, do *not* automatically wrap up the temporary holding file.

- **Initialize (Reset)**

  Indicates that file gathering should begin anew, purging first any contents of the temporary holding file and also, when the e*Way is shutdown, *automatically* wrap up the temporary holding file.

Additional Information

The gathered data is placed initially into a temporary holding file, which is eventually renamed to the filtered file name during wrap-up. If the e*Way is shutdown and the file hasn't yet been wrapped up, the temporary holding file may exist the next time the e*Way is restarted. This parameter allows the user to specify whether or not data should continue to be gathered into the temporary holding file and, at e*Way shutdown time, whether or not the temporary holding file should automatically be wrapped up.

File Gathering Filter Type

Description

Determines the filter type to use for gathering files.

Required Values

One of the following:

- **Constant**

  A constant file name is to be used, for example:

  ```
  ORDERS.DAT
  ```

- **Counter**

  Some part of the file name represents a numeric counter, for example:

  ```
  ORD00051.DAT
  ```

- **Chronometer**

  Some part of the file name represents a *current* date-time epoch, for example:

  ```
  ORD30Apr1998.DAT
  ```

- **Monk Determined**

  A Monk variable is used to determine the name of the new file.

File Filter Pattern

Description

Selects the appropriate Filter Pattern for the above chosen **File Gathering Filter Type**.

Required Values

One of the following:

- **Constant**

  Only files with the *exact name* specified, for example:

  ```
  ORDERS.DAT
  ```

- **Counter**

  C printf()-like format specifier for numeric display, for example:

  ```
  ORD%05d.DAT
  ```

  (such as for ORD00051.DAT) or

  ```
  ORD%d.DAT
  ```

  (such as for ORD1.DAT or ORD123.DAT)

- **Chronometer**

  C strftime()-like format specifier for the *current* epoch display such as:

  ```
  ORD%d%b%Y.DAT
  ```

  (such as for ORD30Apr1998.DAT). The only supported specifiers are:

%H	(hour: 00-24)
%M	(min: 00-59)
%S	(sec: 00-59)
%T	(millisec: 000-999)
%d	(day: 01-31)
%b	(mnth: 01-12)
%Y	(ccyy: 1970-2036)

  The current date and time is used for the format specifiers.

- **Monk Determined**

  A Monk variable's value is used to determine the file name format.

Counter Start Mode

Description

Determines the starting counter mode. This parameter needs to be defined only for the **Counter** filter type (see **File Gathering Filter Type**), and is ignored by all other filter types.

Required Values

One of the following:

- **Resume**

Indicates that the counter should resume from its previous value. If there is no previous value, it starts from its **Counter Initial Value**.

- **Initialize**

  The counter resets and uses its **Counter Initial Value**.

Counter Initial Value

Description

Determines the initial value for a numerical counter. This parameter needs to be defined only for the **Counter** filter type (see **File Gathering Filter Type**), and is ignored by all other filter types.

Required Values

A number between **1** and **2147483647**, inclusive. The default is **1**.

Only filenames with counter values greater than or equal to this value is used in the file transfer to and from the remote system.

File Gathering Closure Method

Description

Specifies when to stop gathering Data records and close a Data file, making the file available for processing by another system.

Required Values

One of the following:

- **Monk Determined**

  The Monk function determines when to close the data-gathering file.

- **Schedule Based**

  The data-gathering file is closed at specified periods.

- **Count Based**

  The data-gathering file is closed after a certain configurable number of Events are written.

File Gathering Schedule

Description

Specifies a schedule of when to gather all the e*Gate Events received (and possibly translated) into a file (named according to the previously defined **File Gathering Filter Type**) if the **Schedule Based** option was selected for the **File Gathering Closure Method**.

Required Values

A time interval in **Seconds**.; the default is **20**.

Number of Events Per File

Description

Specifies the number of Events to store in an NFS file when the **Count Based** option is selected was selected for the **File Gathering Closure Method**. After receiving this preset number of Events, the Events are written into an NFS file.

Required Values

A number between **1** and **2147483647**, inclusive. The default is **1**.

File Ready Notification Method

Description

Specifies if and how notification indicating that a file is now ready for use is to be sent to the receiving component.

Required Values

One of the following:

- **No Notification**

  No notification is sent

- **Return Receipt**

  A Return Receipt Event is generated and sent to e*Gate for further processing, for example:

  ```
  RR|<filename>|<basename>|<file size>
  ```

- **Shell Script Execution**

  A Shell Script is executed to send notification, such as to transfer the gathered file to some remote server via FTP.

Notification Shell Script

Description

Specifies the script to be used by this e*Way after an NFS file has been completely gathered.

Required Values

A file name, with an optional absolute path. The default is **bin/send_idocs_to _r3.bat**.

If you provide a file name that is *not* an absolute path, then the value of **SystemData** from **.egate.store** is prefixed to the value of the parameter automatically.

Additional Information

After having been completely gathered, the script is given two parameters in order:

1. Fully qualified path name of Gathered File, for example **C:/mydata/ord.dat**
2. Base name of the Gathered File (e.g., **ord.dat**) and has the following environmental variable defined:

Variable	Description
EGSYSTEM	e*Gate System Data Directory, for example **\"/home/eGate\"**
FTUSER	File Transport User Name (if defined above)
FTPASSWORD	decrypted File Transport User Password
SAPUSER	SAP R/3 User Name
SAPPASSWORD	decrypted SAP R/3 User Password

Use **stcftp_put.exe** to assist with delivery of a file to a remote R/3 system:

```
stcftp_put.exe -h<host name> [-t<host type>] [-l<local host>]
[-n<net timeout (s)>][-s<svr timeout (s)>]
[-f<file size stable timeout (s)] [-g|-G (turn ON debugging)
[-b (binary transfer mode)] -u<user name> -p<user password>
[-d{<Shared Data directory> | <FTP Heuristics File>}]
[-a<archive to local directory> | -r (remove local file)]
<local filename> <remote {filename | dirname}>
```

Note: *'dirname' specified must end with a directory delimiter native to the respective OS. Available host type: UNIX, HCLFTPD 5.1, HCLFTPD 6.0.1.3, VMS, MSFTPD 2.0, MVS.*

Shell Script Execution Mode

Description

Specifies whether the **Notification Shell Script** should be executed in the foreground or in the background. This parameter is used only if **Notify via Shell Script** is selected as the **Notification Style**.

Required Values

One of the following:

- **Background**

  The script is executed in a separate program thread, allowing the immediate processing of another inbound IDoc file.

- **Foreground**

  The e*Way ensures that the Trigger Script is executed completely before processing another inbound IDoc file.

Rejected Event Journal

Description

Specifies a file name to which the system sends

- all Events sent in by the external system that could not be identified or translated successfully by this e*Way
- all Events translated by this e*Way which were rejected by e*Gate

Required Values

A file name, with an optional absolute path.

If you provide a file name that is *not* an absolute path, then the value of **SystemData** from **.egate.store** is prefixed to the value of the parameter automatically.

Additional Information

Journaling is enabled if this configuration parameter contains a value. Leaving this parameter blank disables the journaling in the e*Way.

The Events are appended to the Journal file (if the file does not exist, it is created). This Journal can be converted into a readable flat file by using the **stcjdump.exe** utility (see **Appendix A**).

7.6 Transaction Processing

These parameters control the way that Transactions are identified, processed, and sent to e*Gate for further processing and routing.

Transaction Identification and Translation

Description

Each incoming transaction needs to be Identified for valid IDoc Structure Syntax and then (if successful), translated to an appropriate Event format.

Required Values

A file name with a *relative* path. The default is **EG_ID_ALL^<none>**.

Additional Information

Enter pairs of Event ID Collaboration and Collaboration Rule files, each name separated by a caret (^). A *relative* path name is essential, since these are keys to objects in the SeeBeyond e*Gate Registry. Also, it is assumed that the Event ID Collaboration and Collaboration Rule function names correspond to the names of their respective files.

Both the Event ID Collaboration and Collaboration Rule may require extensive mapping of data into IDoc structures. In order to improve performance, both Collaborations can be combined into a single Collaboration File and the same file name be entered for both Event ID Collaboration and Collaboration Rule.

If the Event ID Collaboration file name is *exactly* **EG_ID_ALL** (this is *not* a real file name), then *all* IDocs or Events are successfully Identified. By the same token, if the Collaboration Rule file name is *exactly* **none**, then *no* processing is performed on the IDoc. Thus to allow all IDocs to be sent to e*Gate (Inbound) or SAP (Outbound) unprocessed, type in

EG_ID_ALL^ none

If a Collaboration Rule file name is used, the Monk function must set a global Translation Status Monk Variable (whose name is configurable, see below) to one of the recognized statuses (further detailed for the Translation Status Monk Variable parameter) before returning, using a Monk statement such as:

(define MONK_X_Status "OK")

Translation Status Monk Variable

Description

Specifies the name of the global Monk Variable which determines the status of a Monk Translation

Required Values

MONK_X_Status

Additional Information

After translation is completed, the Translation function *must* set this Monk Variable with one of the following status (the first part of the value must start with it):

- **OK**

  The translation is successful and the Event is returned as a single string from the Monk function

- **OKWORKAREA**

  The translation is successful but the data is contained instead in a file given by the **Work Area File Monk Variable**

- **OKNODATA**

  This translation is successful but there is no data available at this time. This also implies that any **Status Record Data** generated is to be *ignored* at this time.

- **HOLD**

  The translation is successful but no Events resulting from the translation are to be sent to e*Gate at this time. This also implies that any **Status Record Data** generated is to be ignored at this time.

- **ERROR:**

  (Including the colon) This indicates that an error condition occurred and an SAP EDI Subsystem Archivable Status (*just* the Status code) *must* follow the colon (**:**). Some of the pertinent Archivable Statuses are:

04	Error within control information of EDI subsystem
05	Error during translation
06	Translation OK
07	Error during syntax check
08	Syntax check OK
09	Error during interchange handling
10	Interchange handling OK
11	Error during dispatch (general error during execution of Monk function)
12	Dispatch OK (This is the SUCCESSful translation code)
13	Retransmission OK
14	Interchange Acknowledgement positive
15	Interchange Acknowledgement negative
16	Functional Acknowledgement positive
17	Functional Acknowledgement negative
22	Dispatch OK, acknowledgement still due
23	Error during retransmission
24	Control information of EDI subsystem OK

36 Electronic signature not performed (timeout)

For example, if the Monk Translation errored due to a problem in the Control Record of the IDoc, then use the following Monk statement to set the Translation Status global variable:

```
(define MONK_X_Status "ERROR:04")
```

- **WRAPUP** (SAP-outbound Only)

  The translation is successful, and it's time to wrap up the file being gathered. An Event is returned as a single string from the Monk function and is gathered into a new file

- **WRAPUPWORKAREA** (SAP-outbound Only)

  The translation is successful but the data is contained instead in a file given by the **Work Area File Monk Variable**, and it's time to wrap up the file being gathered.

- **WRAPUPNODATA** (SAP-outbound Only)

  This translation is successful but there is no data available and it's time to wrap up the file being gathered.

Resultant Status Record Monk Variable (SAP to e*Gate Only)

Description

Specifies the global Monk Variable which stores the ID Status Record(s) resulting from translating each IDoc from the Inbound Data File. This parameter needs to be specified if an IDoc Data File Layout has been previously selected.

Required Values

MONK_ResultStatRecord.

Additional Information

SAP R/3 requires the Status Record to determine what happened to the SAP-outbound IDoc file in order to update it's own database.

There are three operational modes for Status Record generation: none, automatic, and manual:

- **none**

  If the SAP R/3 system has been altered or configured to not require the Status Record, you may choose to *not* generate one by setting the value for the Resultant Status Record Monk Variable to an empty string, such as:

  ```
  (define MONK_ResultStatRecord  " ")
  ```

- **automatic**

  This mode is activated when the Monk Translation used does *not* define the value of the Resultant Status Record Monk Variable. The type of Status Record generated depends on the returned Translation Status Monk Variable. An **ERROR** value generates a type **05** (Error during translation) Status Record, whereas all other

values, except for **HOLD** and **OKNODATA**, automatically generate a type **12** (Dispatch OK) Status Record

- **manual**

  In this mode, the user generates the desired Status Record via the Status REcord Accessory Monk functions (see below)

  When manually generating the Status Record, the following minimum number of fields from the Status Record must be set:

TABNAM	Copy from inbound IDoc's control record
DOCNUM	Copy from inbound IDoc's control record
LOGDAT	Current date in CCYYMMDD format
LOGTIM	Current time in HHMMSS format (24 hour)
STATUS	SAP Status code (see below)
ARCKEY	Location of where Inbound Data File was archived (70 characters max)

Built-in **Status Record Accessory** Monk functions to facilitate the settings of these fields are available from the e*Gate Collaboration Editor Library button with the following forms.

For R/3 release 4.x

```
(SAP4_SR_Put_XXXXXX StatRecEvent instance value)
```

For R/3 release 3.x:

```
(SAP3_SR_Put_XXXXXX StatRecEvent instance value)
```

where:

- **XXXXXX** is the name of a Status Record Field; for example, SAP4_SR_Put_STATUS (or SAP3_SR_Put_STATUS) to set the status field
- **StatRecEvent** is the Status Record Event Monk variable being built
- **instance** is the instance, beginning with zero (0), of the Status Record for this IDoc; that is, multiple Status Records can be generated to represent different stages of translation
- **value** is the value with which to set the particular field

For example, to set the **STATUS** field for a successful translation, use Monk instructions such as the following:

```
(define statRecEvent " ")
(define statRecEvent #f)
(define statRecEvent (SAP4_SR_Put_STATUS statRecEvent 0 "12"))
(define statRecEvent (SAP3_SR_Put_STATUS statRecEvent 0 "12"))
```

When setting the status field, the following two-digit codes are recognized by the SAP R/3 Host:

04	Error within control information of EDI subsystem
05	Error during translation

06	Translation OK
07	Error during syntax check
08	Syntax check OK
09	Error during interchange handling
10	Interchange handling OK
11	Error during dispatch (general error during execution of Monk function)
12	Dispatch OK (This is the SUCCESSful translation code)
13	Retransmission OK
14	Interchange Acknowledgement positive
15	Interchange Acknowledgement negative
16	Functional Acknowledgement positive
17	Functional Acknowledgement negative
22	Dispatch OK, acknowledgement still due
23	Error during retransmission
24	Control information of EDI subsystem OK
36	Electronic signature not performed (timeout)

Status Record(s) returned from the Monk Translation function are appended to a correlating **Status Record File** and the file is saved when all the IDoc records from the Inbound Data file have been processed.

See also:

Status Record Gathering Directory (SAP to e*Gate Only) on page 116

Status Record Gathering Directory (SAP to e*Gate Only)

Description

Specifies the directory on the Local file system in which IDoc Status Record files are built, prior to being sent to the SAP R/3 Host to acknowledge the Inbound Data File received. This parameter needs to be specified if an IDoc Data File Layout has been previously selected.

Required Values

A valid directory, with an optional absolute path.

If you provide a directory that is *not* an absolute path, then the value of **SystemData** from **.egate.store** is prefixed to the value of the parameter automatically.

Status Record Trigger Script (SAP to e*Gate Only)

Description

Specifies the name of the shell script which is invoked to transport the Status Record file to the SAP R/3 Host, and then trigger SAP R/3 to read in and process the Status Record file. This parameter needs to be specified if an IDoc Data File Layout has been previously selected *and* a Status Record file is to be sent back to SAP R/3. If SAP R/3 does not require a Status Record file, this parameter *may* be left unspecified to suppress the sending of the Status Record

Required Values

A file name, with an optional absolute path.

If you provide a file name that is *not* an absolute path, then the value of **SystemData** from **.egate.store**, is prefixed to the value of the parameter automatically.

Additional Information

The script receives a fully qualified file name, and a base name for the Status Record file as parameters and has the following environmental variables defined:

Variable	Description
EGSYSTEM	e*Gate System Data Directory, for example **/home/eGate**
FTUSER	File Transport User Name (if defined above)
FTPASSWORD	decrypted File Transport User Password
SAPUSER	SAP R/3 User Name
SAPPASSWORD	decrypted SAP R/3 User Password

Trigger Script Execution Mode (SAP to e*Gate Only)

Description

Specifies the execution mode for the Status Record Trigger Script. This parameter needs to be specified if an IDoc **Data File Layout** has been selected previously.

Required Values

One of the following:

- **Background**

  The script is executed in a separate program thread, allowing the immediate processing of another inbound IDoc file.

- **Foreground**

  The e*Way ensures that the Trigger Script is executed completely before processing another inbound IDoc file.

See also:

Status Record Trigger Script (SAP to e*Gate Only) on page 117

Work Area File Monk Variable

Description

Specifies the name of the global Monk Variable which determines the location of the Work Area File. The e*Way presets this Monk Variable prior to calling the Monk Translation function to a guaranteed unique file name under a directory specified above with the Temporary Working Directory parameter, which the Monk Translation function can use as-is, or could overload to another unique file name if so required.

Required Values

MONK_WorkFile

Additional Information

The Monk Translation function can choose to place voluminous data (using Monk File I/O functions) in this Work Area File (rather than storing in memory) and then set the Translation Status to **OKWORKAREA**.

For example, if your Monk Translation function stores the output records somewhere else, use something like the following Monk statement:

```
(define MONK_WorkFile "c:/temp/SAP Idoc.txt"
```

This file is free-format and can be manipulated via Monk File I/O functions:

Function *Parameter*	Description
file-exists? *file_name*	Returns a boolean #t or #f depending on whether or not the file exists.
open-output-file *file_name*	Returns a file port for a new output file. If the file existed already, it is truncated and reopened.
open-append-file *file_name*	Returns a file port to a file for appending. If the file doesn't already exists, a new one is created and opened.
open-random-access-file *file_name*	Returns a file port to an existing file for update (reading and writing).
close-port *file_port*	Closes file port opened by (open-output-file...) (open-append-file...) (open-random-access-file...)
file-delete *file_name*	Deletes a file.
ftell *file_port*	Returns the current file offset.
seek *file_port file_offset*	Positions the file to the specified offset.

Function *Parameter*	Description
seek-to-end *file_port*	Positions the file to the end.
rewind *file_port*	Positions the file to the beginning.
display *string file_port*	Writes the string to the file.
read *file_port number_of_bytes*	Returns the string of requested length read from the file.
read-line *file_port number_of_bytes*	Returns the line (subject to max length specified) read from the file.

Input File Monk Variable (SAP to e*Gate Only)

Description

Specifies the name of the global Monk variable which has the fully-qualified path name of the Input file being processed by this e*Way, after being archived.

Required Values

MONK_InFile

Additional Information

This information is necessary to complete the IDoc Status Record if an IDoc **Data File Layout** is selected.

Input File Position Monk Variable (SAP to e*Gate Only)

Description

Specifies the name of the global Monk variable which has the file offset position corresponding to the start of the Event read in from the Input file being processed by this e*Way. When the End-Of-File is reached, this value becomes **EOF**, and no significant data is handed to the Monk Function for transformation. This mechanism is provided to trigger a release of any stored data **HELD** by the Monk function—for example, when many data records are read in from the Input Data file and accumulated until some condition is satisfied or an End-Of-File is reached.

For example, a Monk function (in pseudo monk code) could be:

```
(if (string=? MONK_InFilePosn "EOF")
  (begin
    ; returned any remaining gathered data
  )
  (begin
    ; gather data until some criteria is reached
  )
)
```

Note: *Some sort of End-Of-File logic must be implemented in the Monk transformation; otherwise the last record in an Input IDoc file is duplicated.*

Persistent Memory File Monk Variable

Description

Specifies the name of the global Monk Variable which locates the Persistent Memory File, useful for storing information that needs to transcend cycling (shutting down and restarting) of this e*Way and hence the associated Monk Engine. This Persistent Memory File is especially useful when the Translation Status is **HOLD**, and may contain the actual held Event itself or a pointer to the file containing the accumulated Event, such that if the e*Way were restarted, Event accumulation could continue until some criterion is achieved. The e*Way sets this Monk Variable to the value specified in the next configurable parameter, **Persistent Memory File Name**, prior to calling the Monk Translation.

Required Values

MONK_PMemFile

Persistent Memory File Name

Description

Specifies the file where the transitional Monk Translation data may be saved, such as when the Translation Status is **HOLD** (see above).

Required Values

A file name, with an optional absolute path.

If you provide a file name that is *not* an absolute path, then the value of **SystemData** from **.egate.store**, is prefixed to the value of the parameter automatically.

This file is free-format and can be manipulated via Monk File I/O functions:

Function *Parameter*	Description
file-exists? *filename*	Returns a boolean #t or #f depending on whether or not the file exists.
open-output-file *filename*	Returns a file port for a new output file. If the file existed already, it is truncated and reopened.
open-append-file *filename*	Returns a file port to a file for appending. If the file doesn't already exists, a new one is created and opened.
open-random-access-file *filename*	Returns a file port to an existing file for update (reading and writing).
close-port *file port*	Closes file port opened by (open-output-file...) (open-append-file...) (open-random-access-file...)

Function *Parameter*	Description
file-delete *filename*	Deletes a file.
ftell *file port*	Returns the current file offset.
seek *file port file offset*	Positions the file to the specified offset.
seek-to-end *file port*	Positions the file to the end.
rewind *file port*	Positions the file to the beginning.
display *string file port*	Writes the string to the file.
read *file port number_of_bytes*	Returns the string of requested length read from the file.
read-line *file port number_of_bytes*	Returns the line (subject to max length specified) read from the file.

For example, to create a file that contains the following:

LastIdocNumber=17

StorageFile=/home/eGate/data/gather.dat

The Monk code used could be:

```
(define LastIdocNumber 17)
(define StorageFile "/home/eGate/data/gather.dat")
(define fp (open-output-file MONK _PMemFile))
(display (format "LastIdocNumber=%d\n" (number->string
LastIdocNumber)) fp)
(display (format \"StorageFile=%s\n\" StorageFile) fp)
(close-port fp)
```

To read the parameters back from the file, the following Monk code could be used:

```
(define fp (open-random-access-file MONK_PMemFile))
(define line (read-line fp 80))
(define LastIdocNumber
   (string->number (substring line 15 (string-length line)))
)
(define line (read-line fp 80))
(define StorageFile (substring line 12 (string-length line)))
(close-port fp)
```

Determined File Name Monk Variable (e*Gate to SAP Only)

Description

Specifies the global Monk variable containing the Monk-Determined name for the Gathered File. The determined name must be just a base name, that is neither a fully qualified nor a relative path name.

Required Values

MONK_DetFilename.

Gathering Directory Name Monk Variable (e*Gate to SAP Only)

Description

Specifies the global Monk variable containing the configured NFS File Gathering Directory full path name. It can be used when determining the Gathered File name (**File Gathering Filter Type** set to **Monk Determined**) to check if a particular file exists (with Monk function **file-exists?**) before setting the Determined File Name Monk Variable.

Required Values

MONK_GatherDirname

See also:

Determined File Name Monk Variable (e*Gate to SAP Only) on page 121

File Bracket Indicator Monk Variable (e*Gate to SAP Only)

Description

Specifies the global Monk variable that informs the Monk Translation when the data sent to it correlates with the beginning or the ending of an originating Data file.

Required Values

MONK_FileBracket.

Additional Information

The value is preceded by the Start Of File Bracket Prefix and End Of File Bracket Prefix (specified below), respectively, for the two conditions, followed by the originating Data file name. When the End Of File Bracket indication is given, no significant data is given to the Monk Translation function.

Start Of File Bracket Prefix (e*Gate to SAP Only)

Description

Defines the prefix for an Event that represents a Start Of File Bracket Indicator from some Data file source. If file bracketing is required, this parameter *must* be specified.

Required Values

SOF|

End Of File Bracket Prefix (e*Gate to SAP Only)

Description

Defines the prefix for an Event that represents the End Of File Bracket Indicator from some Data file source. If file bracketing is required, this parameter *must* be specified, along with the Start of File Bracket Prefix parameter.

Required Values

EOF|

See also:

Start Of File Bracket Prefix (e*Gate to SAP Only) on page 122

Appendix A

Utilities

A.1 stcftp_get

Description

This utility routine queries the SAP R/3 Outbound Directory for the presence of an IDoc file. If found, the file is retrieved and stored in the e*Way's NFS Retrieval Directory.

Syntax

```
ftp_get -h host name -t host type -l local host -n net timeout -s
server timeout -f file size stability timeout -g -b -u user name -p
user password -d (Shared data directory| FTP heuristics file) (-a | -
r) [(remote filename local filename) | (remote filename local
dirname) | (remote dirname local dirname)]
```

Note: *It does not matter if a space is provided between the parameter and argument or not.*

Parameters:

Parameter	Argument	Description
-h	host name	Name of remote FTP server host.
-t	host type	Operating system of remote FTP server host (UNIX, VMS, MSFTPD 2.0, MVS). Default is UNIX.
-l	local host	Local host IP address (optional).
-n	net timeout	Network timeout, default = 30 sec.
-s	server timeout	Server timeout, default = 35 sec.
-f	file size stability timeout	Activity check to determine whether file is still growing or stable, default = 2 sec.
-g	(none)	Debugging switch, goes to file: ftp_get_dbg.xx (use only when necessary).
-b	(none)	Binary transfer mode switch (treats input as data, rather than ASCII text).
-u	user name	Remote system user name.
-p	user password	Remote system user password.

Parameter	Argument	Description
-d	Shared data directory	Shared data directory (default = \.egate.store\SyatemData)
	FTP heuristics file	Fully-qualified path for **FtpHeuristics.cfg**, including file name.
-a	remote/local dir/ filenames	Archive to specified remote directory (see note below).
-r	remote/local dir/ filenames	Remove remote file (see note below).

Notes

The parameters {-a} and [-r} are mutually exclusive. The directory and file names may be specified as:

- <remote filename> <local filename>
- <remote filename> <local dirname>
- <remote dirname> <local dirname>

The <dirname> must be terminated with a directory delimiter native to the respective operating system, as shown below.

OS	Delimiter
UNIX	/
VMS	]
MVS	(

A.2 stcftp_put

Description

This utility routine sends the designated IDoc file to the SAP EDI Interface for processing by the ABAP Program EDI_DATA_INCOMING.

Syntax

```
ftp_get -h host name -t host type -l local host -n net timeout -s
server timeout -f file size stability timeout -g -b -u user name -p
user password -d (Shared data directory| FTP heuristics file) (-a | -
r) local filename remote (filename | dirname)
```

Note: *It does not matter if a space is provided between the parameter and argument or not.*

Parameters:

Parameter	Argument	Description
-h	host name	Name of remote FTP server host.
-t	host type	Operating system of remote FTP server host (UNIX, VMS, MSFTPD 2.0, MVS). Default is UNIX.
-l	local host	Local host IP address (optional).
-n	net timeout	Network timeout, default = 30 sec.
-s	server timeout	Server timeout, default = 35 sec.
-f	file size stability timeout	Activity check to determine whether file is still growing or stable, default = 2 sec.
-g	(none)	Debugging switch, goes to file: ftp_get_dbg.xx (use only when necessary).
-b	(none)	Binary transfer mode switch (treats input as data, rather than ASCII text).
-u	user name	Remote system user name.
-p	user password	Remote system user password.
-d	Shared data directory	Shared data directory (default = \.egate.store\SyatemData)
	FTP heuristics file	Fully-qualified path for FtpHeuristics.cfg, including filename.
-a	remote/local dir/ filenames	Archive to specified remote directory (see note below).
-r	remote/local dir/ filenames	Remove remote file (see note below).

Notes

The parameters {-a} and [-r} are mutually exclusive. The directory and file names may be specified as:

- <remote filename> <local filename>
- <remote filename> <local dirname>

The <dirname> must be terminated with a directory delimiter native to the respective operating system, as shown below.

OS	Delimiter
UNIX	/
VMS	]
MVS	(

A.3 stcjdump.exe

Journal files can be converted to a readable flat file by using the **stcjdump.exe** utility, for example:

```
stcjdump.exe/home/eGate/journal/SapAleIn.journal/home/eGate/tmp/xxx
```

Usage:

Dumps an e*Gate System Journal file into a flat file or vice versa relative to the order of the -j and -f options

```
stcjdump.exe -j {journal_pathname} -f {flatfile_pathname}
[{delim_char_1} [...]]
```

where:

-j	precedes a Journal path name
-f	precedes a flat file path name
?_pathname	can be a relative path name, fully qualified path name, and for infile/outfile, '-' is contextually either standard in or standard out
delim_char_1	can be specified as '\r' or 0xd or 13 or ^M and separates records in the flat file
--ver	show version
-h	this screen

Note: *The addition of the -j and -f prefix before the respective filenames. Depending on whether the Journal Filename or Flat Filename is specified first, the action taken is to dump a Journal into a flat file or vice versa.*

Appendix B

IDoc Java Methods

Object methods associated with the SAP IDoc are as follows.

next

Description

Gets the next source event.

Signature

```
next()
```

Parameters

None.

Return Type

boolean

Throws

None

available

Description

This method checks to see whether or not the ETD has input data.

Signature

```
available()
```

Parameters

None.

Returns

A boolean **true** if data exists, **false** if no data exists.

Throws

None

send

Description

Puts the Event into the IQ as the pertinent Event Type.

Signature

```
send()
```

Parameters

None

Return Type

void

Throws

None

reset

Description

Resets the IDoc data content.

Signature

```
reset()
```

Parameters

None.

Returns

A boolean **true** if data content was reset successfully; **false** if not.

Throws

None.

getIdocMsgFormat

Description

Returns the IDoc message format used by this IDoc.

Signature

```
getIdocMsgFormat()
```

Parameters

None.

Returns

A string indicating the IDoc message format (either "**ALE**" or "**EDI**").

Throws

None.

setIdocMsgFormatALE

Description

Changes the IDoc message format to be **ALE**.

Signature

```
setIdocMsgFormatALE()
```

Parameters

None.

Return Type

void

Throws

None.

setIdocMsgFormatEDI

Description

Changes the IDoc message format to be **EDI**.

Signature

```
setIdocMsgFormatEDI()
```

Parameters

None.

Return Type

void

Throws

None.

idocMsgFormatIs

Description

Checks the IDoc message format.

Signature

```
idocMsgFormatIs(MsgFormat)
```

Parameters

Name	Type	Description
MsgFormat	java.lang.String	A string indicating the IDoc message format (either "ALE" or "EDI")

Return Type

boolean

Throws

None

Index

A

B

C

D

E

F

G

I

J

L

M

N

O

P

Q

R

S

T

U

W