IBM Sterling Gentran:Server for UNIX - Process Control Manager

Data Flow Administration Guide

Version 6.2

Copyright

This edition applies to the 6.2 Version of IBM® Sterling Gentran:Server® for UNIX - Process Control Manager and to all subsequent releases and modifications until otherwise indicated in new editions.

Before using this information and the product it supports, read the information in *Notices* on page N-1.

Licensed Materials - Property of IBM

IBM® Sterling Gentran:Server® for UNIX - Process Control Manager

© Copyright IBM Corp. 1991, 2011. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

About This Guide

- Welcome xii
- Contents of Chapters xiii
- Related Publications. xv
- Documentation Conventions xix

Understanding the Basics

Overview

- Introduction 1-2
- The Data Flow Administration Features 1-4
- Operating Environment 1-6
- File Name Conventions 1-9
- Basic Procedures 1-12
- The Data Flow Administration Main Menu 1-12
- How to Start and Exit Data Flow Administration 1-14
- How to Select an Option From a Menu or List 1-16
- How to Use Function Keys to Initiate an Action 1-17
- How to Display Version and Copyright Information 1-18

Introduction to Process Flows

Overview

- Introduction 2-2
- Process Flows 2-4
- Agents (Data Managers) 2-6
- Agent Personalities 2-8
- Agent Personalities and Flow Types 2-10
- Source and Destination of Files 2-11
- Example Inbound Process Flow 2-14
- Scripts 2-17
- Example Process Flow with Script 2-18

Designing a Process Flow

- Overview 2-19
- How to Identify a Flow's Purpose 2-20
- How to Select a Flow Type 2-22

- How to Add Other Components 2-23
- Example: Designing an Inbound Process Flow 2-24
- Process Flow Worksheet 2-28

Creating a Flow with the PCM Wizard

Overview

- Introduction 3-3
- The Process Control Manager Wizard 3-5
- Process Flows 3-7
- The Flow of Work 3-9

Beginning a Flow

- Overview 3-11
- Creating the Supporting Files 3-12
- Flow Identification Dialog Box 3-14
- How to Name and Describe the Flow 3-16

Creating an Inbound Flow

- Overview 3-18
- Source Setup Dialog Box (Inbound Flow) 3-20
- How to Set Up the Source Agent (Inbound Flow) 3-23
- Processing Agent Dialog Box (Inbound Flow) 3-25
- Translation Options Dialog Box 3-27
- How to Set Up the Processing Agent (Inbound Flow) 3-29
- Delivery Agent Dialog Box (Inbound Flow) 3-31
- How to Set Up the Delivery Agent (Inbound Flow) 3-35

Creating an Outbound Application Flow

- Overview 3-38
- Source Setup Dialog Box (Outbound Application) 3-40
- How to Set Up the Source Agent (Outbound Application) 3-43
- Processing Agent Dialog Box (Outbound Application) 3-45
- Translation Options Dialog Box (Outbound Application) 3-47
- How to Set Up the Processing Agent (Outbound Application) 3-49
- Delivery Agent Dialog Box (Outbound Application) 3-51
- How to Set Up the Delivery Agent (Outbound Application) 3-55

Creating an XML Flow

- Overview 3-58
- Source Setup Dialog Box (XML) 3-60
- How to Set Up the Source Agent (XML Flow) 3-62
- Processing Agent Dialog Box (XML Flow) 3-63
- Translation Options Dialog Box (XML Flow) 3-64
- How to Set Up the Processing Agent (XML Flow) 3-66

- Delivery Agent Dialog Box (XML Flow) 3-67
- How to Set Up the Delivery Agent (XML Flow) 3-71

Creating an Inbound NCPDP Flow

- Overview 3-75
- Source Dialog Box (Inbound NCPDP Flow) 3-77
- How to Set Up the Source Agent (Inbound NCPDP Flow) 3-79
- Processing Dialog Box (Inbound NCPDP Flow) 3-81
- Translation Options Dialog Box 3-83
- How to Set Up the Processing Agent (Inbound NCPDP Flow) 3-85
- Delivery Dialog Box (Inbound NCPDP Flow) 3-86
- How to Set Up the Delivery Agent (Inbound NCPDP Flow) 3-89

Completing a Flow

- Overview 3-92
- Error Handling Dialog Box 3-93
- How to Set Up Error Handling Instructions 3-95
- Trading Partner Records Dialog Box 3-97
- How to Add Trading Partnership Records to the Flow 3-99
- How to Delete Trading Partnerships from the Trading Partner Records Dialog Box 3-102

Using Flow Summaries

- The Flow Summary 3-104
- Flow Summary Views 3-106
- How to Expand and Collapse the Flow View 3-109
- Flow Summary Reports 3-110
- How to Print Flow Summary Reports 3-112

Maintaining Process Flows

- Overview 3-114
- How to Edit a Process Flow 3-115
- How to Delete Trading Partnerships From the Flow 3-117
- How to Delete a Flow 3-118

Using Queues

Overview

- Introduction 4-2
- Queues 4-4
- The Queue Process 4-6
- Queue Select Screen 4-7
- Queue File Screen 4-8
- How to Create a Queue 4-10

Maintaining Queue Entries

- Overview 4-12

- How to View Entries in a Queue 4-13
- Add Queue Entry Screen 4-15
- How to Add an Entry to a Queue 4-17
- How to Delete an Entry From a Queue 4-19

Maintaining Queues

- Overview 4-21
- How to Remove a Queue from the Select List 4-22
- How to Delete a Queue 4-23

Working with Scripts

Overview

- Introduction 5-3
- Sterling Gentran:Server Scripts 5-5
- The Script Manager 5-6
- Ways to Use Scripts 5-7

Parts of a Sterling Gentran:Server Script

- Introduction 5-8
- ENVIRON Group 5-11
- LOCKS Group 5-13
- DATA Group 5-17
- DPROCS Group 5-19
- STEPS Group 5-20
- PROCS Group 5-23
- RESULTS Group 5-27

Creating Sterling Gentran:Server Scripts

- Overview 5-30
- The Flow of Work 5-32
- Guidelines for Writing Scripts 5-33
- The Script Maintenance Screen 5-36

Procedures

- How to Add a Script with the Script Editor 5-38
- How to Add a Script with Another Editor 5-41
- How to Copy a Script 5-43
- How to Add Lock File Names to the LOCKS Directory 5-45

Working With Translation Scripts

- Translation Scripts 5-46
- How to Create a Translation Script 5-47

Maintaining Sterling Gentran:Server Scripts

- How to Edit a Script with the Script Editor 5-50
- How to Delete a Sterling Gentran:Server Script 5-52

Working with UNIX Mail Scripts

- Overview 5-54
- How to Add or Edit a UNIX Mail Script 5-55
- How to Delete a UNIX Mail Script 5-57

Running Scripts

Overview

- Introduction 6-2

Running Scripts

- Overview 6-3
- How to Run a Script from the Command Line 6-5
- How to Run a Script from Another Script 6-6
- How to Run a Script from the Script Maintenance Screen 6-8

Running Scripts on a Schedule

- Overview 6-10
- The Permanent Schedule 6-11
- The Permanent Schedule Maintenance Screen 6-12
- The Permanent Schedule Screen 6-15
- How to Create an Environment File 6-18
- How to Add a Script to the Permanent Schedule 6-19
- How to Copy a Schedule 6-21
- How to Change a Script's Processing Schedule 6-24
- How to Remove a Script From the Permanent Schedule 6-26

Restarting Scripts Automatically

- Overview 6-28
- The Recover Script 6-29
- How to Make a Script Restart Automatically 6-30

Defining the Document Reference Number

Overview

- Introduction 7-3
- The Document Reference Number 7-5
- How the Document Reference Number is Set 7-6
- Document Specifier Tables 7-9

Defining Document Reference Numbers

- Overview 7-11
- Document Reference Number Specifier Screen 7-12

Creating a Document Specifier Table

- Overview 7-14
- How to Add a Document Specifier Table 7-15

- How to Display a Document Specifier Table 7-18
- How to Copy a Document Specifier Table 7-21

Mapping Document Specifier Tables

- Overview 7-23
- EDI Add Screen 7-27
- APP Add Screen 7-32
- XML Add Screen 7-36
- NCPDP Add Screen 7-39
- How to Add a Map to a Document Specifier Table 7-42
- Using the Qualifier and Occurrence Fields 7-45
- How to Display a Map Picture 7-47

Attaching a Trading Partnership Code to a Table

- How to Attach a Trading Partnership Code to a Table 7-50
- How to Verify Trading Partnership Code Attachments 7-54

Maintaining Document Reference Number Tables

- Overview 7-56
- How to Change a Trading Partnership Code and Table Attachment 7-57
- How to Remove a Trading Partnership Code from a Table 7-58
- How to Delete a Document Specifier Table 7-60

Maintaining Document Specifier Maps

- Overview 7-61
- EDI Mapping Screen 7-62
- APP Mapping Screen 7-65
- XML Mapping Screen 7-68
- NCPDP Mapping Screen 7-71
- How to Display a Document Specifier Map 7-73
- How to Change a Document Specifier Map 7-75
- How to Delete a Map from a Table 7-78

Setting Up Life Cycle

Overview

- Introduction 8-2
- Database Software 8-4
- Life Cycle Components 8-6

Life Cycle Tables

- Life Cycle Table lc221 8-8
- Life Cycle Table: Informix 8-10
- Life Cycle Table: Oracle 8-14
- Life Cycle Table: Sybase 8-18

Life Cycle Configuration

- Overview 8-22
- Life Cycle Configuration Process 8-23
- How to Set Up Life Cycle for Oracle 8-26
- How to Set Up Life Cycle for Informix 8-32
- How to Set Up Life Cycle for Sybase 8-35
- How to Edit Data Manager Initialization Files 8-38

Testing Life Cycle Setup

- How to Test Outbound Processing 8-47
- How to Test Inbound Processing 8-49

Tracking Data with Life Cycle Files

Overview

- Introduction 9-2
- Understanding Life Cycle Records 9-4
- The Life Cycle Facility 9-6
- The Life Cycle Table 9-7

The Life Cycle Load Programs

- Overview 9-8
- The lcld and xlld Life Cycle Process 9-10
- The lcld Program 9-12
- How to Run lcld from the Command Line 9-14
- The xlld Program 9-15
- How xlld Updates the Life Cycle Table 9-16
- How xlld Handles Functional Acknowledgments 9-18
- How xlld Handles Update Failures 9-21
- The xlld Log File 9-22
- How to Run xlld from the Command Line 9-23

Life Cycle Reports

- Overview 9-24
- The Translation Activity Report 9-26
- The FA Exception Report 9-28
- The FA Due Report 9-30
- The TP List Report 9-32
- The Translation Traffic Report 9-34

Running Life Cycle Reports

- Overview 9-37
- How to Run Informix Life Cycle Reports 9-39
- How to Run Oracle Life Cycle Reports 9-40
- How to Run the Translation Traffic Report 9-41

Monitoring Processes

Overview

- Introduction 10-2
- The Screen Viewer 10-4

Monitoring Data Manager Processes

- Overview 10-6

Data Manager Control Screen

- Data Manager Log Files 10-10
- How to Check a Data Manager's Status 10-12
- How to View a Data Manager's Log File 10-13

Starting and Stopping Data Managers

- Overview 10-15
- How to Use the Data Manager Control Screen 10-16
- How to Stop Data Managers 10-19
- How to Start Data Managers 10-22

Maintaining Data Manager Log Files

- Overview 10-25
- How to Purge Data Manager Log Entries 10-26
- How to Delete a Data Manager's Log File 10-28

Monitoring Scripts

- Overview 10-29
- Script Logs and Journals 10-30
- How to Check the Status of a Script 10-32
- How to View a Script Log or Journal 10-34
- How to Display a Script's Processing Time Statistics 10-35

Maintaining Script Logs and Journals

- Overview 10-37
- How to Purge Entries from Journals 10-38
- How to Delete a Log or Journal 10-40

Notices

Glossary

Index

About This Guide

Contents

- Welcome xii
- Contents of Chapters xiii
- Related Publications xv
- Documentation Conventions xix

Welcome

Welcome to the *IBM® Sterling Gentran:Server® for UNIX - Process Control Manager Data Flow Administration Guide*.

Who should use this guide

This guide is for Sterling Gentran:Server users who develop, manage, and monitor electronic data interchange operations.

In this guide

This manual:

- Introduces you to Sterling Gentran:Server data flow administration concepts
- Shows you how to use the Process Control Manager (PCM) wizard to create your data flows
- Explains how to monitor your EDI processes.

In this preface

This preface:

- Describes the set of Sterling Gentran:Server user documentation
- Describes the contents of the chapters in this guide
- Lists the typographic conventions, symbols, and icons used in the documentation
- Explains how to get help.

Contents of Chapters

The *IBM® Sterling Gentran:Server® for UNIX - Process Control Manager Data Flow Administration Guide* contains 10 chapters. This table describes the contents of each chapter.

Chapter	Contents
Chapter 1, Understanding the Basics	Contains basic information about the host software features and basic operating procedures, such as how to access and exit the system.
Chapter 2, Introduction to Process Flows	Presents an overview of process flows and their components.
Chapter 3, Creating a Process Flow with the PCM Wizard	Describes the procedures for using the Process Control Manager (PCM) to create and maintain simple process flows.
Chapter 4, Using Queues	Describes queues and explains how to use them in your system.
Chapter 5, Working With Scripts	Describes the components of Sterling Gentran:Server scripts and explains how to create, maintain, and use scripts to control processes and carry out commands in your data flows.
Chapter 6, Running Scripts	Explains how to run scripts, monitor the results, and restart scripts automatically after a machine halt.
Chapter 7, Defining the Document Reference Number	Describes how to specify the characters in a document that the inbound data manager, the application data manager, and the translator use to determine the document reference number.
Chapter 8, Setting Up Life Cycle	Explains how to configure your system to use the Sterling Gentran:Server Life Cycle feature. Life Cycle enables you to load Sterling Gentran:Server event files to a relational data base.

(Contd) Chapter	Contents
Chapter 9, Tracking Data With Life Cycle Files	Explains how to use the Sterling Gentran:Server Life Cycle feature to load process event records to a relational database. (Optional)
Chapter 10, Monitoring Processes	Explains how to: ◗ View data manager log files ◗ Maintain data manager log files ◗ Monitor script processes ◗ Maintain script logs and journals.

Related Publications

Sterling Gentran:Server documentation

This table describes additional documentation for the Sterling Gentran:Server software.

Document	Description
IBM® Sterling Gentran:Server® for UNIX Upgrade and Data Conversion Guide	Instructions for upgrading from previous versions of IBM® Sterling Gentran:Server® for UNIX and IBM® Sterling Gentran:Server® for UNIX - Workstation. Also includes instructions for converting the files that are part of the upgrade.
IBM® Sterling Gentran:Server® for UNIX Installation Checklist	Description of the recommended sequence in which you should install and configure system components.
IBM® Sterling Gentran:Server® for UNIX Getting Started Guide	Instructions for installing the Sterling Gentran:Server software and performing setup tasks, such as setting up security. Instructions for starting and exiting Sterling Gentran:Server and for setting preferences and default values. Also includes instructions for checking files in and out and saving files.
IBM® Sterling Gentran:Server® for UNIX - Workstation Getting Started Guide	Instructions for installing the IBM® Sterling Gentran:Server® for UNIX - Workstation software and performing setup tasks. Instructions for starting and exiting Sterling Gentran:Server and for setting preferences and default values. Also includes instructions for checking files in and out and saving files.
IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide	Instructions for performing mapping and translation tasks using the Sterling Gentran:Server Application Integration system.
IBM® Sterling Gentran:Server® for UNIX HIPAA Compliance and NCPDP User Guide	Instructions for mapping and translating NCPDP files with the Application Integration system.

Document	Description
IBM® Sterling Gentran:Server® for UNIX GENCOD User Guide	Instructions for mapping and translating GENCOD files with the Application Integration system.
IBM® Sterling Gentran:Server® for UNIX VDA User Guide	Instructions for mapping and translating VDA files with the Application Integration system.
IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide	Describes processes, lists command-line commands in alphabetical order, and describes file record layouts and data type formats.
IBM® Sterling Gentran:Server® for UNIX - EC Workbench Data Flow Administration Guide	User instructions for configuring data flows using the Sterling Gentran:Server software.
IBM® Sterling Gentran:Server® for UNIX Maintenance and Troubleshooting Guide	Instructions for maintaining your Sterling Gentran:Server installation. Also provides troubleshooting information to help determine the cause and solution of problems that may occur.
IBM® Sterling Gentran:Server® for UNIX - Workstation Maintenance and Troubleshooting Guide	Instructions for maintaining your workstation installation. Also provides troubleshooting information to help determine the cause and solution of problems that may occur.
IBM® Sterling Gentran:Server® for UNIX with ADD User Guide	Instructions for configuring and using the Advanced Data Distribution system.
IBM® Sterling Gentran:Server® for UNIX XML Translation User Guide	Instructions for mapping and translating XML files with the Application Integration system.

Document	Description
IBM® Sterling Gentran:Server® for UNIX with ADD FTP Daemon User Guide	Instructions for configuring and using the FTP Daemon tool with IBM® Sterling Gentran:Server® for UNIX with ADD.
Online Help	Context-sensitive help screens describing the Sterling Gentran:Server dialog boxes and features. Also includes procedures for using the mapping and translation and the data flow administration software.

Other documentation

This table lists other types of documentation you may need to refer to when developing and maintaining your EDI processes.

Description	Source
Instructions for using the operating system on your UNIX computer	Documentation provided by your hardware vendor Documentation provided by the computer manufacturer
Information about one of the relational databases compatible with the Sterling Gentran:Server Life Cycle audit tracking facility	The Informix, Oracle, or Sybase documentation provided with your database product
Instructions for using the vi text editor or another text editor	Documentation provided with the text editor

Documentation Conventions

Typographic conventions

This table describes the typographic conventions used in this guide.

Convention	Use
Italics	This typeface is used for titles of other manuals and documents and for names of files and file extensions. **Example** *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide*
Bold	Bold type is used for program names, for key terms the first time they are used within a chapter, and for characters entered onto a screen. **Example** A **password** is a set of characters a user must enter to gain access to a system.
<Angle brackets>	Angle brackets indicate variable information such as a file name that you defined. **Example** *<scriptname>.scr*

Symbols used within syntax statements

This table describes symbols used within syntax statements.

Symbol	Use
< >	Substitute a value for any term that appears within angle brackets. Do not enter angle brackets unless specifically told to do so. **Example** **rm** <filename> means that you should type the name of the file you want to delete.
{ }	Braces indicate a required part of a statement. Do not enter the braces. **Example** {-f <filename>} means you must enter the f parameter followed by a filename.

(Contd) Symbol	Use
[ ]	Brackets indicate an optional part of a statement. Do not enter the brackets. **Example** [-f <filename>] means you could type the f parameter followed by a file name, but you are not required to do so.
…	An ellipse indicates that the immediately preceding item can be repeated indefinitely. Do not enter the ellipse. **Example** -e… means that you can repeat -e with other values.
( )	Parentheses should be entered as shown. They are part of the syntax of a statement and are not special symbols. **Example** (n) means that you should type a number enclosed by parentheses.

CHAPTER
1

Understanding the Basics

Contents

Overview

- Introduction 1-2
- The Data Flow Administration Features 1-4
- Operating Environment 1-6
- File Name Conventions 1-9
- Basic Procedures 1-12
- The Data Flow Administration Main Menu 1-12
- How to Start and Exit Data Flow Administration 1-14
- How to Select an Option From a Menu or List 1-16
- How to Use Function Keys to Initiate an Action 1-17
- How to Display Version and Copyright Information 1-18

Overview

Introduction

In this chapter

This chapter contains basic information about the Sterling Gentran:Server Data Flow Administration features and basic operating procedures.

Key terms

This table describes the key terms used in this chapter.

Term	Description
background process	A process that runs without user interaction.
electronic data interchange (EDI)	The application-to-application transfer of business transaction information in a standard format via a computer-to-computer communication link.
file name	In UNIX and DOS operating systems, a name that identifies a file to the system. In UNIX, the file can be a directory, subdirectory, or a data file.
foreground process	A process that requires user interaction through a terminal.
function key	A keyboard key (usually labeled F1, F2, F3, and so on) used to execute an option, such as saving a record.
group	In the UNIX operating system, a collection of user accounts. Users in the same group can share files and directories if they have the appropriate level of permissions.
host	The server in a client/server network that performs the system security, data storage, and major computing tasks. The Sterling Gentran:Server Data Flow Administration software resides on the host.
Main Menu	The primary list of options that is the starting point for most Data Flow Administration tasks.

(Contd) Term	Description
path name	In the UNIX operating system, a sequence of file names separated by slashes (/). The path name indicates the sequence of subdirectories traversed to get to the file.
permissions	In the UNIX operating system, a security measure that determines the level of access that a user has to read, write, and execute commands in files or directories.
server	See *host*.
shell	In the UNIX operating system, a program that takes user commands and changes them into terms that the UNIX system can understand and act upon.
shortcut key	A key sequence used to perform an action from a Data Flow Administration screen. **Example** The CTRL + A sequence accesses the UNIX command line.
Trading Partnership search	The Sterling Gentran:Server feature used to locate a Trading Partnership record.
user	In the UNIX operating system, a person who uses the system. Each user has a user identifier and account.

The Data Flow Administration Features

Introduction

Sterling Gentran:Server Data Flow Administration is a comprehensive feature that centralizes your **electronic data interchange** (EDI) functions into one computer environment.

EDI is the application-to-application transfer of business transaction information in a standard format via a computer-to-computer communication link. Some examples of standard formats are X12 and EDIFACT.

Set of tools

You can think of Data Flow Administration as a set of tools you can use to:

- Control the flow, translation, and processing of EDI documents
- Set up communications with host computers and networks
- Govern processing functions such as data archiving, error notification, and demands on scarce resources.

Knowledge required

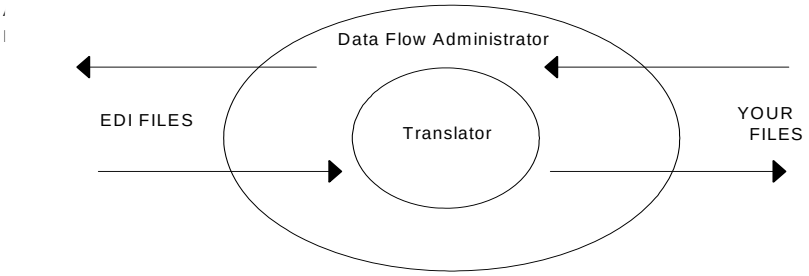
Data Flow Administration is easy to use. You do not need programming knowledge to accomplish most tasks. However, to fully understand and use Data Flow Administration capabilities, you should be familiar with the UNIX operating system and UNIX scripting commands.

Connects networks

Data Flow Administration connects separate data communications networks. It passes information between networks and processes data to prepare it for the Sterling Gentran:Server translator.

Illustration

This illustration shows the relationship between Data Flow Administration and the translator.



Note
If you have the Sterling Gentran:Server XML translation option, your system can process XML files as well as EDI files.

The translator

The Sterling Gentran:Server translator is the process control facility that translates the data format.

Operating Environment

Introduction

Data Flow Administration operates in the UNIX operating environment.

Users, groups, and permissions

To use a UNIX system, you must be set up as one of its **users**. Each user has a login name, a password, and an area of the electronic file system reserved for storing the user's files. Your system administrator creates your UNIX user accounts.

In addition, each user belongs to a **group**. Your system administrator assigns the group to which you belong when he or she creates your UNIX user account.

When you create a file, UNIX recognizes you as the owner of the file. Your UNIX administrator controls the kinds of access members of a group have to files you own by specifying the file **permissions** (read, write, and execute).

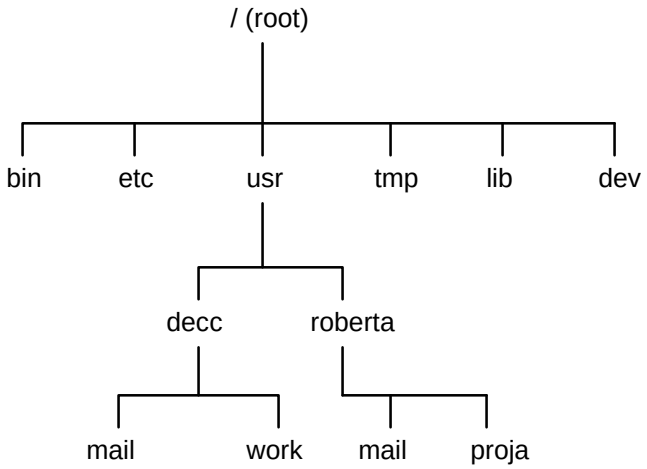
Administrative group permissions

For security reasons, the Sterling Gentran:Server administrative login ID is usually the only member of the Sterling Gentran:Server administrative group. All other users on the system are given either no permissions or read-only permissions for files that the administrative group owns.

Reference
See the *IBM® Sterling Gentran:Server® for UNIX Getting Started Guide* for information about creating an administrative user login account.

Directory tree

In UNIX, files are organized within a hierarchical directory structure. The UNIX file system structure looks like an upside down tree with the parent directory at the top. The highest level directory is named "/" and is the **root directory**. All other directories are subordinate to the root directory.



File names

A **file name** identifies a file to the system. In UNIX, the file name identifies a directory, a subdirectory, or a data file.

File names:

- Can include alphanumeric characters and underscores
- Are case sensitive, just like UNIX commands.

Path names

You can tell the system where to find a file by specifying its **path name**. The path consists of a sequence of file names (directories and subdirectories) separated by slashes (/). The path tells the system where to locate the file by showing the sequence of subdirectories you must traverse in the directory tree to get from the starting point to the file.

Example
The file name /usr/roberta/proja means:

- Start from the root directory and find the **usr** file
- Find the subdirectory named **roberta**
- Find the file named **proja**.

Relative path names

Path names are relative to the starting point. Relative directories are designated by periods. One period represents the current directory. Two periods represent the parent directory.

Example
If you were already in the *roberta* subdirectory, the path name would be *./proja*.

Shells

When you log on to UNIX, your interaction with the system is managed by a command interpreter called a **shell**. The shell analyzes and executes the commands you type at your terminal.

Examples
Korn shell

Bourne shell

WARNING

Sterling Gentran:Server scripts run in the Korn shell.

Foreground and background processing

A process that requires user interaction through the terminal is called a **foreground** process. A process that runs without intervention is called a **background** process.

Sterling Gentran:Server Data Flow Administration is designed to run normal processes in the background.

Use of third-party products

Some Data Flow Administration features are supported by third-party products.

Example
Sterling Gentran:Server uses third-party database drivers and software to enable Sterling Gentran:Server to read and write to databases.

File Name Conventions

Introduction

This topic describes the file name conventions and limitations you must follow when you name a Sterling Gentran:Server file or directory.

Directory and path name length

Directory names can be up to 63 characters in length, not counting the slash or backslash character used to separate a directory name from a subdirectory name.

The total maximum for a path, file name, and file name extension is 128 characters.

Validation

When you save or open a file, Sterling Gentran:Server checks the length of the file name and the characters used in the file name. If the file name exceeds the maximum length or if the file name includes an invalid character, Sterling Gentran:Server displays a message to alert you of the problem.

File name length

This table lists length restrictions of the file names of specific types of Sterling Gentran:Server files:

File Type	Maximum Length of Name
Map	60 (plus 4-character extension)
File definition	60 (plus 4-character extension)
Application description	60 (plus 4-character extension)
Implementation guide	60 (plus 4-character extension)
Input file	60 (plus 4-character extension)
Output file	60 (plus 4-character extension)
Mapping table	60 (plus 4-character extension)
Script	14 characters
Data manager	4 characters
Data manager pattern	15 characters
Transaction Register	9 characters
Queue	8 characters

Character limitations

This table lists the characters that Sterling Gentran:Server does not allow in a file name.

Character	Description
\	Backslash (reserved for separating directory, subdirectory, and file names)
/	Slash (reserved for separating directory, subdirectory, and file names)
*	Asterisk
“	Double quotation mark
‘	Forward single quotation mark
'	Backward single quotation mark
<	Less-than sign
>	Greater-than sign
|	Vertical bar
	Spaces
@	“At” sign
&	“And” character
(	Open round bracket
)	Closed round bracket
;	Semi-colon
:	Colon
?	Question mark
$	Dollar sign

Basic Procedures

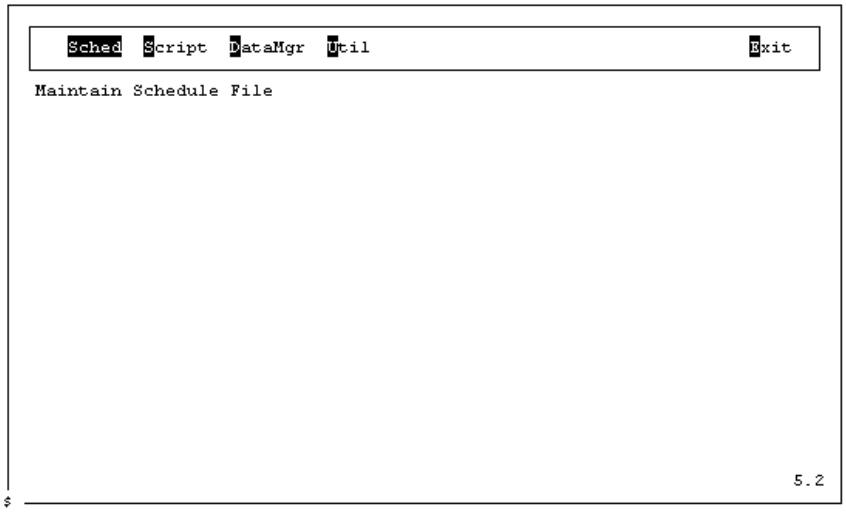
The Data Flow Administration Main Menu

Introduction

The Data Flow Administration Main Menu is the launching point for Data Flow Administration tasks on the host.

Illustration

This is the Main Menu for Data Flow Administration.



Menu options

This table describes the functions of the Main Menu options.

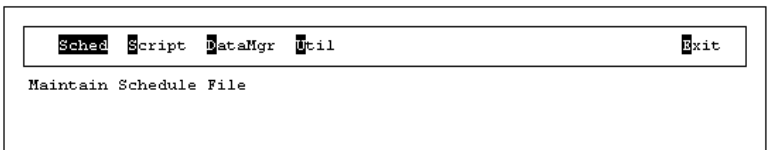
Option	Function
Sched	Accesses the Permanent Schedule function.
Script	Displays a list of scripts in the script library.
Audit	If the $EDI_AUDIT environment variable is set and your organization created a shell that accesses the database you use for Life Cycle data, this option accesses the Life Cycle database.

(Contd) Option	Function
DataMgr	Displays the **Data Manager Control** screen.
Util	Displays the Utilities menu.
Exit	Exits Data Flow Administration .

How to Start and Exit Data Flow Administration

Starting Data Flow Administration

Use this procedure to start Data Flow Administration.

Step	Action
1	Is Sterling Gentran:Server running? ◗ If YES, continue with Step 2. ◗ If NO, double-click on the Sterling Gentran:Server icon to start the system.
2	◗ Open a telnet/ssh client session to the server. ◗ From the $EDI_ROOT directory, run the program "server". **System Response** Sterling Gentran:Server prompts you for a login name and password.
3	Type the login ID and password of the owner of the environment and then press ENTER. **System Response** Sterling Gentran:Server displays a “connecting to Server” message and then displays the copyright screen.
4	Press any key on the keyboard. **System Response** Sterling Gentran:Server displays the Data Flow Administration Main Menu.  **WARNING** **If you receive the message server: command not found then check the PATH to make sure it contains $EDI_ROOT and $EDI_ROOT/bin.**

Exiting Data Flow Administration

Use this procedure to exit Sterling Gentran:Server Data Flow Administration.

Step	Action
1	Do you want to save your changes on the current screen? ◗ If YES, press the function key that corresponds to SAVE (usually F10). ◗ If NO, continue with Step 2. **Reference** See the How to Display Version and Copyright Information topic in this chapter for information about using function keys.
2	Press the function key that corresponds to QUIT (usually F9) or press the ESC key twice to return to the Data Flow Administration Main Menu.
3	Select **Exit** and press ENTER. **WARNING** **You can also exit** Sterling Gentran:Server **Data Flow Administration by pressing E for exit or by pressing the** ESC **key twice.**

How to Select an Option From a Menu or List

Introduction

There are two ways to select an item from a menu.

Method 1

Press the TAB key or arrow keys until the desired option is selected (highlighted) and then press ENTER.

Example
To select **Script** from the Main Menu, press the TAB key or an arrow key until **Script** is selected and then press ENTER.

Method 2

Another way to select an item from a menu is to press the selected letter in the option.

Example
Press the letter S to choose **Script** from the Main Menu.

Selecting from a list

To select an item from a list on a screen, you can type the first letter of the item. If more than one item begins with the letter, the first occurrence is selected. If you want to select a subsequent occurrence:

- Use an arrow key to select it, or
- Type the same letter again to select the next occurrence.

How to Use Function Keys to Initiate an Action

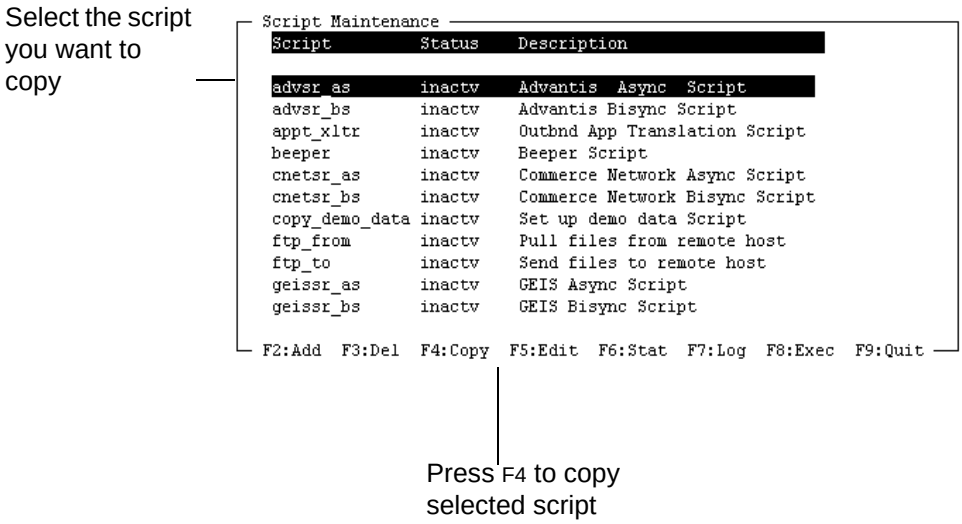
Introduction

Data Flow Administration screens have a row of function keys listed near the bottom of the screen. Use these keys to perform a function or to take action on an item you've selected.

Example

This example shows how function keys are used in a procedure.

- Select **Script** from the main menu.
- Select the name of the script you want to copy from the list.
- Press F4 to copy the script.



CAUTION

For the VT100 terminal emulator, you must use the ESC key with numeric keys in place of the function keys. Press the ESC key, release it, and then press the numeric key.

Function Key	VT100 Keys
F1- F9	ESC 1 - ESC 9
F10	ESC 0

How to Display Version and Copyright Information

Introduction

If you call IBM Customer Support, you may be asked for the version number of your software. The version number is on the Sterling Gentran:Server copyright screen.

When to use

Use this procedure when you need to determine the version of Sterling Gentran:Server that you are using.

Procedure

Use this procedure to display version and copyright information.

Step	Action
1	Select **Util** from the Main Menu. **System Response** Sterling Gentran:Server displays the Utility menu.
2	Select **About...** from the Utilities menu. **System Response** The system displays the copyright screen.

CHAPTER
2

Introduction to Process Flows

Contents

Overview

- Introduction 2
- Process Flows 4
- Agents (Data Managers) 6
- Agent Personalities 8
- Agent Personalities and Flow Types 10
- Source and Destination of Files 11
- Example Inbound Process Flow 14
- Scripts 17
- Example Process Flow with Script 18

Designing a Process Flow

- Overview 19
- How to Identify a Flow's Purpose 20
- How to Select a Flow Type 22
- How to Add Other Components 23
- Example: Designing an Inbound Process Flow 24
- Process Flow Worksheet 28

Overview

Introduction

In this chapter

This chapter introduces the basic components of a process flow that you build with the Process Control Manager (PCM) wizard.

Reference
See the Creating a Flow with the PCM Wizard chapter in this guide for information about using the Process Control Manager.

Key terms

This table lists the key terms used in this chapter.

Term	Description
data manager	An intelligent agent program that periodically scans a directory or queue for data files and then processes the files it finds. Processing can include: ◗ Routing data ◗ Invoking scripts ◗ Archiving data ◗ Handling data errors.
delivery agent	The last (third) agent in a process flow created with the PCM wizard.
destination directory	The end location of data in a process flow step.
intelligent agent	An event-driven computer program that can operate without interaction from a person at a computer terminal. Data managers are intelligent agents.
personality	The agent type, such as inbound, application, translation, and so on. The personality determines what type of processing the agent performs on the data.
process flow	A set of processing components that process and move data from one location in the Sterling Gentran:Server system to another.

(Contd) Term	Description
processing agent	The second (middle) agent in a process flow created with the PCM wizard.
purpose statement	The statement that summarizes the goal of a process flow.
queue	A list of files to be processed.
run directory	The directory that an agent uses to process files.
script	A set of commands that controls processes or performs some action.
script directory	The directory that contains all Sterling Gentran:Server scripts.
Script Manager	The Sterling Gentran:Server program that directs the script interpreter to execute the commands in a script.
source agent	The first agent in a process flow created with the PCM wizard.
source directory	The starting location of data in a process flow.
Trading Partner record	One of the five records maintained in trading partner files: Trading Partnership record, Interchange Organization record, Group Organization record, Contact record, and TRADACOMS record.
translation script	A Sterling Gentran:Server script that starts the translator, **lftran**, and other Sterling Gentran:Server programs. A translation data manager invokes a translation script.
UNIX mail script	A UNIX script that you can use to send electronic mail messages based on the results of a Sterling Gentran:Server script operation.
Value Added Network (VAN)	A nationwide or worldwide communications network owned by a third party that contracts with companies to provide network services.
work directory	The directory or queue in which an agent looks for the files or queued files it is to process.

Process Flows

Definition

A process flow is a set of Sterling Gentran:Server components that handles and moves data from one point to another in your system.

Basic types of process flows

In this chapter, we discuss the two main types of process flows:

- **Inbound**, which routes and translates EDI data that a trading partner sends to you
- **Outbound application**, which processes and translates your application data into EDI data so that you can send it to a trading partner.

Inbound process flows

Inbound process flows route and translate data from an EDI standard format to:

- Another EDI standard format (standard-to-standard)
- An application format (standard-to-application).

Outbound application process flows

Outbound process flows route and translate data from an application format to:

- Another application format (application-to-application)
- An EDI standard format (application-to-standard).

Optional process flows

If you have a Sterling Gentran:Server optional product, such as the XML translation option, you have available additional types of process flows.

Examples

Here are some examples of optional process flows:

- XML-to-application
- standard-to-XML

Source and destination

You can think of a process flow as a way to move data from a **source** in your system to a **destination**. Along the way, the processing components in the process flow change the data in some way to prepare it for the next step in the flow.

Illustration

This illustration shows the basic process flow between files in a source directory and a destination directory.

Source

The source of files for the start of a process flow is one of the following:

- **Directory**, which contains the actual files.
- **Queue**, which contains a list of information that enables a data manager to find the files they are to process, but not the actual data files.

Reference
See the Using Queues chapter in this guide for detailed information about queues.

Destination

The destination of files for the process flow is one of the following:

- **Directory**, which contains the actual files.
- **Queue**, which contains a list of information that enables a data manager to find the files they are to process, but not the actual data files.
- **Set type**, which selects the transaction set type as the symbolic value for the destination directory. This selection results in multiple possible destinations.
- **TP Code**, which selects the Trading Partnership Code as the name of the destination directory. This selection results in multiple possible destinations.
- **Categories**, which selects a Trading Partnership category as the name of the destination directory. This selection results in multiple possible destinations.

Components of a process flow

These are the main processing components of a process flow:

- Agents
- Scripts.

Agents (Data Managers)

Introduction

The primary processing components in a Sterling Gentran:Server process flow are called **agents**.

In a process flow, agents:

- Retrieve data from a directory or queue
- Process data
- Route data between directories and queues
- Handle data that contains errors
- Invoke other process flow components, such as scripts
- Deposit processed files into a directory or queue.

Also known as data managers

In Sterling Gentran:Server, agents are sometimes called **data managers**.

Agents in a flow

Process flows you create with the PCM wizard have three agents:

- **Source agent** begins the process flow
- **Processing agent** invokes the translator
- **Delivery agent** completes the flow.

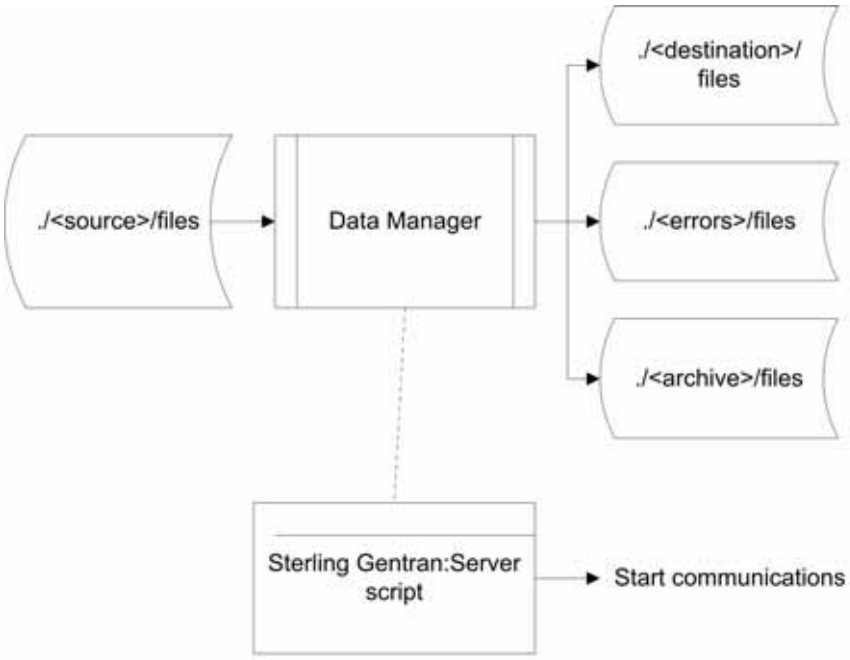
The exact role and function of each agent vary according to the type of flow.

Agent names

Each agent in your system has a unique name that is up to four characters in length. The PCM wizard automatically supplies a name when you create the agent, but allows you to change the name.

Illustration

This illustration shows the role of an agent in a process flow.



Railroad track analogy

One way to view data flowing through Sterling Gentran:Server is to think of the data as following a railroad track with switching points. For data to be switched to another track, it must be in the right format and have the right name. Agents are the components that serve as switching devices.They sort data, reformat or transform it, and route it to the appropriate destination.

Agent Personalities

Introduction

Agents have different personality types. The personality distinguishes the:

- Type of data to which the agent is designed to respond
- Possible actions the agent can take.

Personality table

This table lists the functions of the various types of agent personalities.

Agent	Sample Names	Functions
Inbound	inbd edii edio	◗ Understands EDI data (X12, EDIFACT, etc.) and verifies that the data is properly enveloped, and constructed. ◗ Sorts the input file by trading partner at the interchange, group, or set level. ◗ Records a document reference number, which is used to detect duplicate data and to retrieve archived data.
Application	appm	◗ Compares the data file to the application description or file definition as specified by the *<filename>.app* or *<filename>.ddf* file used in the map for this trading partner. ◗ Can sort the input file by trading partner at the set level. ◗ Records a document reference number, which is used to detect duplicate data and to retrieve archived data.
Translation	xltr xlti xlto	Calls the Script Manager to execute either a translation script or another embedded script. The translation script calls the translator and other runtime programs, such as **ediarc** and **envelope**. Translation agents handle either application or EDI data. Translation can be to a database table.

(Contd) Agent	Sample Names	Functions
XML (Available only if you have the XML translation option)	xmli xmlo	◗ Processes XML data and checks syntax. ◗ Splits XML files according to the splitter table rules and determines the Trading Partner code. ◗ Records a document reference number, which is used to detect duplicate data and to retrieve archived data.

Agent Personalities and Flow Types

Introduction

When you create a process flow, you select a flow type, such as standard-to-application. Sterling Gentran:Server selects the agent personalities for the source, processing, and delivery agents based on the flow type you specify.

Agents in basic flow types

This table lists the agent personalities in each type of flow.

Flow Type	Source Agent	Processing Agent	Delivery Agent
Standard-to-standard	Inbound	Translation	Translation
Standard-to-application	Inbound	Translation	Translation
Application-to-standard	Application	Translation	Translation
Application-to-application	Application	Translation	Translation

Agents in optional XML flow types

This table lists the agent personalities in the optional XML flows.

Flow Type	Source Agent	Processing Agent	Delivery Agent
Standard-to-XML	Inbound	Translation	Translation
XML-to-application	XML	Translation	Translation
XML-to-standard	XML	Translation	Translation
XML-to-XML	XML	Translation	Translation

Source and Destination of Files

Introduction

Every agent routes files from a source to a destination. Sources and destinations can be either **directories** or **queues**.

Directory contents

If the source or destination is a directory, the directory contains the actual data file.

Queue contents

A **queue** is a list of files to be processed. If the source or destination is a queue, the information in the queue points to the file's actual location. A queue does not contain data files.

There are some time-saving advantages to using queues.

Reference
See the chapter Using Queues in this guide for information about queues.

How files arrive

Another agent or a script deposits files into an agent's work directory or queue.

Source and destination directories and queues in a process flow

An agent's **work directory** or **work queue** is the source of the files the agent processes. An agent processes files when it finds files in its work directory or queue during a periodic scan.

An agent's **destination directory** or **destination queue** is the ending place of the files that the agent has processed.

Directories and queues you can choose

The PCM wizard allows you to choose the:

- Source directory or queue for the source agent
- Destination directory or queue for the delivery agent.

The PCM wizard automatically creates and names the other source and destination queues in a process flow.

Source agent

When you create a process flow, you decide if the source agent uses a directory or a queue as its source of files. If you select a queue, you select the queue that the source agent uses.

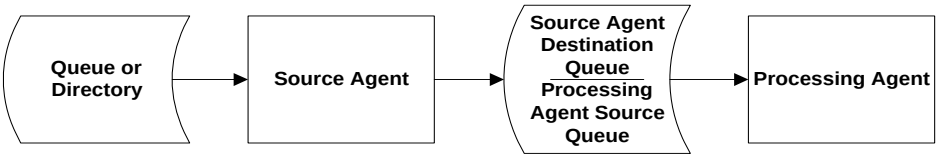
The source agent always uses a queue as the destination for the files it processes. The PCM wizard creates and names this queue for you. You cannot choose the queue.

Choose the directory or queue on the Source Agent dialog box

The PCM wizard creates and names the queue

Processing agent

The source of files for the processing agent is the source agent's destination queue. The PCM wizard automatically links these components.

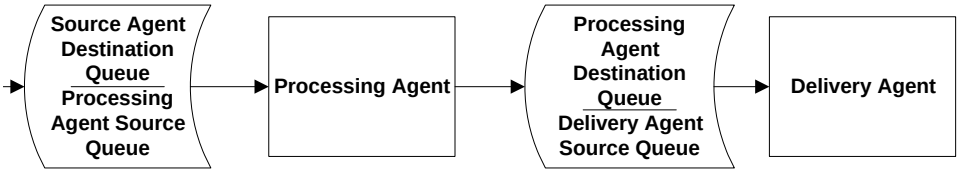


The PCM wizard creates and names the queue

The processing agent always uses a queue as the destination for the files it processes. The PCM wizard creates and names this queue for you. You cannot choose the queue.

Delivery agent

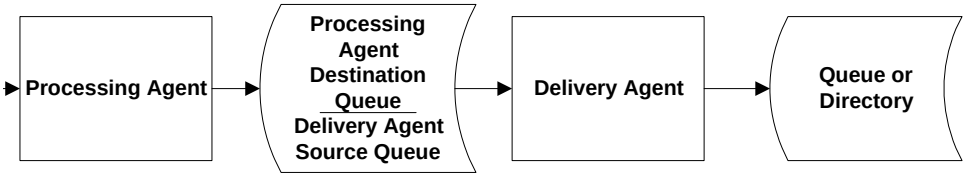
The source of files for the delivery agent is the processing agent's destination queue. The PCM wizard automatically links these components.



The PCM wizard creates and names the queue

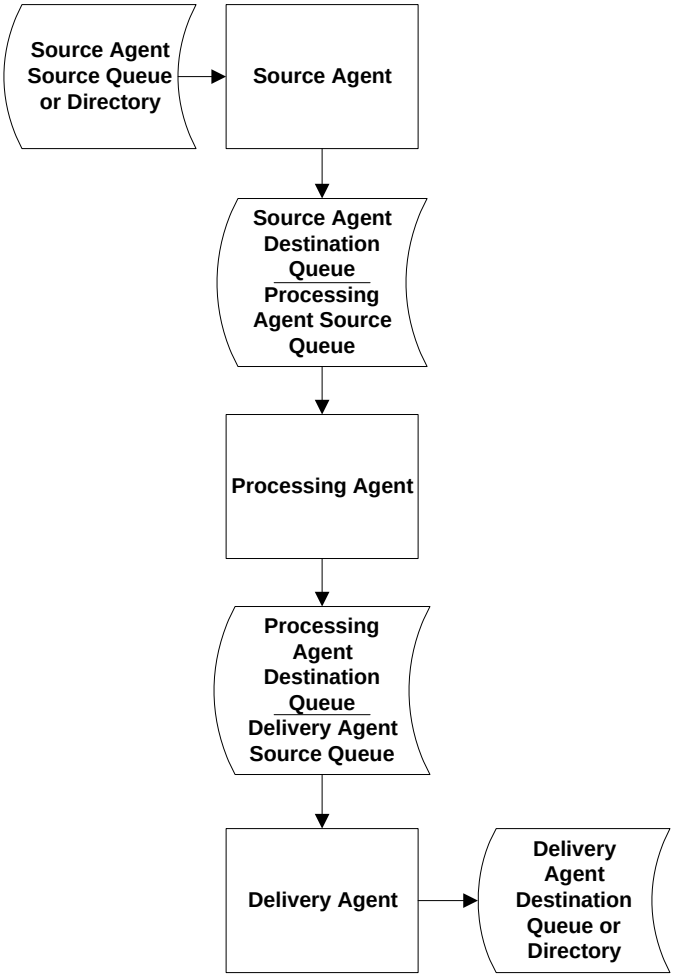
The PCM wizard creates and names the queue

When you create a process flow, you decide if the delivery agent uses a directory or a queue as the file destination. If you select a queue, you select the queue that the delivery agent uses.



Flow diagram

This diagram shows the complete flow.



Example Inbound Process Flow

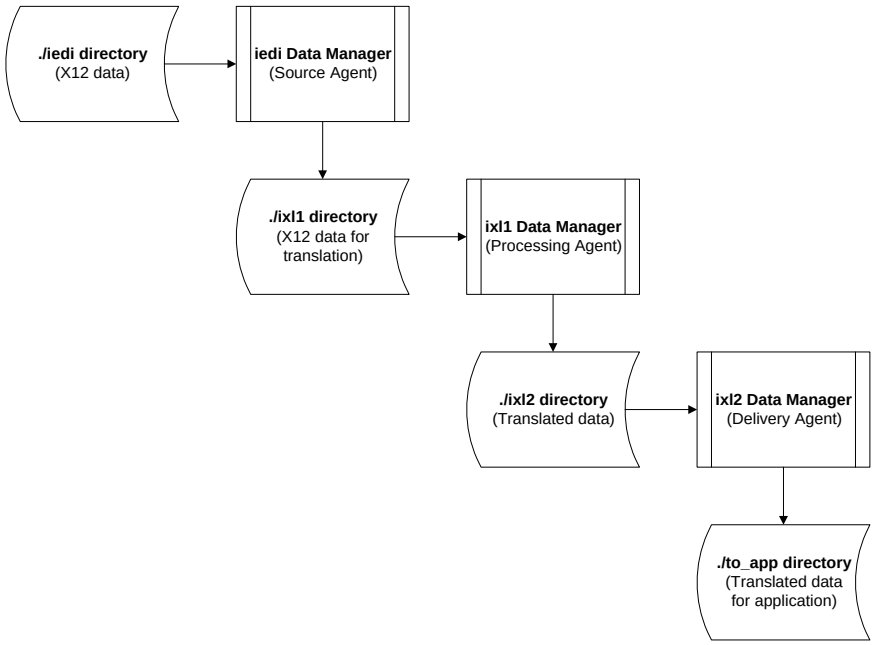
Introduction

An inbound process flow consists of a series of agents configured to:

- Accept data that is in an EDI standard format
- Invoke the translator to translate the data into an application format or into another EDI standard format
- Route the translated data.

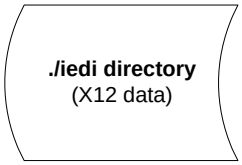
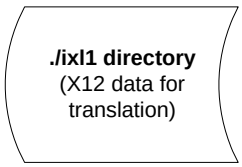
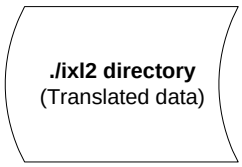
Illustration

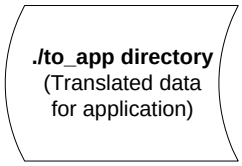
This illustration shows how process flow components move EDI data.



Parts of the process flow

This table describes the parts of example process flow.

Part	Description
	The source directory that contains data in X12 format. This is the iedi agent's work directory.
	The inbound agent that scans the *./iedi* directory for data and then processes the files it finds. The agent determines how to process the data by: a. Extracting the Trading Partnership code from the six key fields of the EDI interchange b. Locating the configuration record that tells the agent what to do with the data from the trading partner.
	The iedi1 agent's destination directory and the ixl1 agent's source directory. This is where the iedi1 agent sends files that it has processed. It is also the work directory that the ixl1 agent looks in for files to process.
	The inbound translation agent. This agent scans the *./ixl1* directory for files, invokes the translator to translate the files into application format, and routes the resulting files to the *./ixl2* directory.
	The ixl1 agent's destination directory and the ixl2 agent's source directory. This is where the ixl1 agent sends the translated files. It is also the work directory that the ixl2 agent looks in for files to process.

(Contd) Part	Description
	The delivery agent. This agent scans the *./ixl2* directory for files and routes the files to the *./to_app* directory.
	The destination directory for the ixl2 delivery agent. This is where the ix21 agent sends the translated data that is ready for the application.

Scripts

Introduction

Another common component of a process flow is a Sterling Gentran:Server script.

Definition

A Sterling Gentran:Server **script** is a set of commands that include:

- UNIX commands
- Names of data files you want used in the commands
- Discrete steps with statements that tell Sterling Gentran:Server what to do.

Role of scripts

Sterling Gentran:Server scripts start and control processes in a process flow. They also perform processing actions that are out of the scope of an agent.

You can use Sterling Gentran:Server scripts to:

- Pull host data through an Ethernet or other host connection and deposit it into an inbound agent's work directory
- Invoke agents, other Sterling Gentran:Server scripts, and shell scripts
- Start communication scripts
- Search for files that match certain conditions or patterns
- Convert and copy files
- Move files into a directory.

Reference

See the chapter Working with Scripts in this guide for more information about creating and using Sterling Gentran:Server scripts.

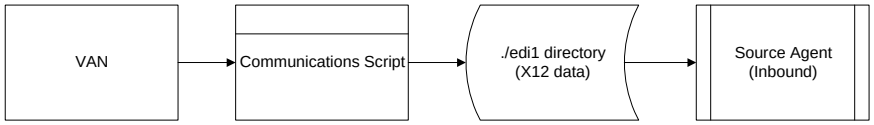
Example Process Flow with Script

Introduction

In an inbound process flow, you can use a Sterling Gentran:Server communications script to retrieve EDI data from a VAN and deposit it into the source agent's work directory.

Illustration

This illustration shows how a Sterling Gentran:Server script moves EDI data from a VAN to the source agent's work directory.



Designing a Process Flow

Overview

In this section

This section describes how to design a process flow.

Task summary

This table summarizes the tasks in process flow design.

Task	Description
1	Determine the purpose of the process flow. **Reference** See the topic How to Identify a Flow's Purpose in this section.
2	Select the type of flow. **Reference** See the topic How to Select a Flow Type in this section.
3	Add other components. **Reference** See the topic How to Add Other Components in this section.

How to Identify a Flow's Purpose

Introduction

The purpose of a process flow determines the design decisions that you make.

Questions to answer

To help identify the purpose of your process flow, answer these questions:

- Is the data coming into Sterling Gentran:Server or moving out of Sterling Gentran:Server?
- From where is the data coming?
- Where do you want the data to go?
- In what format is the data?
- What do you want Sterling Gentran:Server to do with the data?
  - Sort it
  - Move it
  - Reformat it
  - Translate it
  - Prepare it for outbound communications software

Purpose statement

To guide your design decisions, you should write a **purpose statement**. The purpose statement summarizes the answers to the above questions.

Examples of purpose statements

Here are some examples of purpose statements.

Example 1 - Inbound EDI
The purpose of my inbound EDI process flow is to:

- Retrieve inbound EDI data from a VAN
- Sort the data by group
- Translate the data into application data
- Move the data to the appropriate application software.

Example 2 - Outbound EDI

The purpose of my outbound EDI process flow is to:

- Translate my application data to the EDI data format that my trading partner uses
- Prepare the data for my communications software
- Send the data to my trading partner's VAN address.

How to Select a Flow Type

Introduction

The type of flow you select determines the personalities of the agents in your flow.

Section criteria

Select the flow type that:

- Handles the type of data or data format you want to process
- Processes the data in the way that you want it handled.

Selecting a flow type

Use this table to select the type of process flow.

Use this flow type	When you want to route and translate…
Standard-to-standard	EDI data from one EDI standard format to another EDI standard format.
Standard-to-application	EDI data from an EDI standard format to an application format.
Application-to-application	Data in an application format to another application format.
Application-to-standard	Data in an application format to an EDI standard format.

Selecting a an optional flow type

If you have a Sterling Gentran:Server translation option, such as the XML translation option, you have additional flow types available. Optional flow types appear in the drop-down list for the flow type on the Flow Identification dialog box.

How to Add Other Components

Introduction

After choosing the flow type, you need to think about the other process flow components.

Components list

This is a list of other common components in a process flow:

- Source directories or queues
- Destination directories or queues
- Communications scripts
- Other Sterling Gentran:Server scripts
- UNIX mail message scripts.

You must have source and destination directories or queues. Scripts are optional components that perform specific tasks.

Examples

This table contains a few examples of reasons to add other components to your process flow.

IF you want to…	THEN add…
Have Sterling Gentran:Server pull record layout file files from the application host	A script to your process flow that pulls files from the application host.
Move EDI files to a VAN	A communications script and directory for the VAN to your process flow.
Convert the data in record layout files so that your accounting system application can understand the data	A script or program to convert the data into a format the application understands.

Example: Designing an Inbound Process Flow

In this topic

This topic leads you through the process of designing an inbound process flow.

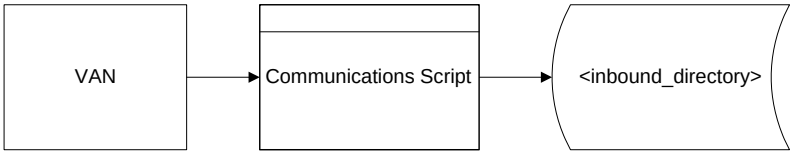
Purpose of the process flow

The first step in creating a process flow is to decide the purpose of the flow. In this example, our purpose is to:

- Retrieve inbound X12 data from a VAN
- Translate the X12 data into application data
- Forward the translated data to our application.

Moving the files from the VAN

We need to retrieve the inbound X12 data from the VAN, so we start our inbound flow with a communications script to pull the data from the VAN and deposit it into a directory.



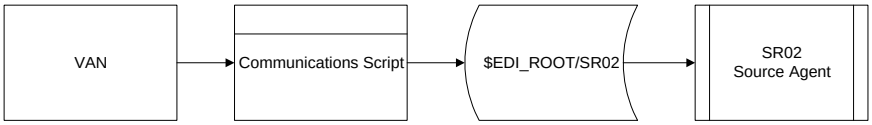
Selecting the flow type

The incoming data is in EDI format. We want to translate the data into an application format. Therefore, the flow type is “standard-to-application.”

Creating a directory for the files

The communications script needs a directory in which to deposit the X12 files; the source agent needs the same directory to look in for files. Because this directory is the source or **scan directory** for the agent that starts our process flow, we name the directory for the source agent, SR02.

Note
The PCM wizard creates directories for you if they do not exist.

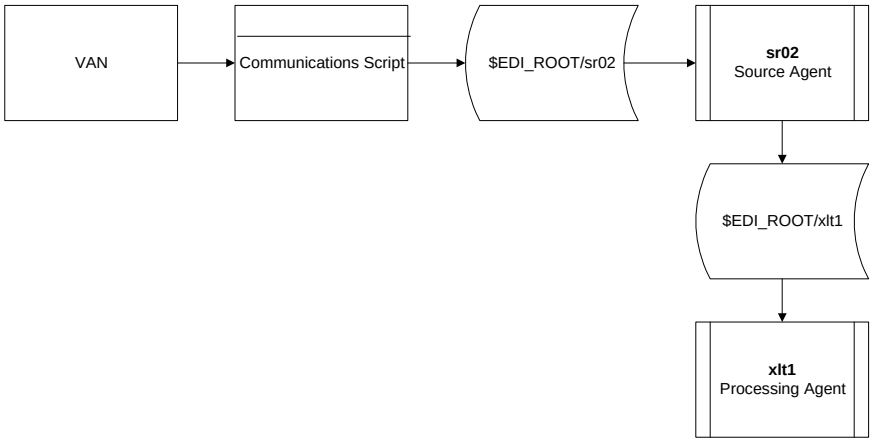


Configuring the processing agent

The processing agent is the second agent in a process flow. The processing agent has a translation personality. Its role is to start the **translation script**, which is a Sterling Gentran:Server script that invokes the translator. The translator translates the X12 data into a format that our application can understand. We name this agent xlt1.

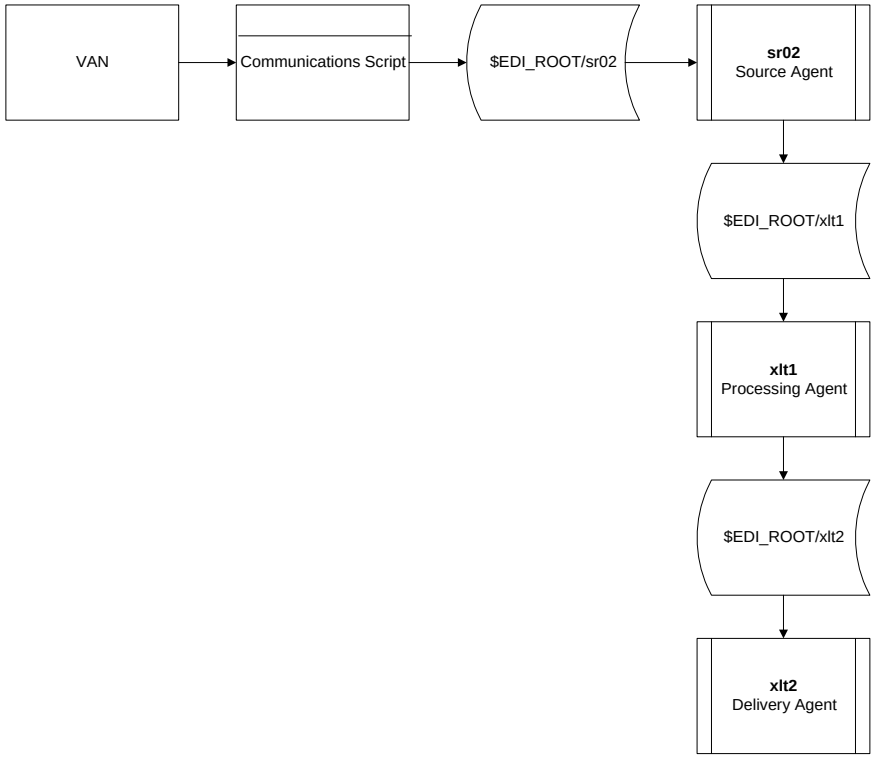
Comment

In this illustration, notice that the source agent's destination directory is the same as the processing agent's work directory. Also note that the directory is named for the processing agent. The PCM wizard automatically creates this directory for you.



Configuring the delivery agent

The delivery agent is the final agent in a process flow. The delivery agent has a translation personality. Its role is to route the translated data to a destination directory. We name this agent xlt2.



Moving the translated files

Getting the translated files to the application for processing is similar to retrieving the data from the VAN. We use a script to handle this final part of the process.

Process Flow Worksheet

Introduction

This topic contains a worksheet to help you develop your process flows.

Worksheet

Use this worksheet to identify the components that you need in your process flow.

Item	
Flow type (standard-to-standard, standard-to-application, application-to-standard, application-to-application)	
Purpose	Sterling Gentran:Server
Communication connections	
Names of application sending or receiving files to and from Sterling Gentran:Server	
Scripts needed to pull record layout files from application?	

(Contd) Item	
Scripts needed Sterling Gentran:Server to "push" incoming record layout file files to application host?	
Paths needed for rush or other special routing?	
User Defined File (record layout file) format requirements	
Preprocessing requirements	
Post-processing requirements	
Legal archiving requirements	
Format requirements of files sent to and received from a VAN	

(Contd) Item	
Types of error notification	
Where error messages will be directed	

Agents

Use this table to record the names of the agents in the flow.

Type	Name
Source	
Processing	
Delivery	

Scripts

Use this table to record the names of the scripts that you need to create.

Script Name	Function

Queues

Use this table to record the names of the queues that you need to create.

Name	Description

CHAPTER
3

Creating a Flow with the PCM Wizard

Contents

Overview
- Introduction . . . 3
- The Process Control Manager Wizard . . . 5
- Process Flows . . . 7
- The Flow of Work . . . 9

Beginning a Flow
- Overview . . . 11
- Creating the Supporting Files . . . 12
- Flow Identification Dialog Box . . . 14
- How to Name and Describe the Flow . . . 16

Creating an Inbound Flow
- Overview . . . 18
- Source Setup Dialog Box (Inbound Flow) . . . 20
- How to Set Up the Source Agent (Inbound Flow) . . . 23
- Processing Agent Dialog Box (Inbound Flow) . . . 25
- Translation Options Dialog Box . . . 27
- How to Set Up the Processing Agent (Inbound Flow) . . . 29
- Delivery Agent Dialog Box (Inbound Flow) . . . 31
- How to Set Up the Delivery Agent (Inbound Flow) . . . 35

Creating an Outbound Application Flow
- Overview . . . 38
- Source Setup Dialog Box (Outbound Application) . . . 40
- How to Set Up the Source Agent (Outbound Application) . . . 43
- Processing Agent Dialog Box (Outbound Application) . . . 45
- Translation Options Dialog Box (Outbound Application) . . . 47
- How to Set Up the Processing Agent (Outbound Application) . . . 49
- Delivery Agent Dialog Box (Outbound Application) . . . 51
- How to Set Up the Delivery Agent (Outbound Application) . . . 55

Creating an XML Flow

- Overview 58
- Source Setup Dialog Box (XML) 60
- How to Set Up the Source Agent (XML Flow) 62
- Processing Agent Dialog Box (XML Flow) 63
- Translation Options Dialog Box (XML Flow) 64
- How to Set Up the Processing Agent (XML Flow) 66
- Delivery Agent Dialog Box (XML Flow) 67
- How to Set Up the Delivery Agent (XML Flow) 71

Creating an Inbound NCPDP Flow

- Overview 75
- Source Dialog Box (Inbound NCPDP Flow) 77
- How to Set Up the Source Agent (Inbound NCPDP Flow) 79
- Processing Dialog Box (Inbound NCPDP Flow) 81
- Translation Options Dialog Box 83
- How to Set Up the Processing Agent (Inbound NCPDP Flow) . . . 85
- Delivery Dialog Box (Inbound NCPDP Flow) 86
- How to Set Up the Delivery Agent (Inbound NCPDP Flow) 89

Completing a Flow

- Overview 92
- Error Handling Dialog Box 93
- How to Set Up Error Handling Instructions 95
- Trading Partner Records Dialog Box 97
- How to Add Trading Partnership Records to the Flow 99
- How to Delete Trading Partnerships from the Trading Partner Records Dialog Box 102

Using Flow Summaries

- The Flow Summary 104
- Flow Summary Views 106
- How to Expand and Collapse the Flow View 109
- Flow Summary Reports 110
- How to Print Flow Summary Reports 112

Maintaining Process Flows

- Overview 114
- How to Edit a Process Flow 115
- How to Delete Trading Partnerships From the Flow 117
- How to Delete a Flow 118

Overview

Introduction

In this chapter

This chapter describes the procedures for using the Process Control Manager (PCM) wizard to create and maintain basic process flows that have three data managers.

Key terms

This table lists the key terms used in this chapter.

Term	Description
agent	A data manager.
application data manager	A data manager that processes files that are in an application format your organization defined.
category	A user-definable group to which you can assign Trading Partnerships.
configuration record	A record that describes how a data manager directs the data that it handles for a particular Trading Partnership code or file name. The record: ◗ Specifies the Trading Partnership code or file name that the data manager is to use to identify data ◗ Tells the data manager what to do with the data it has identified.
data manager	An intelligent agent program that periodically scans a directory or queue for data files and then processes the files it finds. Processing can include: ◗ Routing data ◗ Invoking scripts ◗ Archiving data ◗ Handling data errors.
delivery agent	The third data manager in a flow created with the PCM wizard. The role of the delivery agent depends on the flow type.

(Contd) Term	Description
flow type	The words that describe the direction of the process flow. If the source document is in a standard format, the system views the direction as inbound. If the source document is in an application format, the system views the direction as outbound. **Examples** ◗ Standard-to-standard ◗ Standard-to-application ◗ Application-to-application ◗ Application-to-standard
inbound data manager	A data manager (agent) that processes EDI data. Can distinguish between EDI data and non-EDI data and sort them into distinct files for routing or further processing.
process flow	A set of parameters and commands that describes how data is moved from a source to a destination.
processing agent	The second data manager in a flow created with the PCM wizard. The role of the processing agent depends on the flow type.
Process Control Manager (PCM)	The Sterling Gentran:Server feature that guides you through the process of creating a process flow that has three data managers.
source agent	The data manager that begins a process flow.
Trading Partnership Code	The unique code that identifies a Trading Partnership record.
Trading Partnership record	The record that contains basic trading partnership information, such as the Trading Partnership code, the translation map to be used when translating business documents for this partner, and whether an acknowledgment is to be generated.
translation data manager	A data manager that runs a script (such as a translation script) to process data.
wizard	A process that automatically presents, in order, a complete sequence of dialog boxes required to perform a task.

The Process Control Manager Wizard

Purpose

The **Process Control Manager (PCM) wizard** guides you through the process of creating a three-data-manager process flow and linking Trading Partnerships to it.

Configuration records

When you complete a flow with the PCM wizard, the Process Control Manager generates a **configuration record** for each Trading Partnership you added to the flow. The configuration records control how a data manager processes and directs data for a specific Trading Partnership code or file name.

Contents of a configuration record

A configuration record contains:

- The Trading Partnership code or file name that the data manager uses to identify data
- The destination locations for the processed data
- The name of the script (if any) the data manager runs when it encounters the data
- An archive indicator that directs the data manager to archive or not archive the data.

WARNING

You cannot access and edit individual configuration records with IBM® Sterling Gentran:Server® for UNIX - Process Control Manager.

Reasons to use the Process Control Manager

The Process Control Manager wizard is an easy way to create the records that drive the flow of data in your system. To help you structure a process flow, the **wizard** guides you through a sequence of dialog boxes. Each dialog box in the sequence represents either a flow component or set of processing instructions.

If you specify a directory that does not exist as the source or destination of data in your flow, the Process Control Manager wizard will create the directory for you. However, if you use queues as the source or destination of files, you must create them before you construct the flow with the wizard. The wizard will not create queues for you.

Limitations on multiple sessions

These are the limitations on multiple sessions:

- If you run multiple sessions of IBM® Sterling Gentran:Server® for UNIX, you can run the Process Control Manager once in a session for each host and environment combination.
- If you start Sterling Gentran:Server for the same host and environment in another Client session on the same Windows desktop, you must use the same user ID and password to log in to the new session.
- If you open another Client session for the same host and environment, you will have view-only mode. This means that you can view records in the session, but you cannot edit or delete them. Sterling Gentran:Server restricts editing options to the first Client session you opened.

Reference
See the *Running Multiple Sessions* topic in the *IBM® Sterling Gentran:Server® for UNIX Getting Started Guide* for more information about process limitations and file locking in multiple sessions.

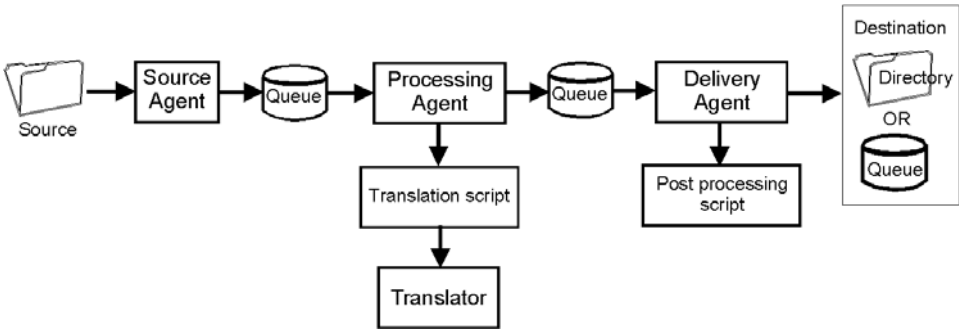
Process Flows

Definition

A **process flow** describes how data is moved from a source to a destination. You can think of it as a data stream that operates between a known source and a known destination.

Illustration

This illustration shows the movement of data from a source data manager to a destination data manager.



Components of a PCM process flow

A PCM process flow consists of:

- Flow identification information
- A source data manager
- A processing data manager
- A delivery data manager
- Error handling information
- A list of Trading Partnership records linked to the flow.

Flow identification

You must give each flow a unique name and optional description. You also identify the **flow type**, which indicates the flow direction.

Flow types

The flow type indicates the direction of the flow. These are the flow types available in the PCM wizard:

- Inbound
  - Standard-to-standard
  - Standard-to-application
- Outbound
  - Application-to-standard
  - Application-to-application

Optional flow types

If you have the Sterling Gentran:Server XML translation optionor the SAP translation option, you have additional flow types available to you.

Examples

Here are some examples of optional flow types:

- XML-to-application
- Standard-to-XML
- XML-to-XML
- XML-to-standard
- Standard-to-SAP
- SAP-to-standard

The Flow of Work

Task summary

This table summarizes the tasks you must complete to create a process flow with the PCM wizard.

<table>
<tr><th>Task</th><th colspan="2">Description</th></tr>
<tr><td>1</td><td colspan="2">Create the supporting files.

Reference
See Creating the Supporting Files.</td></tr>
<tr><td>2</td><td colspan="2">Name the new process flow.

Reference
See How to Name and Describe the Flow.</td></tr>
<tr><td rowspan="5">3</td><td colspan="2">Set up the data managers (agents) for the process flow. The procedure depends on the format of the input (source) file.</td></tr>
<tr><td>IF the input file format is…</td><td>THEN see this section…</td></tr>
<tr><td>An EDI standard</td><td>Creating an Inbound Flow</td></tr>
<tr><td>An application</td><td>Creating an Outbound Application Flow</td></tr>
<tr><td>XML (and you have the XML translation option)</td><td>Creating an XML Flow</td></tr>
<tr><td>An NCPDP standard</td><td>Creating an Inbound NCPDP Flow.</td></tr>
<tr><td>4</td><td colspan="2">Complete the process flow.

Reference
See Completing a Flow.</td></tr>
</table>

Flow guidelines

Follow these guidelines when creating a new process flow:

- Give each flow in your system a unique name.
- Use a unique name for each data manager in your system.
- Use the flow description to help identify the flow.

Beginning a Flow

Overview

Introduction

This section describes how to begin an inbound or outbound Process Control Manager process flow.

Task summary

This table summarizes the tasks you must complete to begin a PCM process flow.

Task	Description
1	Create the directories, queues, Trading Partnership records, and Trading Partnership categories that you want to use in the process flow. **Reference** See Creating the Supporting Files.
2	Name the flow and select the flow type. **Reference** See How to Name and Describe the Flow.

Creating the Supporting Files

Introduction

Before you create a new process flow, you must create the supporting files that you plan to use in the process flow. Supporting files include:

- Directories (nested)
- Queues
- Trading Partnership records
- Trading Partnership categories
- Scripts

This topic lists the tasks you need to complete to create the supporting files.

Directories (not nested)

If a directory you want to create is not nested, the Process Control Manager will create it for you under EDI_ROOT. You also can create it in advance and enter the directory name on the appropriate PCM wizard dialog box.

CAUTION

The PCM wizard always creates source and destination directories under EDI_ROOT. The wizard will not create a nested directory structure.

Directories (nested)

If you need to use a nested directory structure, you can create the directory path in advance and enter the path to the directory on the appropriate PCM wizard dialog box.

NOTE

We recommend that you allow the PCM wizard to create your source and destination directories under EDI_ROOT.

Queues

The PCM wizard will not prompt you to create queues. You must create the source and destination queues before you use the PCM wizard.

Task list

This table lists the three tasks you should complete to create the supporting files.

Task	Description
1	Do you want the source data manager to scan a queue to detect files? ◗ If YES, create the queue and then continue with Task 2. ◗ If NO, go to Task 2. **Reference** For instructions on creating queues, see the chapter Using Queues in this guide.
2	Create the destination you want to use in the process flow. This can be any of the following: ◗ Directory file name You must always create nested directories in advance, but you have a choice with un-nested directories. The PCM wizard will prompt you to create an un-nested directory under EDI_ROOT if you do not create it in advance. ◗ Queue ◗ Trading Partnership record ◗ Trading Partnership category. **References** ◗ For instructions on creating directories, refer to your UNIX manuals. ◗ For instructions on creating queues, see the chapter Using Queues in this guide. ◗ For instructions on creating Trading Partnership records and Trading Partnership categories, see your *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide*.
3	Do you want to run a Sterling Gentran:Server script after files are processed? ◗ If YES, create the script and move it to the *./script* directory. ◗ If NO, you are ready to create the process flow. **References** For instructions on creating scripts, see the chapter Working with Scripts in this guide.

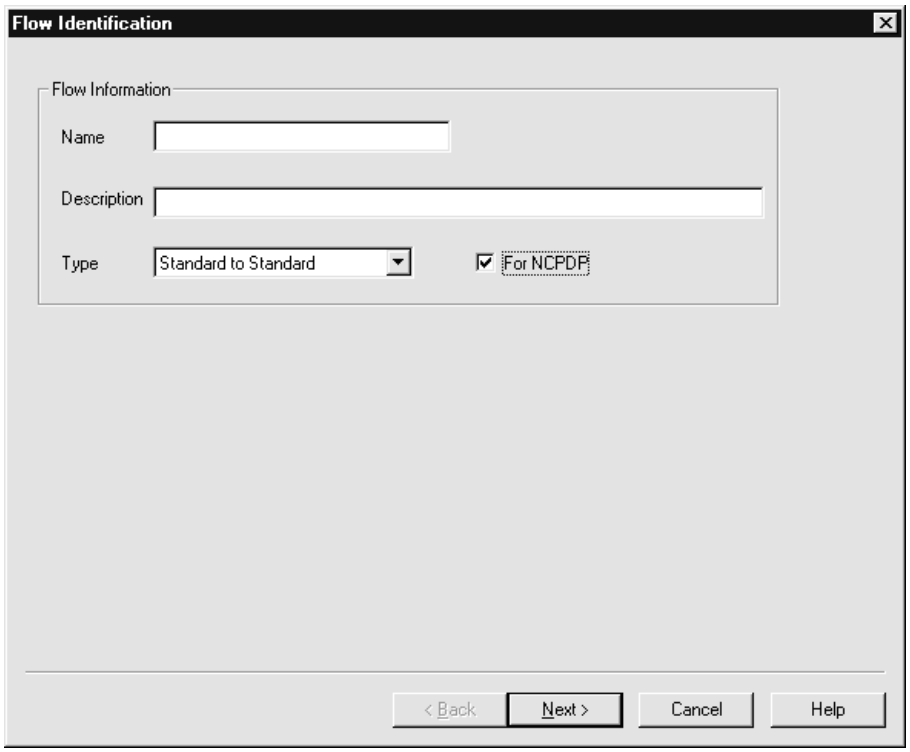
Flow Identification Dialog Box

Introduction

The Flow Identification dialog box is used to name and describe the process flow. This dialog box is the same for all flow types.

Flow Identification dialog box

This illustrations shows the Flow Identification dialog box.



Flow Identification fields and functions

This table describes the fields of the Flow Identification dialog box and their functions.

Field	Function
Name	Defines the name of the process flow. Maximum number of characters is 15. Do not use spaces in the name.
Description	Describes the process flow. Optional.
Type	Enables you to select type of process flow. ◗ Standard-to-standard ◗ Standard-to-application ◗ Application-to-standard ◗ Application-to-application **Note** If you have optional Sterling Gentran:Server products, such as the XML or SAP translation option you have additional flow types available.
For NCPDP	Enables you to create an Inbound NCPDP Flow. **Note** Select the check box only if the source file is in an NCPDP format.

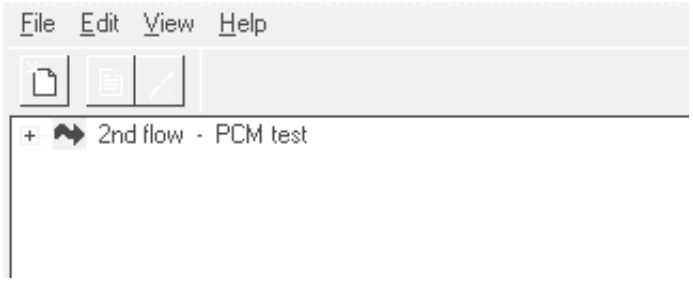
How to Name and Describe the Flow

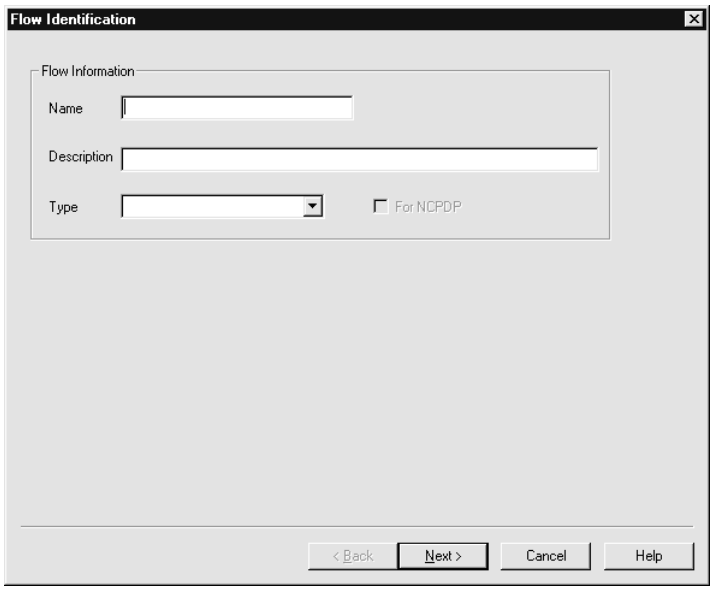
Introduction

The first step in creating a new process flow is to name the flow, describe it, and select the flow type.

Procedure

Use this procedure to name and describe the process flow and select the flow type.

Step	Action
1	Click the **PCM** button on the Sterling Gentran:Server client toolbar to start the Process Control Manager wizard. **System Response** Sterling Gentran:Server displays a tree that shows all the existing flows. This example has only one flow. 

(Contd) Step	Action
2	Click **New** on the File menu to start the flow creation wizard. **System Response** Sterling Gentran:Server displays the Flow Identification dialog box. 
3	Complete the boxes. **WARNING** **You must name the flow and select a flow type. The description is optional, but we recommend that you include it.** **You cannot change the flow's name after you create and save the flow.**
4	Click **Next** to continue to the Source setup dialog box. **Reference** See the appropriate section in this chapter: ◗ Creating an Inbound Flow ◗ Creating an Outbound Application Flow ◗ Creating an XML Flow ◗ Creating an Inbound NCPDP Flow.

Creating an Inbound Flow

Overview

Flow types

If the source document is in a standard format, the flow type is inbound. These are inbound flows:

- Standard-to-standard
- Standard-to-application.

Note
If you have optional Sterling Gentran:Server products, such as the XML translation option, you have additional flow types available.

Data managers in an inbound flow

An inbound flow has these three data managers:

- Source agent - Processes EDI data. Starts the movement of data in the flow.
- Processing agent - Starts a translation script that runs the translation program, **lftran**, with the -i (inbound) option.
- Delivery agent - Runs an embedded script. Primary role of this script is to generate a Life Cycle event record for auditing purposes. Can also run a script to perform any after-translation processing on the data.

Routing direction

This table describes the routing direction in an inbound flow.

Stage	Description
1	A source agent: ◗ Receives EDI files ◗ Splits EDI files by trading partner ◗ Processes data and routes EDI data to the processing agent.

(Contd) Stage	Description
2	The processing agent runs the translation program, **Iftran,** which translates the data and routes it to a delivery agent.
3	The delivery agent runs an embedded script named <*dmname*>*_gen_xltr.scr.* By default, this script does nothing, but the process generates a Life Cycle event record. If you have specified a post-processing script on the Delivery setup dialog box, the delivery agent runs the script.

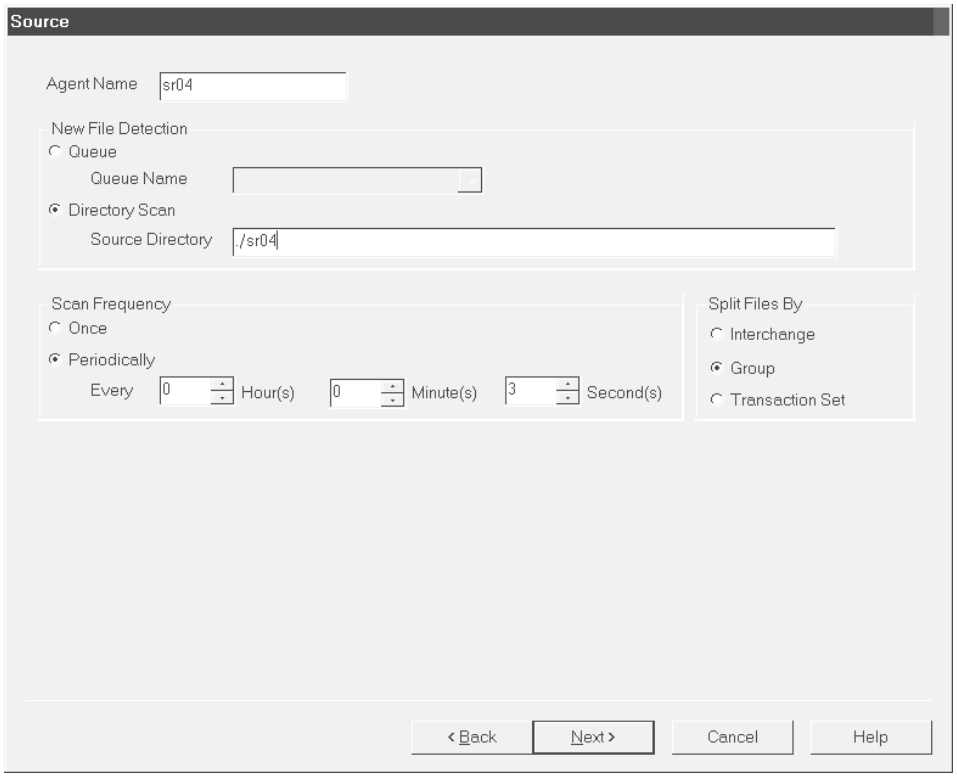
Source Setup Dialog Box (Inbound Flow)

Introduction

The **Source** setup dialog box for an inbound flow is used to create the inbound data manager that starts your process flow.

Illustration

This illustration shows the Source setup dialog box.



Source setup fields and functions

This table describes the fields of the Source setup dialog box and their functions.

Field	Function
Agent Name	Defines the name of the source data manager. The maximum size is 4 characters. **Note** The system supplies a default name, which is based on file type selected on the Flow Identification dialog box. You can override the default name.
New File Detection	
Queue	Specifies that a queue is the source type that the data manager looks in for new files to process and enables the Queue Name box so that you can select the name of the queue.
Queue Name	Enables you to select (from the drop-down list) the name of the queue that the data manager looks in for new files. The drop-down list contains the names of all the existing queues.
Directory Scan	Specifies that a scan directory is the source type that the data manager looks in for new files to process and enables the Source Directory box so that you can enter the name of the directory.
Source Directory	Enables you to type the name of the directory that the data manager looks in for new files. If you want the wizard to create the directory, type the name, using the relative path for EDI_ROOT. **Example** ./sr03 **Note** If you choose to use a nested directory structure, you must create the directory path first and then type it in the Source Directory box.
Scan Frequency	
Once	Selects one time as the scan frequency.
Periodically	Enables you to select the frequency with which you want the source data manager to scan its work directory.
Hour(s)	Defines, in hours, the frequency with which the data manager scans for new files. Value range is 0 to 23.

(Contd) Field	Function
Minute(s)	Defines, in minutes, the frequency with which the data manager scans for new files. Value range is 0 to 59.
Second(s)	Defines, in seconds, the frequency with which the data manager scans for new files. The default value is 3 seconds. Value range is 0 to 59.
Split Files By	
Interchange	Selects interchange code as the splitting method to route files.
Group	Selects group code as the splitting method to route files.
Transaction Set	Selects transaction set as the splitting method to route files.

How to Set Up the Source Agent (Inbound Flow)

Introduction

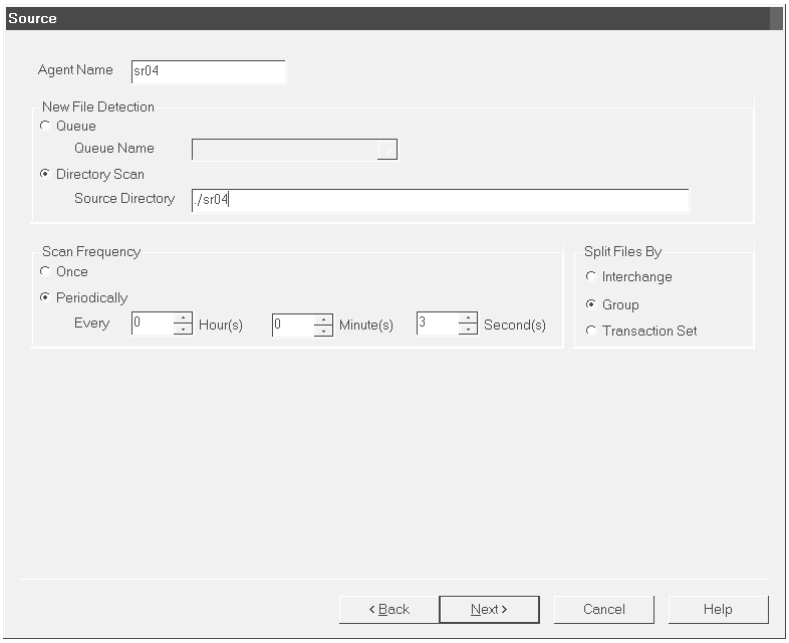
The **source agent** is the data manager that starts your process flow. In an inbound flow, the source agent is a data manager with an inbound personality.

Before you begin

You must complete the procedures in Beginning a Flow first.

Procedure

Use this procedure to set up the source data manager for an inbound flow.

Step	Action
1	Type the name of the source data manager in the Agent Name box. **CAUTION** **Sterling Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.** 

(Contd) Step	Action
2	Select either **Queue** or **Directory Scan** as the type of source that you want the source data manager to examine for files to process.
3	Did you select **Queue** in Step 2? ◗ If YES, select the name of the queue from the drop-down list and continue with Step 4. ◗ If NO, (the source is a directory), type the relative path name to the directory in the text box and continue with Step 4. **WARNING** **If a queue or directory is used by another data manager (source, processing or delivery agent), do not use it as the source for this data manager.**
4	Click **Once** or **Periodically** to select the scan frequency. **Note** The scan frequency you select applies to every data manager in the flow.
5	Did you select **Periodically** in Step 3? ◗ If YES, complete the Hour(s), Minute(s) and Second(s) boxes to select the frequency with which the data manager awakens and scans the queue or directory. ◗ If NO, continue with Step 5.
6	Click **Interchange**, **Group**, or **Transaction Set** to select how you want the data manager to group routed data.
7	Click **Next** to continue to the Processing Agent dialog box. **Reference** See How to Set Up the Processing Agent (Inbound Flow)

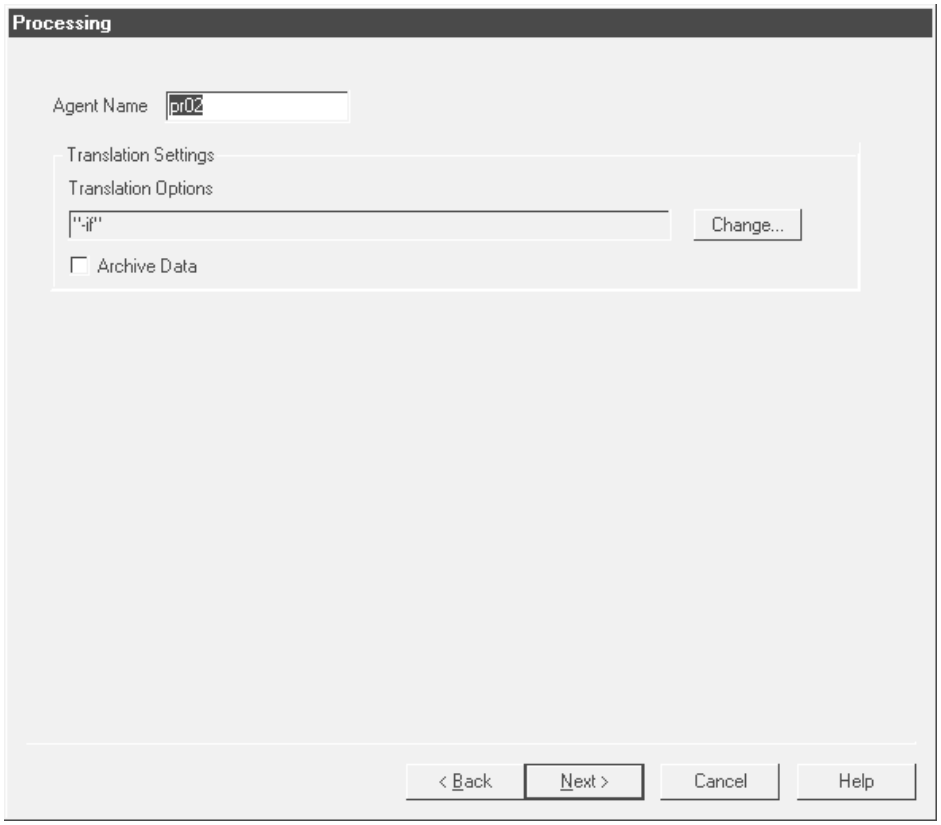
Processing Agent Dialog Box (Inbound Flow)

Introduction

The **processing agent** is the second data manager in a flow. In an inbound flow, its main function is to set translation options.

Processing Agent dialog box

This illustration shows the Processing Agent dialog box for an inbound flow.



Processing Agent fields and functions

This table describes the fields of the Processing Agent dialog box and their functions.

Field	Function
Agent Name	Defines the name of the processing data manager. **Note** The system supplies a default name, which is based on file type you selected on the Flow Identification dialog box. You can override the default name.
Translation Settings	
Translation Options	Displays the currently selected translation options.
Change	Displays the Translation Options dialog box. **Reference** See the Translation Options Dialog Box topic in this section.
Archive Data	Runs the **ediarc** program in the translation script. Archives the EDI-standard version of the file. **Reference** See the *ediarc* topic in the *Command Reference* chapter of the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide* for more information about **ediarc**. See the *Archiving Translation Data* chapter in the *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide* for information about archiving translation data.

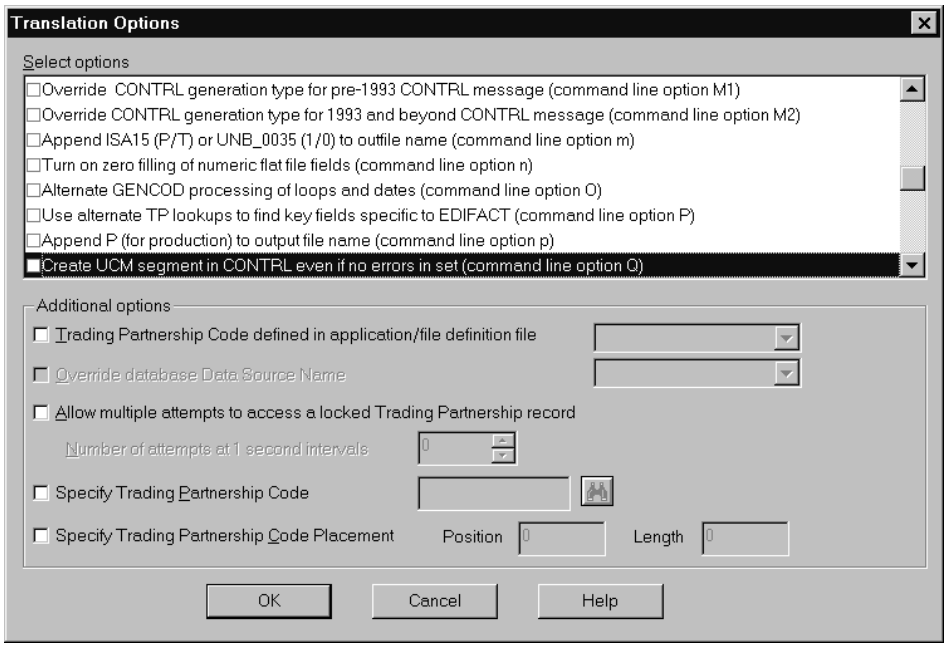
Translation Options Dialog Box

Introduction

Sterling Gentran:Server displays the Translation Options dialog box when you click the Translation Options **Change** button on the Processing Agent dialog box.

Illustration

This illustration shows the Translation Options dialog box.



Translation Option fields and functions

This table describes the fields of the Translation Options dialog box and their functions.

Field	Function
Select Options	Allows you to select parameters for the translation process. **Reference** See the table in the Select Options List topic in the online help system for a description of the available choices.
Trading Partnership Code defined in application/file definition file	Allows you to specify the file containing the Trading Partnership Code you want used for this translation. For outbound translation only.
Allow multiple attempts to access a locked Trading Partnership record	Sets the number of attempts Sterling Gentran:Server can make to access a locked Trading Partnership record. If the file is still locked after the specified number of attempts, the translation process fails.
Number of attempts at 1 second intervals	Sets the number of lock attempts.
Specify Trading Partnership Code	Specifies the Trading Partnership Code to be passed to the translator during translation. Use this option when the Trading Partnership Code is not present in the application data and the entire file can be translated using one Trading Partnership Code and one map. For outbound translation only.
Specify Trading Partnership Code placement	Specifies the location of the Trading Partnership Code within the input file. For outbound translation only.

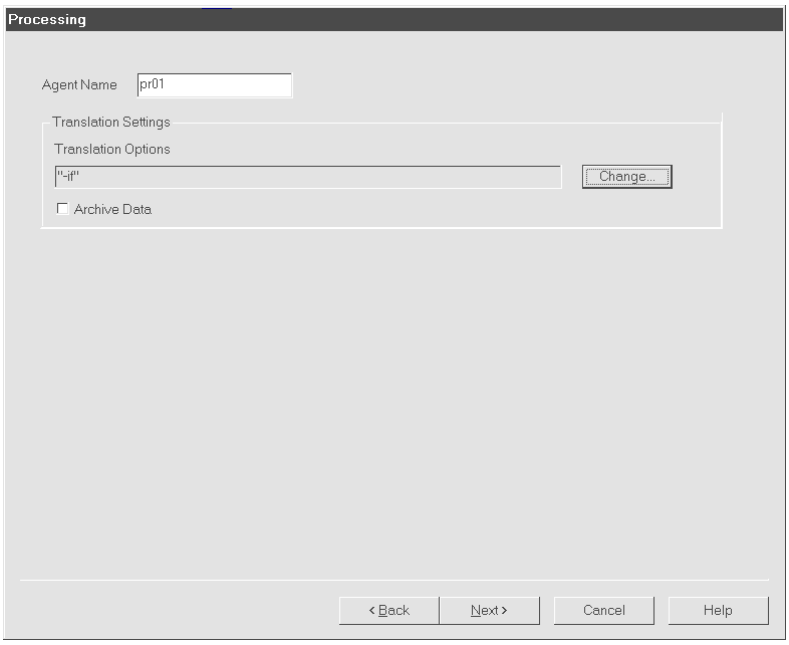
How to Set Up the Processing Agent (Inbound Flow)

Introduction

The **processing agent** in an inbound flow invokes the translator. In an inbound flow, the processing agent is a data manager with a translation (xltr) personality.

Procedure

Use this procedure to set up the processing data manager for an inbound flow.

Step	Action
1	Type the name of the processing data manager in the Agent Name text box. **CAUTION** **Sterling Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.** 
2	Do you want to change the translation options? ◗ If YES, click the **Change** button and complete the Translation Options dialog box. ◗ If NO, continue with Step 3.

(Contd) Step	Action
3	Do you want the translation script to run the **ediarc** program? ◗ If YES, select **Archive Data** and then click **Next** to continue to the Delivery Agent dialog box. ◗ If NO, click **Next** to continue to the Delivery Agent dialog box. **Note** The **ediarc** program archives translation data. **Reference** See How to Set Up the Delivery Agent (Inbound Flow).

Delivery Agent Dialog Box (Inbound Flow)

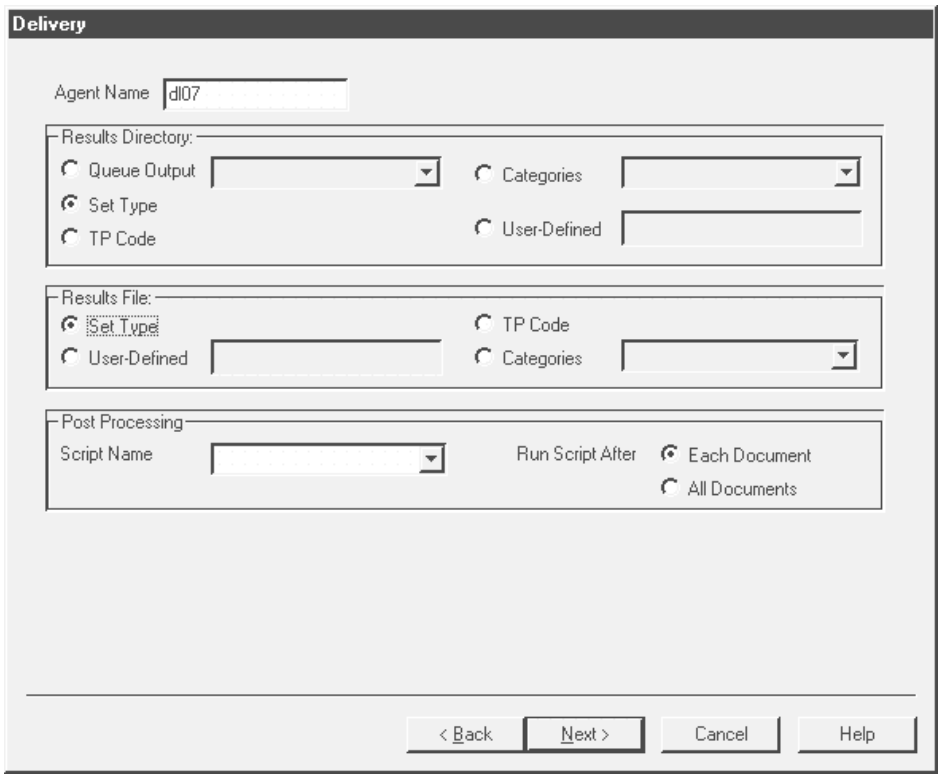
Introduction

The **delivery agent** is the third data manager in an inbound flow. Its function in an inbound flow is to:

- Designate the results (output) directory and file name
- Specify the name of the post processing script and when the script is run.

Delivery Agent dialog box

This illustration shows the Delivery agent dialog box for an inbound flow.



Delivery agent fields and functions

This table describes the fields of the Delivery agent dialog box and their functions.

Field	Function
Agent Name	Defines the name of the delivery data manager. **Note** The system supplies a default name, which is based on file type you selected on the Flow Identification dialog box. You can override the default name.
Results Directory	
Queue Output	Enables you to select (from the drop-down list) the name of a queue as the destination to which the delivery data manager directs the files it has processed. The drop-down list contains the names of all the existing queues.
Set Type	Selects transaction set type as the symbolic value for the Results Directory (destination directory) in the configuration records. The Process Control Manager substitutes the actual value for the type of transaction set in the configuration records. **Note** This option is disabled if the output document is in an XML format.
TP Code	Selects Trading Partnership Code as the Results Directory (destination directory) in the configuration records. The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.
Categories	Enables you to specify a Trading Partnership category as the Results Directory (destination directory) in the configuration records. Select the category from the drop-down list box that is next to the Categories option. The Process Control Manager substitutes the actual category value in the configuration records.

(Contd) Field	Function
User Defined	Enables you to specify the Results Directory (destination directory) for the configuration records. Enter the path in the text box that is next to the User Defined option.
Results File	
Set Type	Selects transaction set type as the symbolic value for the Results File (output file name) in the configuration records. The Process Control Manager substitutes the actual value for the type of transaction set in the configuration records. **Note** This option is disabled if the output document is in an XML format.
TP Code	Selects Trading Partnership Code as the Results File (output file name) in the configuration records. The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.
Categories	Enables you to specify a Trading Partnership category as the Results File (output file name) in the configuration records. Select the category from the drop-down list box that is next to the Categories option. The Process Control Manager substitutes the actual category value in the configuration records.
Post Processing	
User Defined	Enables you to specify the Results File (output file name) for the configuration records. Enter the path in the text box that is next to the User Defined option.
Script Name	Enables you to enter or select the name of the script you want to run after this data manager has processed the files.

(Contd) Field	Function
Each Document	Executes the post-processing Sterling Gentran:Server script after each document has been processed.
All Documents	Executes the post-processing Sterling Gentran:Server script after all documents have been processed.

How to Set Up the Delivery Agent (Inbound Flow)

Introduction

The **delivery agent** is the destination data manager in a process flow. In an inbound flow, the delivery agent is a data manager with a translation (xltr) personality.

Setting configuration record information

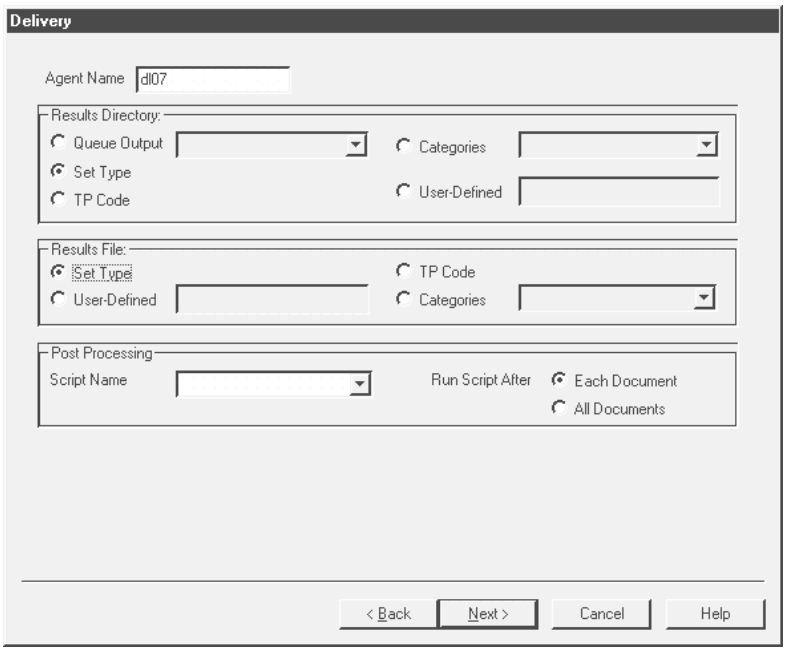
The Delivery Agent dialog box enables you to set information that the Process Control Manager uses in the Trading Partnership configuration records it creates.

You can set:

- Exact destination directory and file name information that you want the Process Control Manager to use in every configuration record that it creates from the flow
- Symbolic destination directory and file name values, such as a category or Trading Partnership Code. The Process Control Manager substitutes the actual value for the symbolic value in the configuration records
- The name of the script (if any) Sterling Gentran:Server runs after processing the Trading Partner's files. You also select whether the script runs after each document is processed or after all documents are processed.

Procedure

Use this procedure to set up the delivery agent for an inbound flow.

Step	Action
1	Type the name of the delivery data manager in the **Agent Name** box. **CAUTION** **Sterling Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.** 
2	Choose the Results Directory by clicking **Set Type**, **TP Code**, **Categories**, or **User-Defined** to select the symbolic destination directory, or typing the path to the directory for the output. **Comment** The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. The Process Control Manager creates directories if they do not exist. **Example** If you select **TP Code**, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.

(Contd) Step	Action
3	Did you select **Categories** in Step 2? ◗ If YES, select a category from the drop-down list. ◗ If NO, continue with Step 4.
4	Did you select **User-Defined** in Step 2? ◗ If YES, type the path to the directory in the text box that is below the User-Defined option. ◗ If NO, continue with Step 5.
5	Choose the Results File name by clicking **Set Type**, **TP Code**, **Categories**, or **User-Defined** to select the symbolic destination file name. **Comment** The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. **Example** If you select **TP Code**, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.
6	Did you select **Categories** in Step 5? ◗ If YES, select a category from the drop-down list and continue with Step 7. ◗ If NO, continue with Step 7.
7	Did you select **User-Defined** in Step 5? ◗ If YES, type the complete file name in the text box that is below the **User-Defined** option and continue with Step 8. ◗ If NO, continue with Step 8.
8	Do you want to execute a script after the translation process? ◗ If YES, select the name of the script from the **Script Name** drop-down list and continue with Step 9. ◗ If NO, continue with Step 9.
9	Click the **Each document** or **All documents** option to select when the system runs the script.
10	Click **Next** to continue to the Error Handling dialog box. **Reference** See How to Set Up Error Handling Instructions for instructions on completing the Error Handling dialog box.

Creating an Outbound Application Flow

Overview

Flow types

These are the possible flow types for outbound application flows:

- Application-to-standard
- Application-to-application.

Note
If you have optional Sterling Gentran:Server products, such as the XML translation option or SAP, you have additional flow types available. If the source document is in an application format, the flow type is outbound.

Input file names

The input files for an outbound application flow must be named for one of the following:

- Trading Partnership code
- Application file that the data represents.

Agents in an outbound application flow

An outbound application flow has three agents:

- Source agent - Processes application data. Starts the movement of data in the flow.
- Processing agent - Starts a translation script that runs the translator with the -o (outbound) option.
- Delivery agent - Runs an embedded script that does nothing. The purpose of running the script is to generate a Life Cycle event record for auditing purposes. Can also run a script to perform any after-translation processing on the data.

Routing direction

This table describes the routing direction in an outbound application flow.

Stage	Description
1	The source agent, which is an application data manager, receives application data and routes it to the processing agent.
2	The processing agent, which is a translation data manager, runs the translator. After translation, the flow routes the translated data to the delivery agent.
3	The delivery agent, which is a translation data manager, runs an imbedded script that does nothing. The purpose of running the script is to generate a Life Cycle event record for auditing purposes. The name of the script is <*dmname*>*_gen_xltr.scr*. If you specified a post-processing script on the Delivery setup dialog box, the delivery agent runs the script.

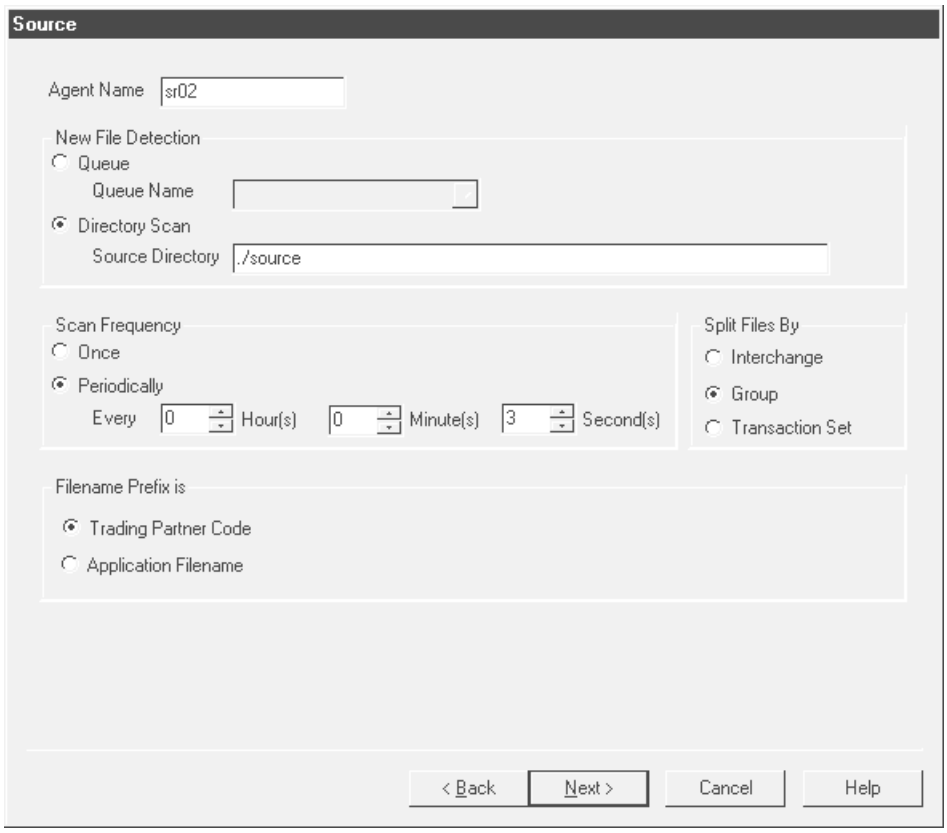
Source Setup Dialog Box (Outbound Application)

Introduction

The **Source** setup dialog box for an outbound application flow is used to create the application (appm personality) data manager that starts your process flow.

Illustration

This illustration shows the Source setup dialog box.



Source setup fields and functions

This table describes the fields of the Source setup dialog box and their functions.

Field	Function
Agent Name	Defines the name of the source data manager. **Note** The system supplies a default name, which is based on file type selected on the Flow Identification dialog box. You can override the default name.
New File Detection	
Queue	Selects queue as the source type that the data manager looks in for new files to process.
Queue Name	Enables you to select (from the drop-down list) the name of the queue that the data manager looks in for new files. The drop-down list contains the names of all the existing queues.
Directory Scan	Selects a scan directory as the source type that the data manager looks in for new files to process.
Source Directory	Enables you to type or select the name of the directory that the data manager looks in for new files. If you choose to type the name, use the relative path for EDI_ROOT.
Scan Frequency	
Once	Selects one time as the scan frequency for every data manager in the flow.
Periodically	Enables you to select the frequency with which you want the source data manager to scan its work directory.
Hour(s)	Defines, in hours, the frequency with which the data managers in the flow scan for new files. Value range is 0 to 23.
Minute(s)	Defines, in minutes, the frequency with which the data managers in the flow scan for new files. Value range is 0 to 59.

(Contd) Field	Function
Second(s)	Defines, in seconds, the frequency with which the data managers in the flow scan for new files. Value range is 0 to 59. The default value is 3 seconds.
Split Files By	
Interchange	Selects interchange code as the splitting method to route files.
Group	Selects group code as the splitting method to route files.
Transaction Set	Selects transaction set as the splitting method to route files.
Filename Prefix is	
Trading Partner Code	Indicates that, if the input file name prefix exists, then the Trading Partnership code is the prefix. If the prefix does not exist, then the Trading Partnership code is the entire file name. Used in outbound flows.
Application Filename	Indicates that, if the file name prefix exists, then the application description file name is the prefix. Otherwise, the application description file name is the entire file name. Used in outbound flows.

How to Set Up the Source Agent (Outbound Application)

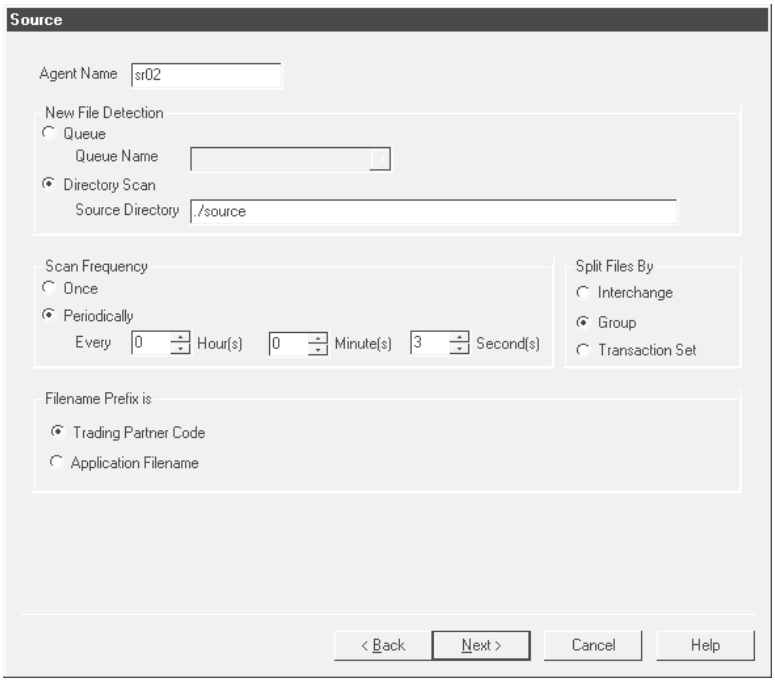
Introduction

The **source agent** is the data manager that starts your process flow.

In an outbound application flow, the source agent is a data manager with an application (appm) personality. It receives application files and splits them for routing by interchange, group, or transaction set.

Procedure

Use this procedure to set up the source agent.

Step	Action
1	Type the name of the source data manager in the Agent Name dialog box. **CAUTION** **Sterling Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.** 

(Contd) Step	Action
2	Select either **Queue** or **Directory Scan** as the type of source that you want the source data manager to examines for files to process. If the source is a queue, select the name of the queue from the drop-down list. If the source is a directory, type the relative path name to the directory in the text box.
3	Click **Once** or **Periodically** to select the scan frequency. **Note** The scan frequency you select applies to every data manager in the flow.
4	Did you select **Periodically** in Step 3? ◗ If YES, complete the **Hour(s**), **Minute(s)** and **Second(s)** boxes to select the frequency with which the data manager awakens and scans the queue or directory. ◗ If NO, continue with Step 5.
5	Click **Interchange**, **Group**, or **Transaction Set** to select how the data manager groups routed data.
6	Select the filename prefix: **Trading Partner Code** or **Application Filename**.
7	Click **Next** to continue to the Processing Agent dialog box. **Reference** See the How to Set Up the Processing Agent (Outbound Application).

Processing Agent Dialog Box (Outbound Application)

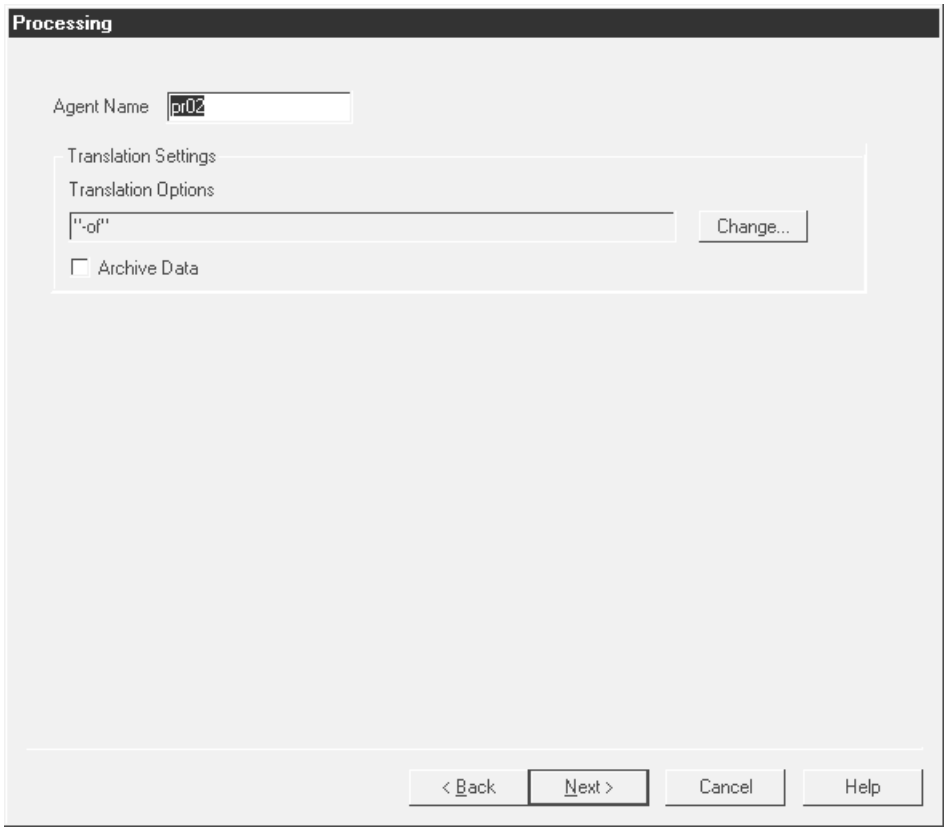
Introduction

The **processing agent** is the second data manager in a flow. In an outbound application flow, its function is to:

- Specify translation settings
- Run **ediarc** in the translation script to archive translation data. This is optional.

Processing Agent dialog box

This illustration shows the Processing Agent dialog box.



Processing Agent fields and functions

This table describes the fields of the Processing Agent dialog box in an outbound application flow and their functions.

Field	Function
Agent Name	Defines the name of the processing data manager. **Note** The system supplies a default name, which is based on file type you selected on the Flow Identification dialog box. You can override the default name.
Translation Settings	
Translation Options	Displays the currently selected translation options.
Change	Displays the Translation Options dialog box. **Reference** See the Translation Options Dialog Box (Outbound Application) topic in this section.
Archive Data	Runs the **ediarc** program in the translation script. Archives the EDI-standard version of the file. **Reference** See the *ediarc* topic in the *Command Reference* chapter of the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide* for more information about **ediarc**. See the *Archiving Translation Data* chapter in the *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide* for information about archiving translation data.

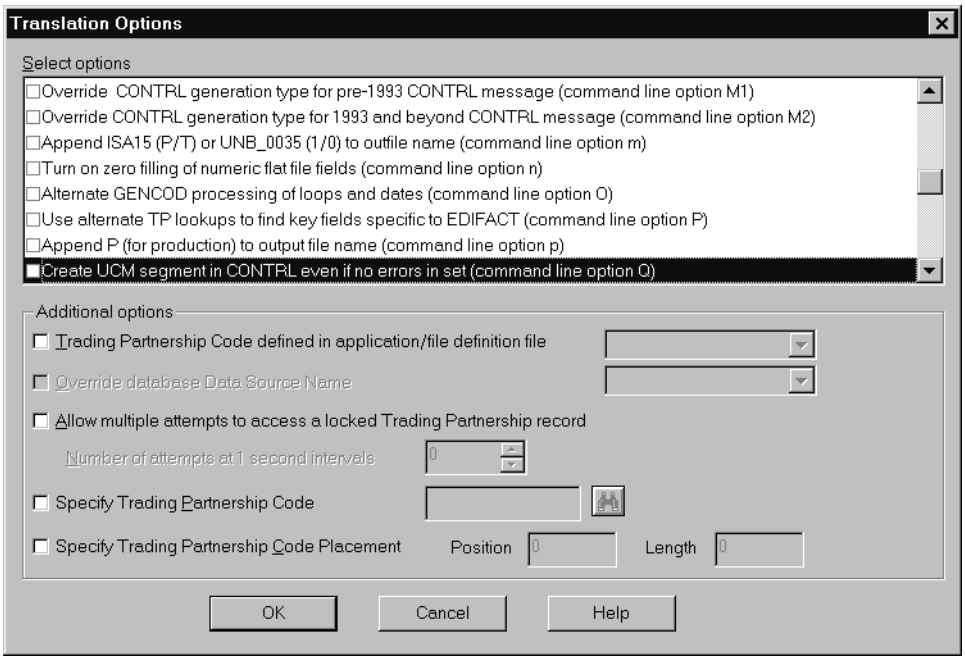
Translation Options Dialog Box (Outbound Application)

Introduction

Sterling Gentran:Server displays the Translation Options dialog box when you click the Translation Options **Change** button on the Processing Agent dialog box.

Illustration

This illustration shows the Translation Options dialog box.



Translation Option fields and functions

This table describes the fields of the Translation Options dialog box and their functions.

Field	Function
Select options	Enables you to select the translation options you want to apply to this flow. **Reference** For a list of translation options, see the *Iftran Overview* topic in the *Command Reference* chapter of the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide*.
Trading Partnership code defined in application/file definition file	Enables you to select the application or file definition file. Used only for outbound translations.
Allow multiple attempts to access a locked Trading Partnership record	Allows the data manager to attempt more than one time to access a locked Trading Partnership record.
Number of attempts at 1 second intervals	Enables you to specify the number of times the data manager should attempt to access a locked Trading Partnership record before translation fails.
Specify Trading Partnership Code	Enables you to search for the Trading Partnership code that you want to use to override Trading Partnership data. Used only for outbound translations.
Specify Trading Partnership Code Placement	Enables you to specify the Trading Partnership code's position in the file and the length of the of the code. Used only for outbound translations.

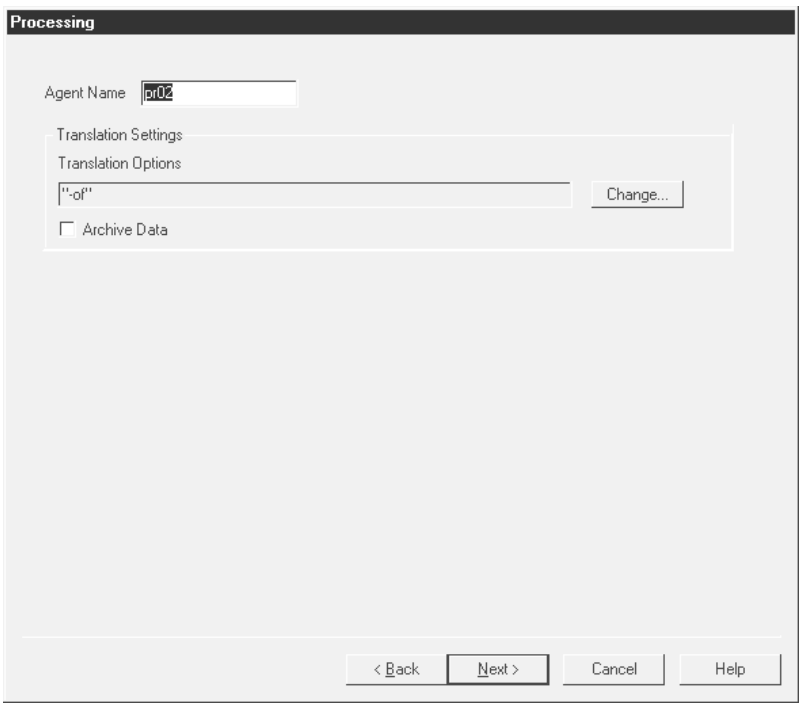
How to Set Up the Processing Agent (Outbound Application)

Introduction

The **processing agent** is the second data manager in a process flow. In an outbound application flow, the processing agent is a data manager with a translation (xltr) personality. It invokes the translator and runs translation with the -o option.

Procedure

Use this procedure to set up the processing agent for an outbound application flow.

Step	Action
1	Type the name of the processing data manager in the Agent Name box. 
2	Do you want to change the translation options? ◗ If YES, click the **Change** button and complete the Translation Options dialog box. ◗ If NO, continue with Step 3.

(Contd) Step	Action
3	Do you want the translation script to run the **ediarc** program? ◗ If YES, select **Archive Data** and then click **Next** to continue to the Delivery Agent dialog box. ◗ If NO, click **Next** to continue to the Delivery Agent dialog box. **Note** The **ediarc** program archives translation data. **Reference** See the How to Set Up the Delivery Agent (Outbound Application).

Delivery Agent Dialog Box (Outbound Application)

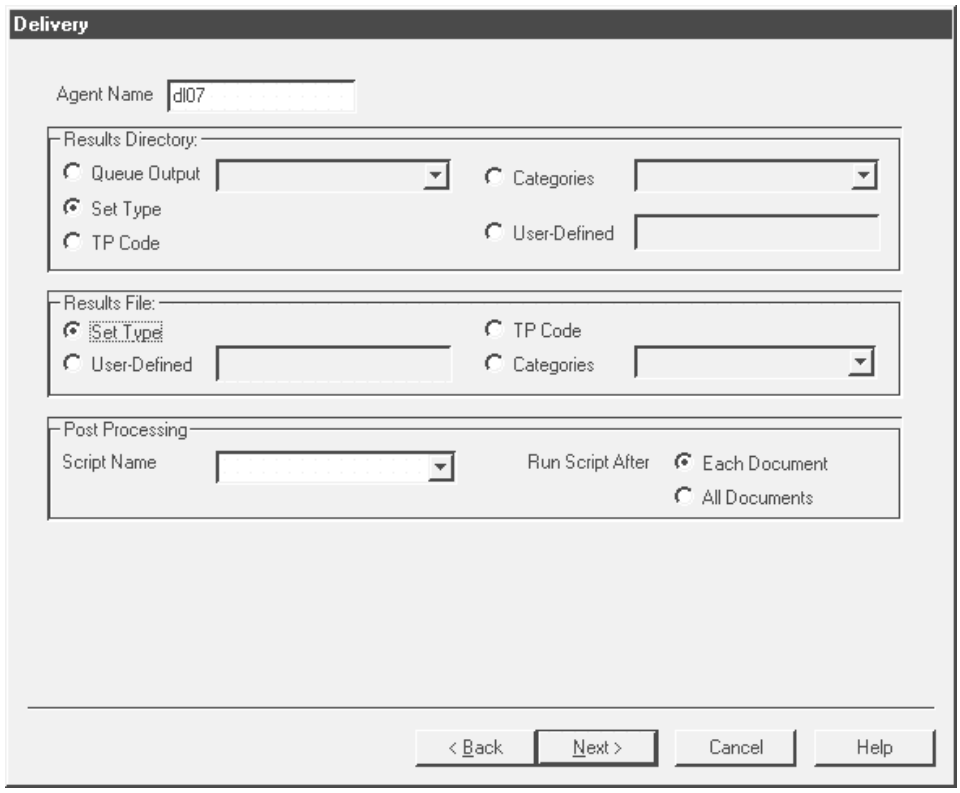
Introduction

The **delivery agent** is the third data manager in an flow. Its function in an outbound application flow is to:

- Designate the results (output) directory and file name
- Specify the name of the post processing script (if any) and select when the script is run.

Delivery agent dialog box

This illustration shows the Delivery agent dialog box.



Delivery Agent fields and functions

This table describes the fields of the **Delivery Agent** dialog box and their functions..

Field	Function
Agent Name	Defines the name of the delivery data manager. **Note** The system supplies a default name, which is based on file type you selected on the Flow Identification dialog box. You can override the default name.
Results Directory	
Queue Output	Enables you to select (from the drop-down list) the name of a queue as the destination to which the delivery data manager lists the files it has processed. The drop-down list contains the names of all the existing queues.
Set Type	Selects transaction set type as the symbolic value for the Results Directory (destination directory) in the configuration records. The Process Control Manager substitutes the actual value for the type of transaction set in the configuration records. **Note** This option is disabled if the output document is in an XML format.
TP Code	Selects Trading Partnership Code as the Results Directory (destination directory) in the configuration records. The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.
Categories	Enables you to specify a Trading Partnership category as the Results Directory (destination directory) in the configuration records. Select the category from the drop-down list box that is next to the Categories option. The Process Control Manager substitutes the actual category value in the configuration records.

(Contd) Field	Function
User Defined	Enables you to specify the Results Directory (destination directory) for the configuration records. Enter the path in the text box that is next to the User Defined option.
Results File	
Set Type	Selects transaction set type as the symbolic value for the Results File (output file name) in the configuration records. The Process Control Manager substitutes the actual value for the type of transaction set in the configuration records. **Note** This option is disabled if the output document is in an XML format.
TP Code	Selects Trading Partnership Code as the Results File (output file name) in the configuration records. The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.
Categories	Enables you to specify a Trading Partnership category as the Results File (output file name) in the configuration records. Select the category from the drop-down list box that is next to the Categories option. The Process Control Manager substitutes the actual category value in the configuration records.
User Defined	Enables you to specify the Results File (output file name) for the configuration records. Enter the path in the text box that is next to the User Defined option.
Post Processing	
Script Name	Enables you to enter or select the name of the script you want to run after this data manager has processed the files.

(Contd) Field	Function
Each Document	Executes the post-processing Sterling Gentran:Server script after each document has been processed.
All Documents	Executes the post-processing Sterling Gentran:Server script after all documents have been processed.

How to Set Up the Delivery Agent (Outbound Application)

Introduction

The **delivery agent** is the destination data manager in a process flow. In an outbound application flow, the delivery agent is a data manager with a translation (*xltr*) personality.

Setting configuration record information

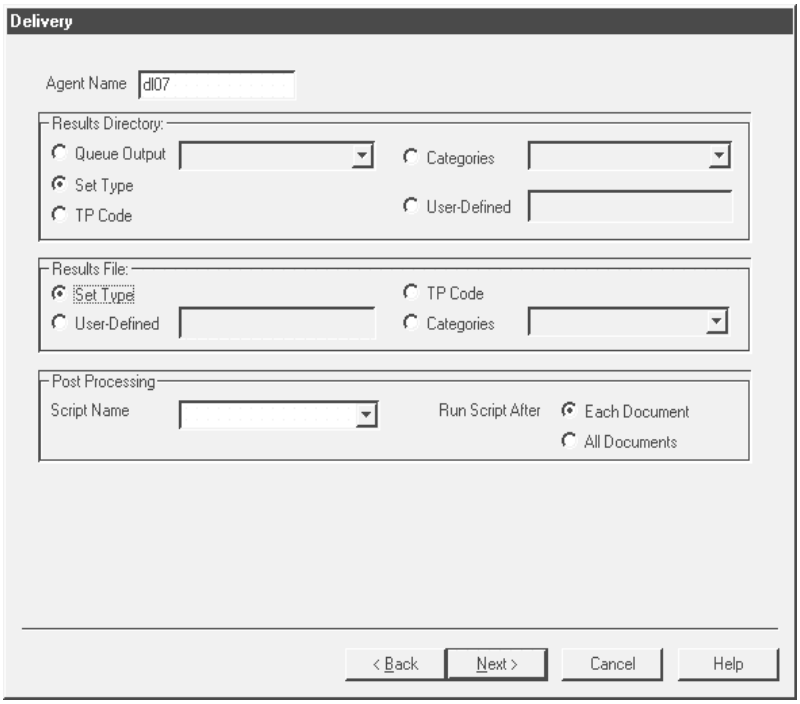
The Delivery agent dialog box enables you to set information that the Process Control Manager uses in the Trading Partnership configuration records it creates.

You can set:

- Exact destination directory and file name information that you want the Process Control Manager to use in every configuration record that it creates from the flow
- Symbolic destination directory and file name values, such as a category or Trading Partnership Code. The Process Control Manager substitutes the actual value for the symbolic value in the configuration records
- The name of the script Sterling Gentran:Server runs after processing the Trading Partner's files. You also select whether the script runs after each document is processed or after all documents are processed.

Procedure

Use this procedure to set up the delivery data manager for an outbound application flow.

Step	Action
1	Type the name of the delivery data manager in the Agent Name box. 
2	Choose the Results Directory by clicking **Set Type**, **TP Code**, **Categories**, or **User-Defined** to select the symbolic destination directory, or type the path to the directory for the translation output. **Comment** The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. **Example** If you select **TP Code**, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.
3	Did you select **Categories** in Step 2? ◗ If YES, select a category from the drop-down list. ◗ If NO, continue with Step 4.

(Contd) Step	Action
4	Did you select **User-Defined** in Step 2? ◗ If YES, type the path to the directory in the text box that is below the **User-Defined** option. ◗ If NO, continue with Step 5.
5	Choose the Results File name by clicking **Set Type**, **TP Code**, **Categories**, or **User-Defined** to select the symbolic destination file name. **Comment** The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. **Example** If you select TP Code, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.
6	Did you select **Categories** in Step 5? ◗ If YES, select a category from the drop-down list and continue with Step 7. ◗ If NO, continue with Step 7.
7	Did you select **User-Defined** in Step 5? ◗ If YES, type the complete file name in the text box that is below the **User-Defined** option and continue with Step 8. ◗ If NO, continue with Step 8.
8	Do you want to run a post-processing script? ◗ If YES, select the name of the script from the Script Name drop-down list and continue with Step 9. ◗ If NO, continue with Step 9.
9	Click the **Each document** or **All documents** option to select when the system runs the script.
10	Click **Next** to continue to the Error Handling dialog box. **Reference** See How to Set Up Error Handling Instructions for instructions on completing the Error Handling dialog box.

Creating an XML Flow

Overview

Introduction

If you have the XML translation option, you can create XML flows with the PCM wizard.

This section describes XML flows when XML format is the source format. If the XML format is the destination format only, see these references:

References

- If the flow type is application-to-XML, see Creating an Outbound Application Flow for instructions on creating the flow.
- If the flow type is standard-to-XML, see Creating an Inbound Flow for instructions on creating the flow.

Flow types

These are the flow types covered in this section:

- XML-to-application
- XML-to-standard
- XML-to-XML

Input file names

The input files for an XML flow must be named for one of the following:

- File definition that the data represents.

Agents in an XML flow

An XML flow has three agents:

- Source agent - Processes XML data. Starts the movement of data in the flow.
- Processing agent - Starts a translation script.
- Delivery agent - Runs a script to generate a Life Cycle event record for auditing purposes. Can also run a script to perform any after-translation processing on the data.

Routing

This table describes the routing in an XML flow.

Stage	Description
1	The source agent, which is an XML data manager, receives XML data and routes it to the processing agent.
2	The processing agent, which is a translation data manager, runs the translator. After translation, the flow routes the translated data to the delivery agent.
3	The delivery agent, which is a translation data manager, runs an imbedded script named *<dmname>_gen_xltr.scr* that runs to generate a Life Cycle event record for auditing purposes. **Note** If you specified a post-processing script on the Delivery setup dialog box, the delivery agent runs the script.

Source Setup Dialog Box (XML)

Introduction

The **Source** setup dialog box for an XML flow is used to create the XML (*xmli* or *xmlo* personality) data manager that starts your process flow.

Source setup fields and functions

This table describes the fields of the Source setup dialog box for XML flows and their functions.

Field	Function
Agent Name	Defines the name of the source data manager for the XML flow. **Note** The system supplies a default name, which is based on file type selected on the Flow Identification dialog box. You can override the default name.
New File Detection	
Queue	Selects queue as the source type that the data manager looks in for new files to process.
Queue Name	Enables you to select (from the drop-down list) the name of the queue that the data manager looks in for new files. The drop-down list contains the names of all the existing queues.
Directory Scan	Selects a scan directory as the source type that the data manager looks in for new files to process.
Source Directory	Enables you to type or select the name of the directory that the data manager looks in for new files. If you choose to type the name, use the relative path for EDI_ROOT.
Scan Frequency	
Once	Selects one time as the scan frequency for every data manager in the flow.
Periodically	Enables you to select the frequency with which you want the source data manager to scan its work directory.

(Contd) Field	Function
Hour(s)	Defines, in hours, the frequency with which the data managers in the flow scan for new files. Value range is 0 to 23.
Minute(s)	Defines, in minutes, the frequency with which the data managers in the flow scan for new files. Value range is 0 to 59.
Second(s)	Defines, in seconds, the frequency with which the data managers in the flow scan for new files. Value range is 0 to 59. The default value is 3 seconds.
Trading Partner Code	Not used in XML flows.
Application Filename	Not used in XML flows.

How to Set Up the Source Agent (XML Flow)

Introduction

The **source agent** is the data manager that starts your process flow.

In an XML flow, the source agent is a data manager with an XML (xlmo, xlmi) personality. It receives XML files and splits them for routing according to the XML splitter table.

Procedure

Use this procedure to set up the source agent.

Step	Action
1	Type the name of the source data manager in the Agent Name box. **CAUTION** **Sterling Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.**
2	Do you want the data manager to scan a queue for files? ◗ If YES, click **Queue** and then select the name of the queue from the drop-down list. ◗ If NO, (you want the data manager to scan a directory), click **Directory Scan** and then type the relative path name to the directory in the text box.
3	Click **Once** or **Periodically** to select the scan frequency. **Note** The scan frequency you select applies to every data manager in the flow.
4	Did you select **Periodically** in Step 3? ◗ If YES, complete the **Hour(s**), **Minute(s)** and **Second(s)** boxes to select the frequency with which the data manager awakens and scans the queue or directory. ◗ If NO, continue with Step 5.
5	Click **Next** to continue to the Processing Agent dialog box. **Reference** See How to Set Up the Processing Agent (XML Flow).

Processing Agent Dialog Box (XML Flow)

Introduction

The **processing agent** is the second data manager in a flow. In an XML flow, its function is to specify translation settings.

Processing Agent fields and functions

This table describes the fields of the Processing Agent dialog box in an XML flow and their functions.

Field	Function
Agent Name	Defines the name of the processing data manager. **Note** The system supplies a default name, which is based on file type you selected on the Flow Identification dialog box. You can override the default name.
Translation Options	Displays the currently selected translation options. **Note** When a flow has XML format as the source or destination format, the default value is -yf.
Change	Displays the Translation Options dialog box. **Reference** See the Translation Options Dialog Box (XML Flow) topic in this section.
Archive Data	If the output file is an EDI standard, runs the **ediarc** program in the translation script to archives the EDI-standard version. **Reference** See the *ediarc* topic in the *Command Reference* chapter of the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide* for more information about **ediarc**. See the *Archiving Translation Data* chapter in the *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide* for information about archiving translation data.

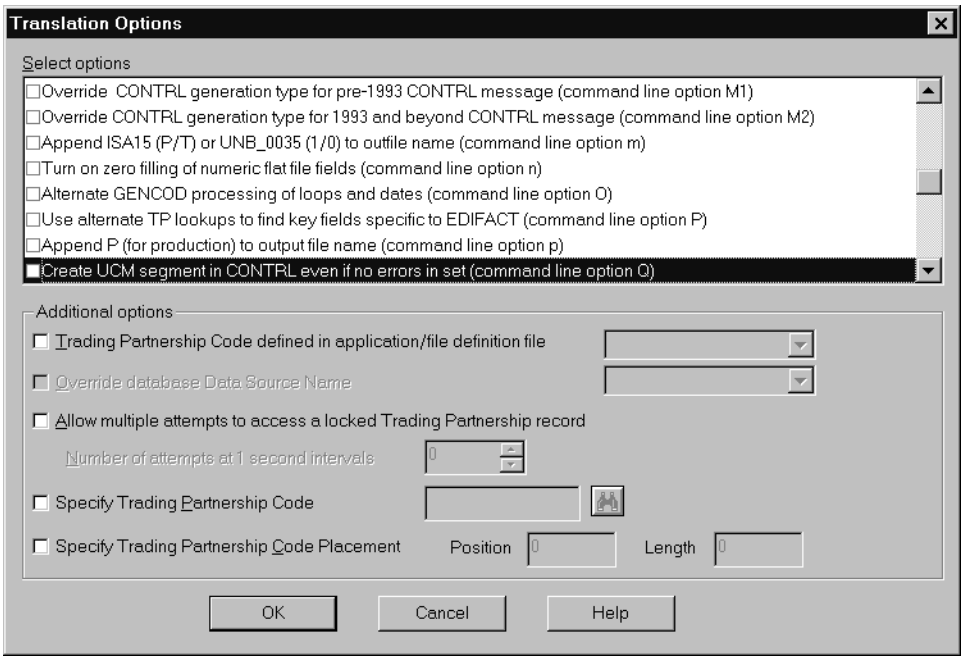
Translation Options Dialog Box (XML Flow)

Introduction

Sterling Gentran:Server displays the Translation Options dialog box when you click the Translation Options **Change** button on the Processing agent dialog box.

Illustration

This illustration shows the Translation Options dialog box.



Translation Option fields and functions

This table describes the fields of the Translation Options dialog box and their functions.

Use this field	To
Select options	Select the translation options you want to apply to this flow. **Reference** For a list of translation options, see the *Iftran Overview* topic in the *Command Reference* chapter of the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide*.
Trading Partnership code defined in application/file definition file	Select the application or file definition file. Used only for outbound translations.
Allow multiple attempts to access a locked Trading Partnership record	Allow the data manager to attempt more than one time to access a locked Trading Partnership record.
Number of attempts at 1 second intervals	Specify the number of times the data manager should attempt to access a locked Trading Partnership record before translation fails.
Specify Trading Partnership Code	Search for the Trading Partnership code that you want to use to override Trading Partnership data. Used only for outbound translations.
Specify Trading Partnership Code Placement	Specify the Trading Partnership code's position in the file and the length of the of the code. Used only for outbound translations.

How to Set Up the Processing Agent (XML Flow)

Introduction

The **processing agent** is the second data manager in a process flow. In an XML-to-application or XML-to-standard flow, the processing agent is a data manager with a translation (xltr) personality. It invokes the translator and runs translation.

Procedure

Use this procedure to set up the processing agent for an XML flow.

Step	Action
1	Type the name of the processing data manager in the Agent Name box.
2	Do you want to change the translation options? ◗ If YES, click the **Change** button and complete the Translation Options dialog box. ◗ If NO, continue with Step 3. **Note** We recommend that you do not change the translation options.
3	If the output file is in an EDI standard format, do you want the translation script to run the **ediarc** program? ◗ If YES, select **Archive Data** and then click **Next** to continue to the Delivery agent dialog box. ◗ If NO, click **Next** to continue to the Delivery agent dialog box. **Note** The **ediarc** program archives EDI translation data.
4	Continue with How to Set Up the Delivery Agent (XML Flow).

Delivery Agent Dialog Box (XML Flow)

Introduction

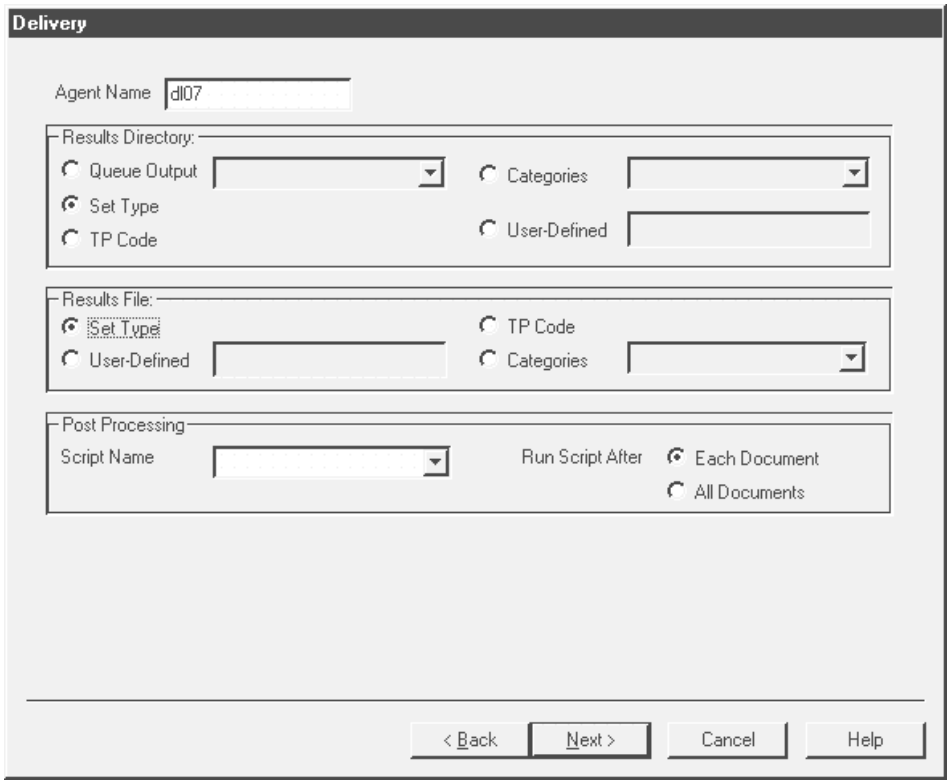
The **delivery agent** is the third data manager in an flow. Its function in an XML flow is to:

- Designate the results (output) directory and file name
- Specify the name of the post processing script (if any) and select when the script is run.

Note
There are two versions of this dialog box. If XML is the destination format, the Set Type option for the Results Directory and Results File does not appear on the dialog box.

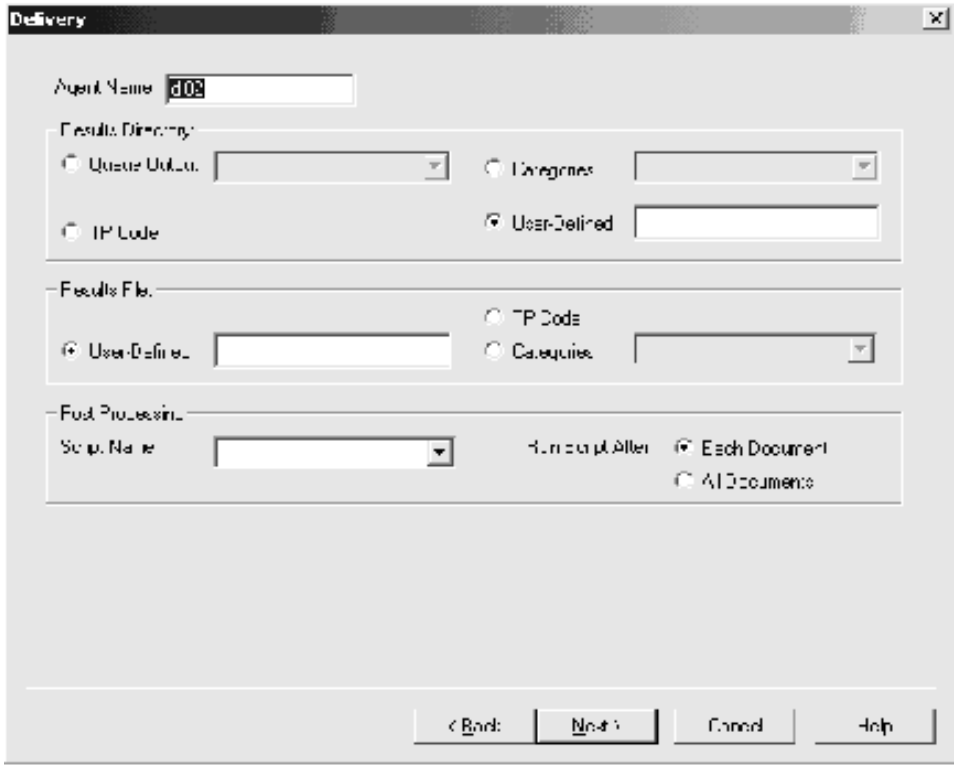
Delivery agent dialog box (1)

This illustration shows the Delivery agent dialog box for XML-to-application and XML-to-standard flows.



Delivery agent dialog box (2)

This illustration shows the Delivery agent dialog box for standard-to-XML, application-to-XML, and XML-to-XML flows.



Delivery Agent fields and functions

This table describes the fields of the Delivery Agent dialog box and their functions..

Use this field	To
Agent Name	Define the name of the delivery data manager. **Note** The system supplies a default name, which is based on file type you selected on the Flow Identification dialog box. You can override the default name.
Queue Output	Select (from the drop-down list) the name of a queue as the destination to which the delivery data manager directs the files it has processed. The drop-down list contains the names of all the existing queues.

(Contd) Use this field	To
Set Type	Select transaction set type as the symbolic value for the Results Directory (destination directory) in the configuration records. The Process Control Manager substitutes the actual value for the type of transaction set in the configuration records. **Note** This option is disabled and does not appear on the dialog box if the output document is in an XML format.
TP Code	Select Trading Partnership Code as the Results Directory (destination directory) in the configuration records. The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.
Categories	Specify a Trading Partnership category as the Results Directory (destination directory) in the configuration records. Select the category from the drop-down list box that is next to the Categories option. The Process Control Manager substitutes the actual category value in the configuration records.
User Defined	Specify the Results Directory (destination directory) for the configuration records. Enter the path in the text box that is next to the User Defined option.
Set Type	Select transaction set type as the symbolic value for the Results File (output file name) in the configuration records. The Process Control Manager substitutes the actual value for the type of transaction set in the configuration records. **Note** This option is disabled and does not appear on the dialog box if the output document is in an XML format.

(Contd) Use this field	To
TP Code	Select Trading Partnership Code as the Results File (output file name) in the configuration records. The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.
Categories	Specify a Trading Partnership category as the Results File (output file name) in the configuration records. Select the category from the drop-down list box that is next to the Categories option. The Process Control Manager substitutes the actual category value in the configuration records.
User Defined	Specify the Results File (output file name) for the configuration records. Enter the path in the text box that is next to the User Defined option.
Script Name	Enter or select the name of the script you want to run after this data manager has processed the files.
Each Document	Execute the post-processing Sterling Gentran:Server script after each document has been processed.
All Documents	Execute the post-processing Sterling Gentran:Server script after all documents have been processed.

How to Set Up the Delivery Agent (XML Flow)

Introduction

The **delivery agent** is the destination data manager in a process flow. In an XML flow, the delivery agent is a data manager with a translation (xltr) personality.

Setting configuration record information

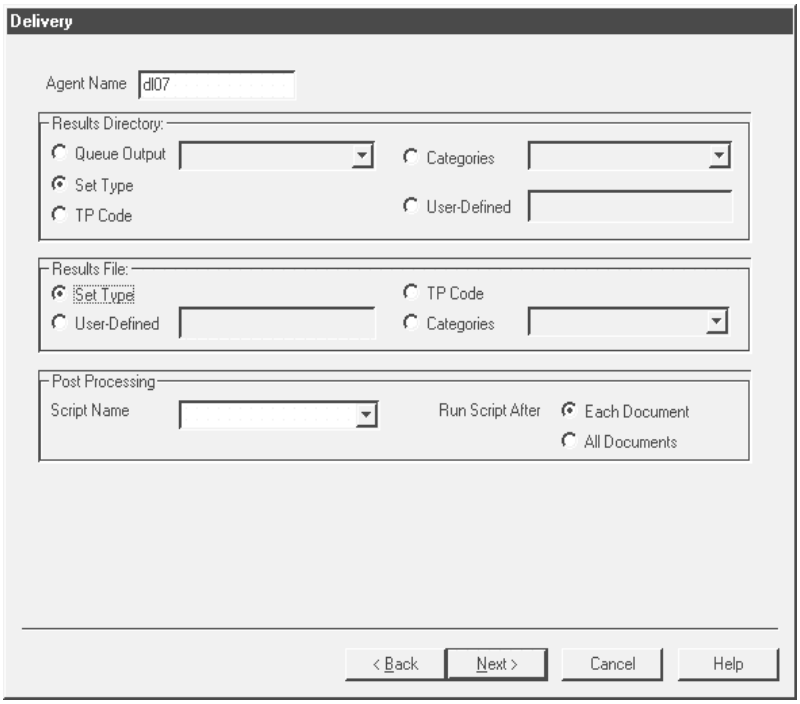
The Delivery agent dialog box enables you to set information that the Process Control Manager uses in the Trading Partnership configuration records it creates.

You can set:

- Exact destination directory and file name information that you want the Process Control Manager to use in every configuration record that it creates from the flow
- Symbolic destination directory and file name values, such as a category or Trading Partnership Code. The Process Control Manager substitutes the actual value for the symbolic value in the configuration records
- The name of the script Sterling Gentran:Server runs after processing the Trading Partner's files. You also select whether the script runs after each document is processed or after all documents are processed.

Procedure

Use this procedure to set up the delivery data manager for an XML flow.

Step	Action
1	Type the name of the delivery data manager in the Agent Name box. 
2	Choose the Results Directory by clicking **Set Type**, **TP Code**, **Categories**, or **User-Defined** to select the symbolic destination directory, or type the path to the directory for the translation output. **Comments** The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. The **Set Type** option is not available if the destination format is XML. **Example** If you select **TP Code**, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.

(Contd) Step	Action
3	Did you select **Categories** in Step 2? ◗ If YES, select a category from the drop-down list. ◗ If NO, continue with Step 4.
4	Did you select **User-Defined** in Step 2? ◗ If YES, type the path to the directory in the text box that is below the **User-Defined** option. ◗ If NO, continue with Step 5.
5	Choose the Results File name by clicking **Set Type**, **TP Code**, **Categories**, or **User-Defined** to select the symbolic destination file name. **Comments** The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. The **Set Type** option is not available if the destination format is XML. **Example** If you select TP Code, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.
6	Did you select **Categories** in Step 5? ◗ If YES, select a category from the drop-down list and continue with Step 7. ◗ If NO, continue with Step 7.
7	Did you select **User-Defined** in Step 5? ◗ If YES, type the complete file name in the text box that is below the **User-Defined** option and continue with Step 8. ◗ If NO, continue with Step 8.
8	Do you want to run a post-processing script? ◗ If YES, select the name of the script from the Script Name drop-down list and continue with Step 9. ◗ If NO, continue with Step 9.
9	Click the **Each document** or **All documents** option to select when the system runs the script.

(Contd) Step	Action
10	Click **Next** to continue to the Error Handling dialog box. **Reference** See How to Set Up Error Handling Instructions for instructions on completing the Error Handling dialog box.

Creating an Inbound NCPDP Flow

Overview

Introduction

If you use the National Council of Prescription Drug Programs (NCPDP) EDI standard, you can create inbound NCPDP flows with the Process Control Manager.

Flow types

These are the inbound NCPDP flows:

- standard-to-standard
- standard-to-application
- standard-to-XML

Note
The source file is in an NCPDP standard format.

Data managers in an inbound NCPDP flow

An inbound NCPDP flow has these three data managers:

- Source agent - Processes EDI data. Starts the movement of data in the flow.
- Processing agent - Starts a translation script that runs the translator program **Iftran** with the -i (inbound) option.
- Delivery agent - Runs an embedded script that does nothing. Primary role is to generate a Life Cycle event record for auditing purposes. Can also run a script to perform any after-translation processing on the data.

Routing direction

This table describes the routing direction in an inbound NCPDP flow.

Stage	Description
1	A source agent: ◗ Receives files in NCPDP format ◗ Splits NCPDP files by trading partner ◗ Processes data and routes data to the processing agent.

(Contd) Stage	Description
2	The processing agent starts the translator program, **lftran,** which translates the data and routes it to a delivery agent.
3	The delivery agent runs an embedded script named *<dmname>_gen_xltr.scr.* By default, this script does nothing, but the process generates a Life Cycle event record. If you have specified a post-processing script on the Delivery setup dialog box, the delivery agent runs the script.

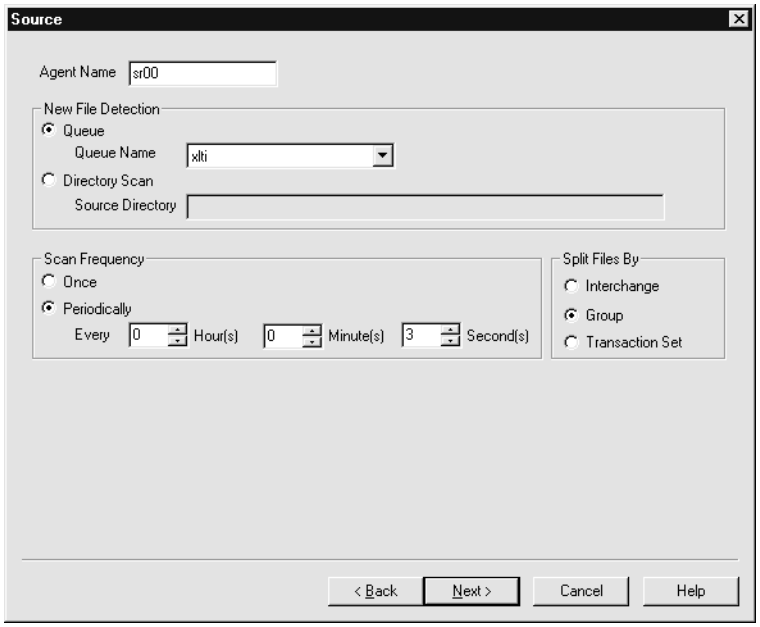
Source Dialog Box (Inbound NCPDP Flow)

Introduction

The **Source** dialog box for an inbound NCPDP flow is used to create the NCPDP data manager that starts your process flow.

Illustration

This illustration shows the **Source** dialog box.



Source fields and functions

This table describes the fields of the **Source** dialog box and their functions.

Field	Function
Agent Name	Defines the name of the source data manager. The maximum size is 4 characters. **Note** The system supplies a default name, which is based on file type selected on the **Flow Identification** dialog box. You can override the default name.
Queue	Selects queue as the source type that the data manager looks in for new files to process.

(Contd) Field	Function
Queue Name	Enables you to select (from the drop-down list) the name of the queue that the data manager looks in for new files. The drop-down list contains the names of all the existing queues.
Directory Scan	Selects a scan directory as the source type that the data manager looks in for new files to process.
Source Directory	Enables you to type the name of the directory that the data manager looks in for new files. If you want the wizard to create the directory, type the name, using the relative path for EDI_ROOT. **Example** ./sr03 **Note** If you choose to use a nested directory structure, you must create the directory path first and then type it in the Source Directory box.
Once	Selects one time as the scan frequency.
Periodically	Enables you to select the frequency with which you want the source data manager to scan its work directory.
Hour(s)	Defines, in hours, the frequency with which the data manager scans for new files. Value range is 0 to 23.
Minute(s)	Defines, in minutes, the frequency with which the data manager scans for new files. Value range is 0 to 59.
Second(s)	Defines, in seconds, the frequency with which the data manager scans for new files. The default value is 3 seconds. Value range is 0 to 59.
Interchange	Selects interchange code as the splitting method to route files.
Group	Selects group code as the splitting method to route files.
Transaction Set	Selects transaction set as the splitting method to route files.

How to Set Up the Source Agent (Inbound NCPDP Flow)

Introduction

The **source agent** is the data manager that starts your process flow. In an inbound NCPDP flow, the source agent is a data manager with an NCPDP (p) personality.

Before you begin

You must complete the procedures in the Beginning a Flow section in this chapter first.

CAUTION

Be sure to check the For NCPDP check box on the Flow Identification dialog box.

Procedure

Use this procedure to set up the source data manager for an inbound NCPDP flow.

Step	Action
1	Type the name of the source data manager in the **Agent Name** box. **Note** Sterling Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.
2	Select either **Queue** or **Directory Scan** as the type of source that you want the source data manager to examines for files to process.

(Contd) Step	Action
3	Did you select **Queue** in Step 2? ◗ If YES, select the name of the queue from the drop-down list and continue with Step 4. ◗ If NO, (the source is a directory), type the relative path name to the directory in the text box and continue with Step 4. **CAUTION** **If a queue or directory is used by another data manager (source, processing or delivery agent), do not use it as the source for this data manager.**
4	Click **Once** or **Periodically** to select the scan frequency. **Note** The scan frequency you select applies to every data manager in the flow.
5	Did you select **Periodically** in Step 3? ◗ If YES, complete the Hour(s), Minute(s) and Second(s) boxes to select the frequency with which the data manager awakens and scans the queue or directory. ◗ If NO, continue with Step 5.
6	Click **Interchange**, **Group**, or **Transaction Set** to select how the data manager groups routed data.
7	Click **Next** to continue to the **Processing** dialog box. **Reference** See How to Set Up the Processing Agent (Inbound NCPDP Flow).

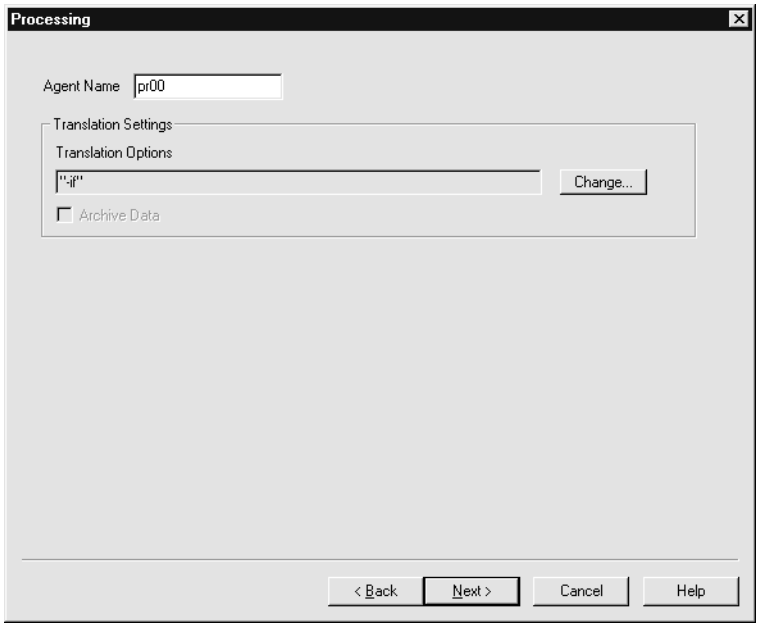
Processing Dialog Box (Inbound NCPDP Flow)

Introduction

The **processing agent** is the second data manager in a flow. In an inbound NCPDP flow, its main function is to set translation options.

Processing Agent dialog box

This illustration shows the **Processing** dialog box for an inbound NCPDP flow.



Processing Agent fields and functions

This table describes the fields of the **Processing** dialog box and their functions.

Field	Function
Agent Name	Defines the name of the processing data manager. **Note** The system supplies a default name, which is based on file type you selected on the **Flow Identification** dialog box. You can override the default name.
Translation Options	Displays the currently selected translation options.
Change	Displays the **Translation Options** dialog box. **Reference** See the Translation Options Dialog Box topic in this section.
Archive Data	Not available for inbound NCPDP flows.

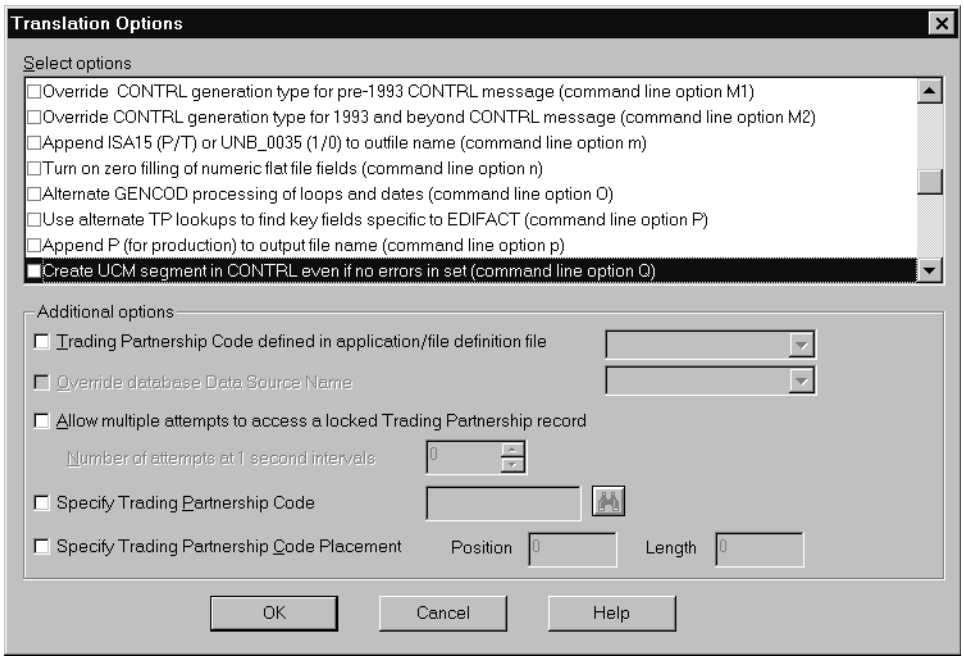
Translation Options Dialog Box

Introduction

Sterling Gentran:Server displays the **Translation Options** dialog box when you click the Translation Options **Change** button on the **Processing** dialog box.

Illustration

This illustration shows the **Translation Options** dialog box.



Translation Option fields and functions

This table describes the fields of the **Translation Options** dialog box and their functions.

Field	Function
Select options	Enables you to select the translation options you want to apply to this flow. **Reference** For a list of translation options, see the *Iftran Syntax* topic in the *Command Reference* chapter of the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide*.
Trading Partnership code defined in application/ file definition file	Enables you to select the application or file definition file. Used only for outbound translations.
Allow multiple attempts to access a locked Trading Partnership record	Allows the data manager to attempt more than one time to access a locked Trading Partnership record.
Number of attempts at 1 second intervals	Enables you to specify the number of times the data manager should attempt to access a locked Trading Partnership record before translation fails.
Specify Trading Partnership Code	Enables you to search for the Trading Partnership code that you want to use to override Trading Partnership data. Used only for outbound translations.
Specify Trading Partnership Code Placement	Enables you to specify the Trading Partnership code's position in the file and the length of the of the code. Used only for outbound translations.

How to Set Up the Processing Agent (Inbound NCPDP Flow)

Introduction

The **processing agent** in an inbound NCPDP flow invokes the translator. In an inbound NCPDP flow, the processing agent is a data manager with a translation (x) personality.

Procedure

Use this procedure to set up the processing data manager for an inbound NCPDP flow.

Step	Action
1	Type the name of the processing data manager in the **Agent Name** box. **Note** Sterling Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.
2	Do you want to change the translation options? ◗ If YES, click the **Change** button and complete the **Translation Options** dialog box. ◗ If NO, continue with Step 3.
3	Do you want the translation script to run **ediarc**? ◗ If YES, select **Archive Data** and then click **Next** to continue to the **Delivery** dialog box. ◗ If NO, click **Next** to continue to the **Delivery** dialog box. **Note** The **ediarc** program archives translation data. **Reference** See How to Set Up the Delivery Agent (Inbound NCPDP Flow).

Delivery Dialog Box (Inbound NCPDP Flow)

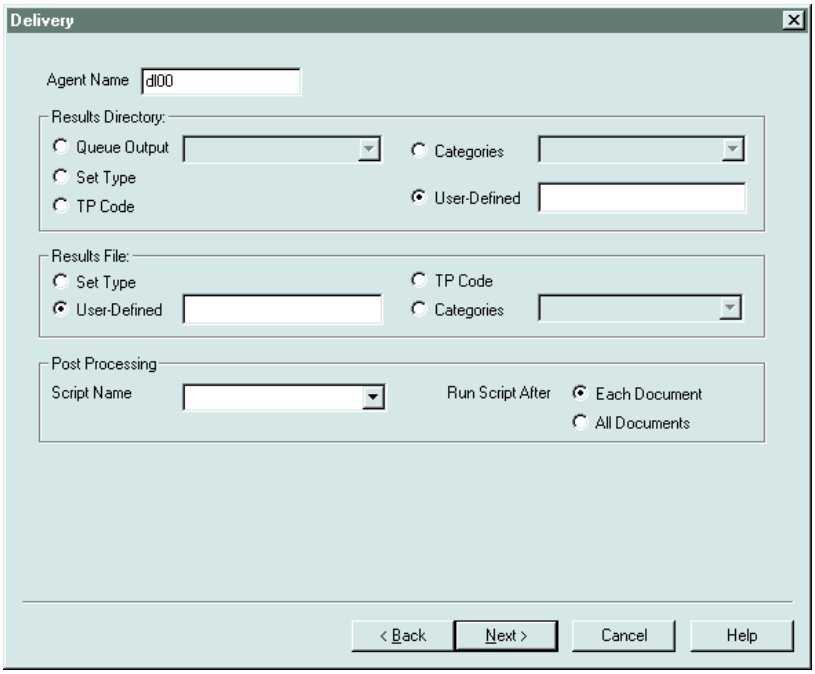
Introduction

The **delivery agent** is the third data manager in an inbound NCPDP flow. Its function in an inbound NCPDP flow is to:

- Designate the results (output) directory and file name
- Specify the name of the post processing script and when the script is run.

Delivery Agent dialog box

This illustration shows the **Delivery** dialog box for an inbound NCPDP flow.



Delivery fields and functions

This table describes the fields of the **Delivery** dialog box and their functions.

Field	Function
Agent Name	Defines the name of the delivery data manager. **Note** The system supplies a default name, which is based on file type you selected on the **Flow Identification** dialog box. You can override the default name.
Queue Output	Enables you to select (from the drop-down list) the name of a queue as the destination to which the delivery data manager directs the files it has processed. The drop-down list contains the names of all the existing queues.
Set Type	Selects transaction set type as the symbolic value for the Results Directory (destination directory) in the configuration records. The Process Control Manager substitutes the actual value for the type of transaction set in the configuration records.
TP Code	Selects Trading Partnership Code as the Results Directory (destination directory) in the configuration records. The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.
Categories	Enables you to specify a Trading Partnership category as the Results Directory (destination directory) in the configuration records. Select the category from the drop-down list box that is next to the Categories option. The Process Control Manager substitutes the actual category value in the configuration records.
User Defined	Enables you to specify the Results Directory (destination directory) for the configuration records. Enter the path in the text box that is next to the User Defined option.

(Contd) Field	Function
Set Type	Selects transaction set type as the symbolic value for the Results File (output file name) in the configuration records. The Process Control Manager substitutes the actual value for the type of transaction set in the configuration records.
TP Code	Selects Trading Partnership Code as the Results File (output file name) in the configuration records. The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.
Categories	Enables you to specify a Trading Partnership category as the Results File (output file name) in the configuration records. Select the category from the drop-down list box that is next to the Categories option. The Process Control Manager substitutes the actual category value in the configuration records.
User Defined	Enables you to specify the Results File (output file name) for the configuration records. Enter the path in the text box that is next to the User Defined option.
Script Name	Enables you to enter or select the name of the script you want to run after this data manager has processed the files.
Each Document	Executes the post-processing Sterling Gentran:Server script after each document has been processed.
All Documents	Executes the post-processing Sterling Gentran:Server script after all documents have been processed.

How to Set Up the Delivery Agent (Inbound NCPDP Flow)

Introduction

The **delivery agent** is the destination data manager in a process flow. In an inbound NCPDP flow, the delivery agent is a data manager with a translation (x) personality.

Setting configuration record information

The **Delivery** agent dialog box enables you to set information that the Process Control Manager uses in the Trading Partnership configuration records it creates.

You can set:

- Exact destination directory and file name information that you want the Process Control Manager to use in every configuration record that it creates from the flow
- Symbolic destination directory and file name values, such as a category or Trading Partnership Code. The Process Control Manager substitutes the actual value for the symbolic value in the configuration records
- The name of the script (if any) Sterling Gentran:Server runs after processing the Trading Partner's files. You also select whether the script runs after each document is processed or after all documents are processed.

Procedure

Use this procedure to set up the delivery agent for an inbound NCPDP flow.

Step	Action
1	Type the name of the delivery data manager in the **Agent Name** box. **Note** Sterling Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.
2	Choose the Results Directory by clicking **Set Type**, **TP Code**, **Categories**, or **User-Defined** to select the symbolic destination directory, or typing the path to the directory for the output. **Comment** The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. The Process Control Manager creates directories if they do not exist. **Example** If you select **TP Code**, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.
3	Did you select **Categories** in Step 2? ◗ If YES, select a category from the drop-down list. ◗ If NO, continue with Step 4.
4	Did you select **User-Defined** in Step 2? ◗ If YES, type the path to the directory in the text box that is below the User-Defined option. ◗ If NO, continue with Step 5.

(Contd) Step	Action
5	Choose the Results File name by clicking **Set Type**, **TP Code**, **Categories**, or **User-Defined** to select the symbolic destination file name. **Comment** The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. **Example** If you select **TP Code**, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.
6	Did you select **Categories** in Step 5? ◗ If YES, select a category from the drop-down list and continue with Step 7. ◗ If NO, continue with Step 7.
7	Did you select **User-Defined** in Step 5? ◗ If YES, type the complete file name in the text box that is below the **User-Defined** option and continue with Step 8. ◗ If NO, continue with Step 8.
8	Do you want to execute a script after the translation process? ◗ If YES, select the name of the script from the **Script Name** drop-down list and continue with Step 9. ◗ If NO, continue with Step 9.
9	Click the **Each document** or **All documents** option to select when the system runs the script.
10	Click **Next** to continue to the **Error Handling** dialog box. **Reference** See the Completing a Flow section in this chapter for instructions on completing the **Error Handling** dialog box.

Completing a Flow

Overview

Introduction

This section describes how to complete a process flow.

Task summary

This table summarizes the tasks in completing a process flow.

Task	Description
1	Select the error handling options. **Reference** See How to Set Up Error Handling Instructions.
2	Select the Trading Partnership codes to use in the process flow. **Reference** See How to Add Trading Partnership Records to the Flow.

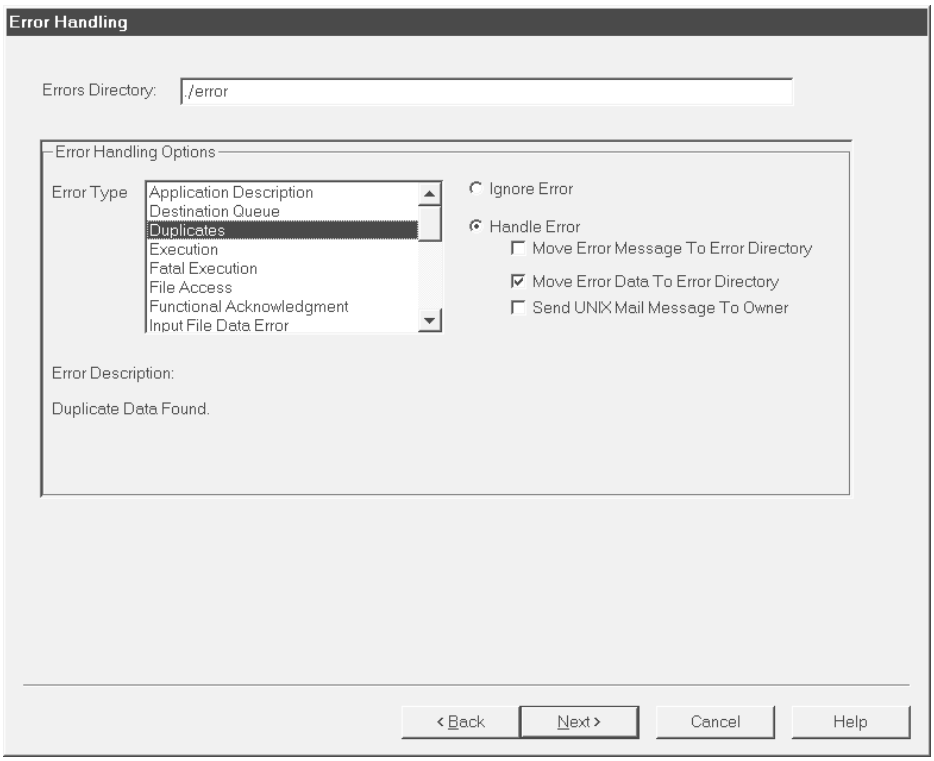
Error Handling Dialog Box

Introduction

The **Error Handling** dialog box is used to define the way in which you want errors handled.

Illustration

This illustration shows the Error Handling dialog box.



Error Handling fields and functions

This table describes the fields of the Error Handling dialog box and their functions.

Use this field	To
Error Directory	Define the name of the destination directory for errors. **Note** The default is *./error*.
Error Type	Select a type of error so that you can specify how you want Sterling Gentran:Server to handle it.
Ignore Error	Turn error handling off.
Handle Error	Turn error handling on.
Move Error Message to Error directory	Route a copy of the error message to the specified error directory.
Move Error Data To Error Directory	Route a copy of the data that is in error to the specified error directory.
Send UNIX Mail Message To Owner	Route the error message to the name specified in the UNIX mail_proc file associated with the error type. **Reference** For instructions on how to add, edit, and delete UNIX mail_proc scripts, see the Working with UNIX Mail Scriptssection in the Working with Scripts chapter in this guide.

How to Set Up Error Handling Instructions

Introduction

The error handling instructions describe how the translation data manager deals with the various types of errors it can encounter. The Process Control Manager supports 20 different types of errors. Each error type has default handling instructions, which you can override.

Error handling options

These are your error handling options:

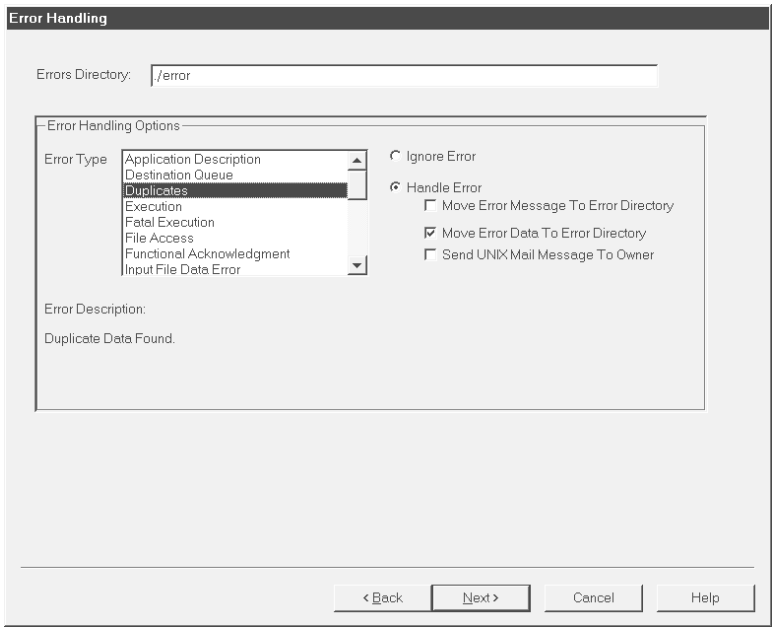
- Ignore the error
- Move the error message to the error directory
- Move the data that is in error to the error directory
- Move both the error message and the data that is in error to the error directory
- Send the error message to the e-mail address specified in the mail_proc file. The default is to send e-mail to the user who started the data manager.
- Move the data in error to the error directory and send the error message to the e-mail address specified in the mail_proc file.

CAUTION

If you are an advanced UNIX user, you can modify the UNIX mail script (mail_proc) file to include the e-mail address for error messages or to make other modifications.

Procedure

Use this procedure to set up error handling instructions for the process flow.

Step	Action
1	Type the name of the directory to which you want errors routed. 
2	Select an error type from the **Error Type** list.
3	Do you want Sterling Gentran:Server to handle this type of error? ◗ If YES, click **Handle Error** and then click on the way you want Sterling Gentran:Server to handle errors of this type: Move Error Message To Error Directory, Move Error Data to Error Directory, Send UNIX Mail Message To Owner. You can choose more than one option. ◗ If NO, click **Ignore Error**. **Note** If you select **Handle Error**, but do not select an instruction, Sterling Gentran:Server ignores the error.
4	Repeat Steps 2 and 3 until you have selected instructions for each error type.
5	Click **Next** to continue to the Trading Partner Records dialog box. **Reference** See How to Add Trading Partnership Records to the Flow.

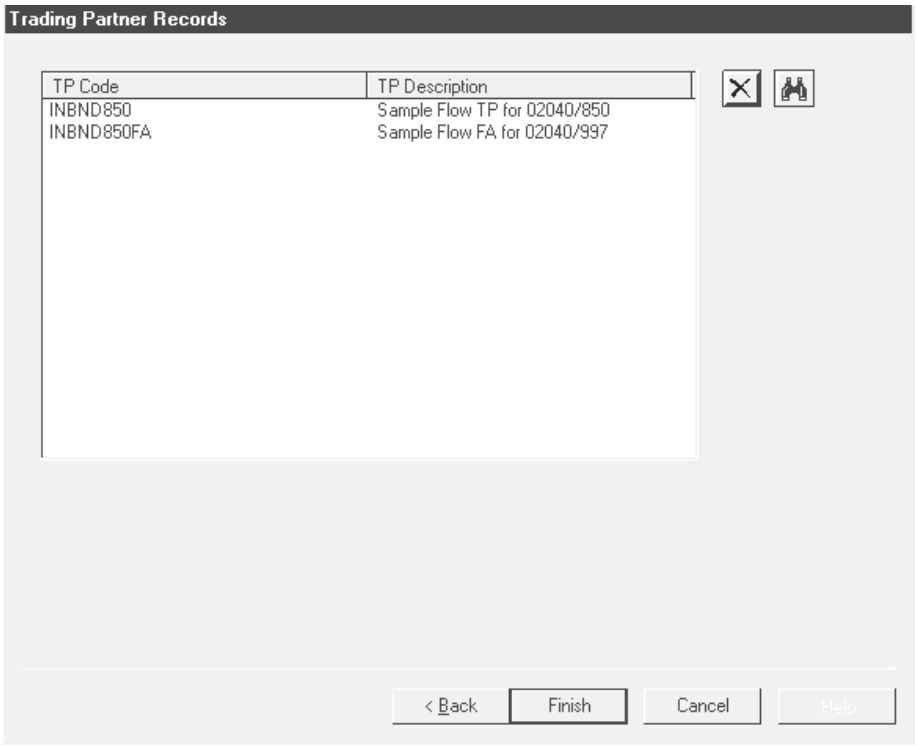
Trading Partner Records Dialog Box

Introduction

The **Trading Partner Records** dialog box enables you to add a list of Trading Partnership records to a process flow. This list appears blank until you add Trading Partnership records to it.

Illustration

This illustration shows the Trading Partner Records dialog box. This illustration shows the addition of two Trading Partnerships.



Trading Partner Records dialog box fields and functions

This table describes the fields of the Trading Partner Records dialog box and their functions.

Field	Function
TP Code	Lists the Trading Partnership codes of the Trading Partnership records in the flow.
TP Description	Describes the Trading Partnership record.

How to Add Trading Partnership Records to the Flow

Introduction

The final step in creating a process flow is to link one or more Trading Partnership records to the flow.

Purpose

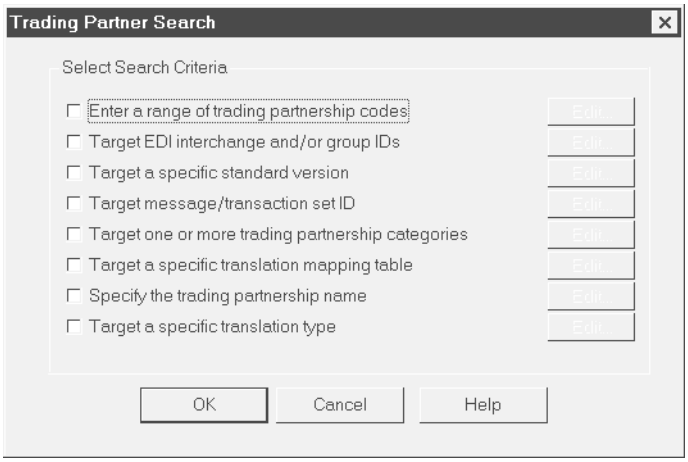
You link Trading Partnership records to the flow so that the Process Control Manager can generate the configuration records. A configuration record describes how a data manager directs the data that it handles for a particular Trading Partnership code or file name.

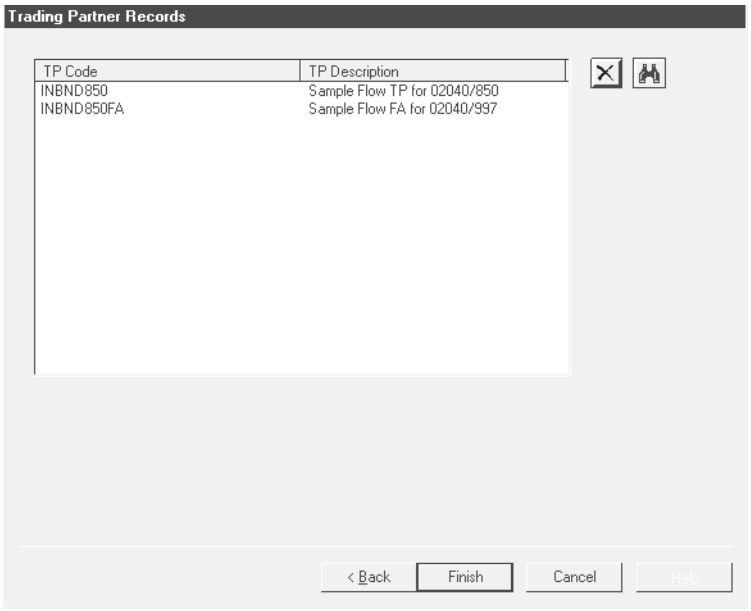
Reference

For information about Trading Partnership records, see the *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide.*

Adding Trading Partnerships

Use this procedure to add a Trading Partnership record to the process flow.

Step	Action
1	Click on the search icon. **System Response** Sterling Gentran:Server displays the Trading Partner Search dialog box. 
2	Search for the Trading Partnership code that you want to link to the flow. **Reference** See the *Working with Trading Partnerships* chapter in the *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide* for instructions on using the Trading Partner Search dialog box. **System Response** Sterling Gentran:Server displays the Trading Partner Search Results dialog box. This dialog box lists the Trading Partnership records that match the criteria you entered.

(Contd) Step	Action
3	Click the Trading Partnerships that you want to link to the flow and then click **OK**. **System Response** Sterling Gentran:Server adds the Trading Partnerships to the flow and lists the codes and descriptions in the Trading Partner Records dialog box. 
4	Click **Finish** to save the new flow.

How to Delete Trading Partnerships from the Trading Partner Records Dialog Box

Introduction

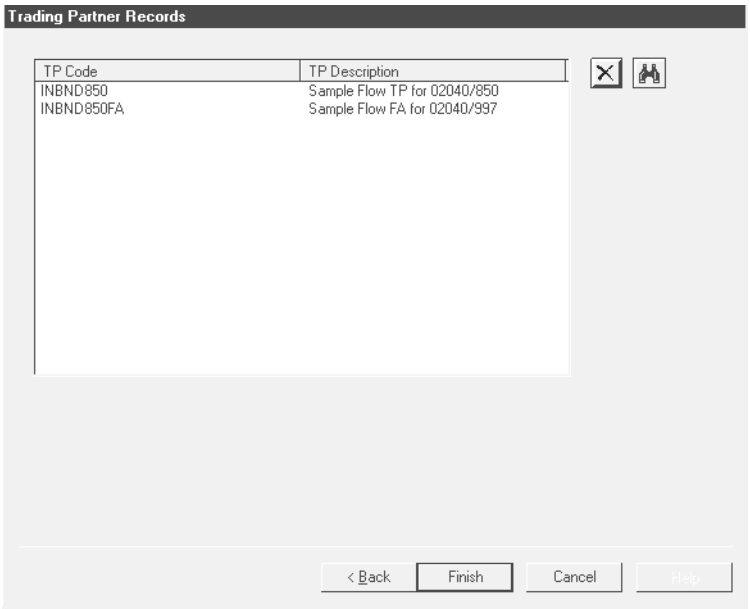
You can delete Trading Partnerships from the Trading Partner Records dialog box to remove them from a flow.

Reference

You can also delete Trading Partnerships from a flow by deleting them from the Process Control Manager flow tree. See How to Delete Trading Partnerships From the Flow for instructions.

Procedure

Use this procedure to remove a Trading Partnership from the Trading Partner Records dialog box.

Step	Action
1	Click the Trading Partnership code you want to delete on the Trading Partner Records dialog box. 

(Contd) Step	Action
2	Click the **Delete** icon. **System Response** The Process Control Manager removes the Trading Partnership code from the list.
3	Click **Finish** to save the changes to the flow.

Using Flow Summaries

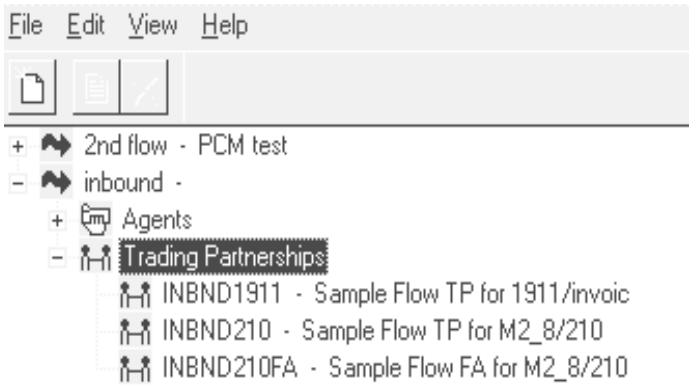
The Flow Summary

Introduction

The **Process Control Manager** window displays a visual summary of your process flows.

Illustration

This is a flow summary. This illustration displays the inbound flow and its Trading Partnerships in the expanded view.



Icons

This table describes the icons in a flow view and their functions.

Icon	Function
	Represents a flow.
	Represents the data managers (agents) in the flow.
	Represents the Trading Partnerships in the flow.

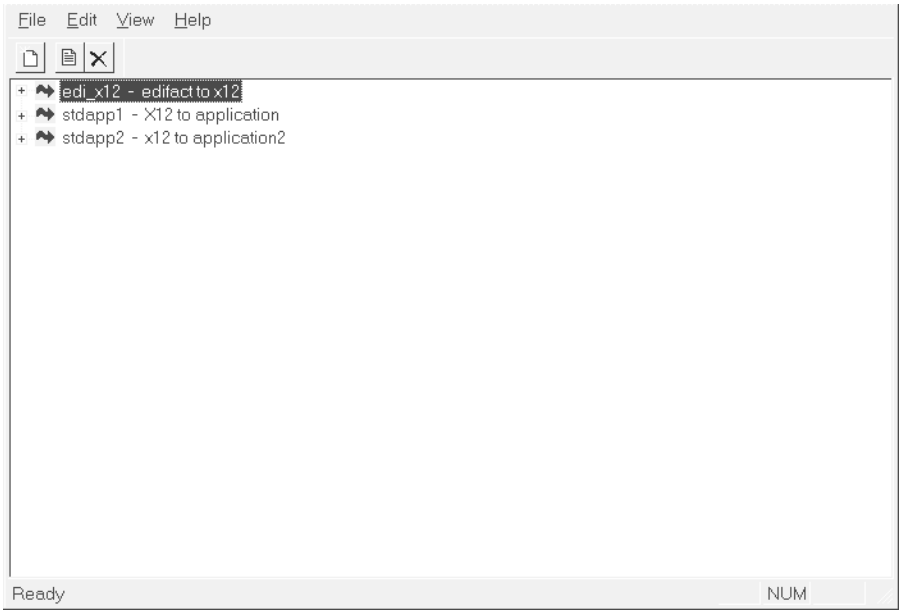
Flow Summary Views

Introduction

You can view flow information on the Process Control Manager window in either **collapsed** view or **expanded** view.

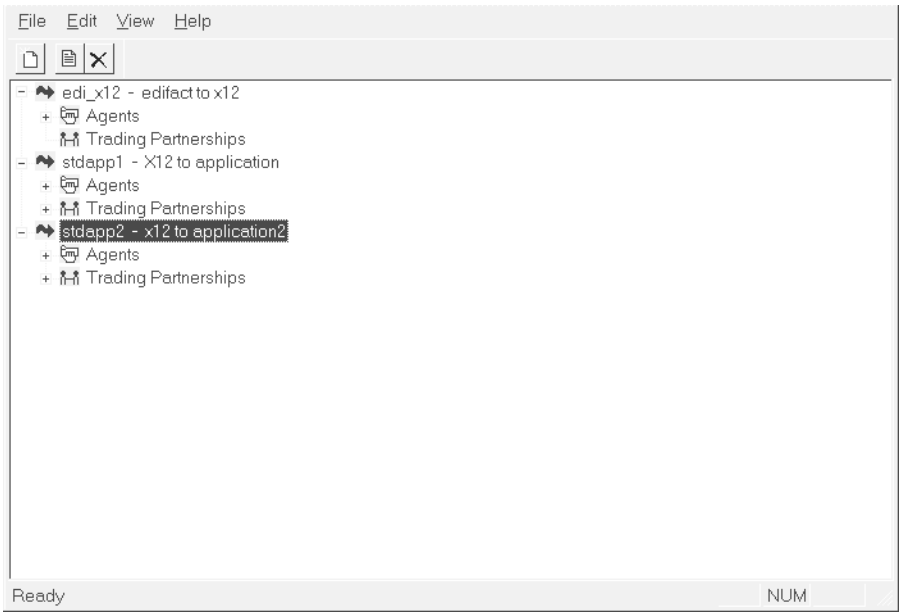
Fully collapsed view

When you first access the Process Control Manager window, all views are collapsed. The window displays a Flow icon for each flow.



Expanded flow; Agents and Trading Partnerships collapsed

When you expand the view for a flow, the window displays the Agents and Trading Partnerships icons for the flow. However, the icons for the individual data managers and Trading Partnerships are not visible; the view is collapsed for the Agents and Trading Partnerships.

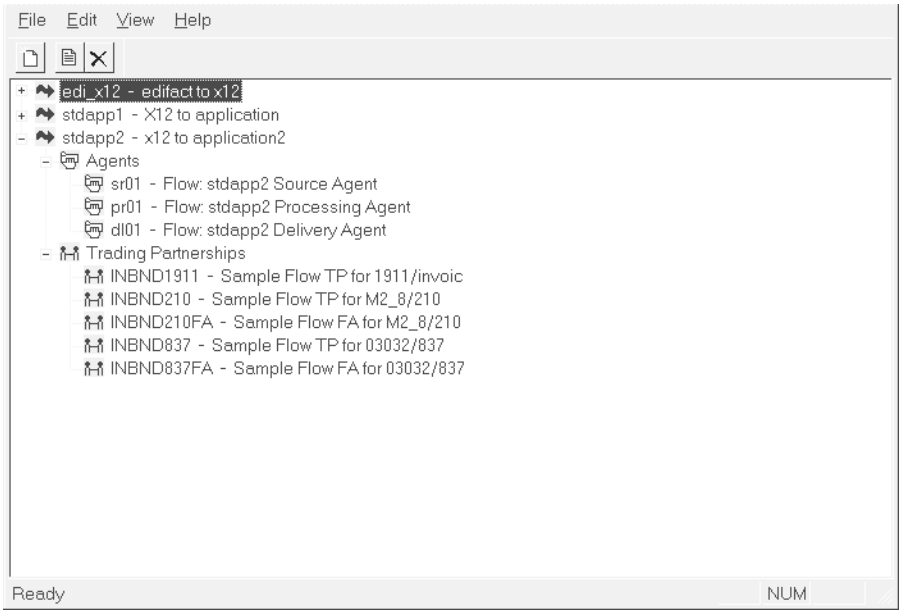


Fully expanded view

When you expand the Agents icon, the window displays an icon for each data manager in the flow.

When you expand the Trading Partnerships icon, the window displays an icon for each Trading Partnership linked to the flow.

In this example, both the Agents and Trading Partnerships are expanded for the flow named **stdapp2**.



How to Expand and Collapse the Flow View

Introduction

You can expand the view for a flow, just the data managers in the flow, or just the Trading Partnerships in the flow.

Procedure

Use this table to expand or collapse the view.

IF you want to view…	THEN…
The Agents icons and Trading Partnerships icons for a flow	Double-click on the **Flow** icon.
The individual data managers in the flow	Double-click on the **Agents** icon.
The individual Trading Partnerships in the flow	Double-click on the **Trading Partnerships** icon.
Only the flow names	Click **Refresh** on the View menu to collapse the view.

Note
You can also use the plus (+) and minus (-) symbols to expand and collapse the view.

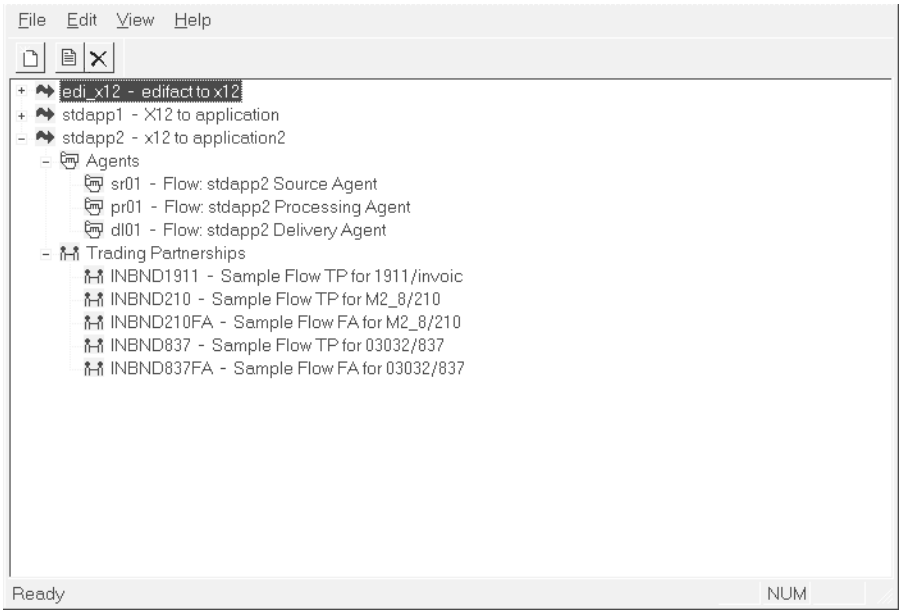
Flow Summary Reports

Introduction

You can print summary text reports from the Process Control Manager window.

Illustration

This is a flow summary. This illustration displays the stdapp2 flow and its components in the expanded view.



Types of reports

This table describes the types of reports you can print.

This report…	Summarizes…
Basic flow Information	The flow identification information and data manager identification information for each data manager in the flow.
Expanded flow information	The flow identification information, data manager identification information for each data manager in the flow, and Trading Partnerships information for every Trading Partnership in the flow.

(Contd) This report…	Summarizes…
All agents Information	Data manager identification information for every data manager in the flow.
Single agent Information	Data manager identification information for the selected data manager.
All Trading Partnerships Information	Trading Partnership identification information for every Trading Partnership in the flow.
Single Trading Partnership Information	Trading Partnership identification information for the selected Trading Partnership.

How to Print Flow Summary Reports

Introduction

You select the level of detail (type of report) you want by:

- Expanding the appropriate icon
- Selecting the icon that represents the level of detail.

Printing summary reports

Use this procedure to print a flow summary report.

Step	Action	
1	Click the **PCM** button on the Sterling Gentran:Server client toolbar to open the Process Control Manager window.	
2	Expand the view as necessary and select the icon that represents the type of report you want to print.	
	IF you want to print…	**THEN…**
	Basic flow Information	Click the flow. Do not expand the view.
	Expanded flow information	◗ Double-click on the Flow icon to expand the view. ◗ Click the Flow icon.
	All agents Information	◗ Double-click on the Flow icon to expand the view. ◗ Click the Agents icon.
	Single agent Information	◗ Double-click on the Flow icon to expand the view. ◗ Double-click on the Agents icon to expand the view. ◗ Click the icon for the individual agent.
	All Trading Partnerships Information	◗ Double-click on the Flow icon to expand the view. ◗ Click the Trading Partnerships icon.
	Single Trading Partnership Information	◗ Double-click on the Flow icon to expand the view. ◗ Double-click on the Trading Partnerships icon to expand the view. ◗ Click the icon for the individual Trading Partnership.

(Contd) Step	Action
3	Click **Print** on the File menu. **System Response** Sterling Gentran:Server displays the Print dialog box.
4	Click OK.

Maintaining Process Flows

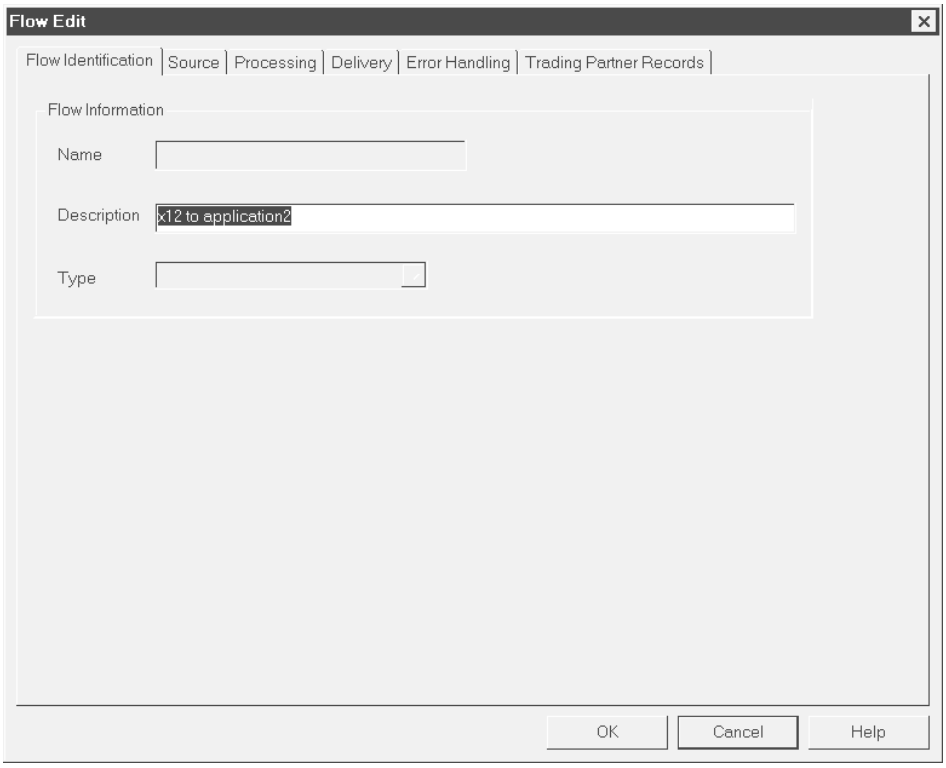
Overview

Introduction

You can change most aspects of a process flow. To edit a flow, you change information on the flow's property sheet tabs.

Flow Edit property sheet

This illustration shows the **Flow Edit** property sheet. Each tab displays a dialog box that was completed when the flow was created.



How to Edit a Process Flow

Introduction

This topic explains how to change aspects of a process flow that was created with the Process Control Manager.

References

If you want to delete a Trading Partnership from a flow, see How to Delete Trading Partnerships From the Flow.

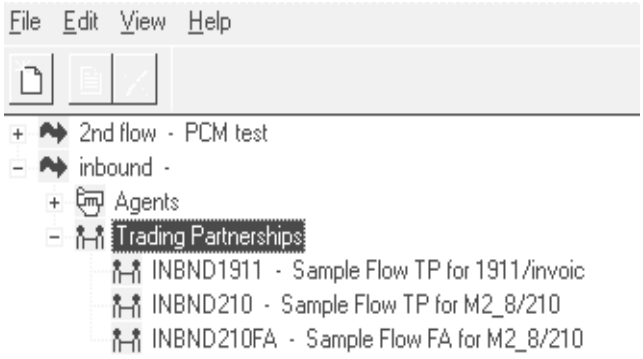
If you want to delete an entire flow, see How to Delete a Flow.

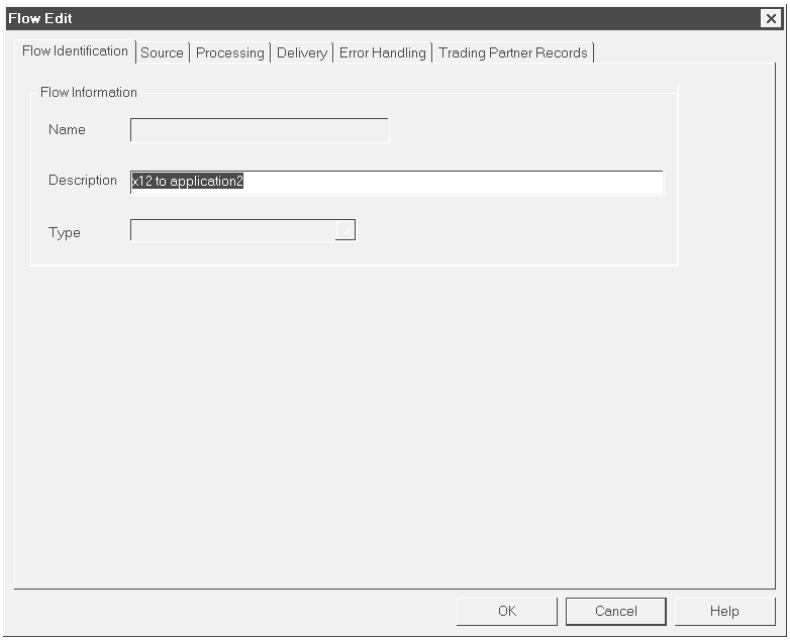
Flow information you cannot change

Once you have created a flow, you cannot change the flow's name or type (direction).

Editing a flow

Use this procedure to edit a process flow.

Step	Action
1	Start the Process Control Manager. **System Response** Sterling Gentran:Server displays the Process Control Manager window. 
2	Click the flow that you want to edit.

(Contd) Step	Action
3	Click **Edit Flow** on the Edit menu. **System Response** Sterling Gentran:Server displays the Flow Edit window for the flow you selected. 
4	Click on the tab that has the information you want to change.
5	Make all your changes. **CAUTION** **The items you can change are active. The items you cannot change are unavailable.**
6	Click **OK** to save your changes.

How to Delete Trading Partnerships From the Flow

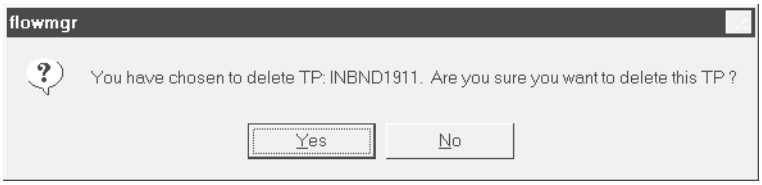
Introduction

This topic explains how to delete a Trading Partnership from a flow.

Reference
You can also delete Trading Partnerships from a flow by deleting them from the Trading Partner Records dialog box. See How to Delete Trading Partnerships from the Trading Partner Records Dialog Box for instructions.

Procedure

Use this procedure to delete a Trading Partnership from the flow.

Step	Action
1	Start the Process Control Manager to display the flow tree.
2	Click on the flow to display the Agents and Trading Partnerships icons.
3	Click on **Trading Partnerships** to display the Trading Partnerships in the flow.
4	Click the Trading Partnership that you want to delete.
5	Click **Delete** on the Edit menu. **System Response** The Process Control Manager displays a confirmation prompt. 
6	Click **Yes** to confirm the deletion.

How to Delete a Flow

Introduction

You can delete a flow if:

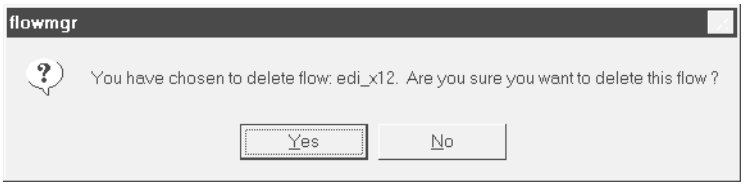
- Your application has changed
- You need to replace a flow.

Consequences of deleting a flow

When you delete a flow, Sterling Gentran:Server deletes the records associated with the flow, but not the directories or queues.

Deleting a flow

Use this procedure to delete a flow.

Step	Action
1	Start the Process Control Manager to display the flow tree.
2	Click the flow that you want to delete.
3	Click **Delete** on the Edit menu. **System Response** The Process Control Manager displays a confirmation prompt. 
4	Click **OK**.

CHAPTER
4

Using Queues

Contents

Overview
- Introduction 2
- Queues 4
- The Queue Process 6
- Queue Select Screen 7
- Queue File Screen 8
- How to Create a Queue 10

Maintaining Queue Entries
- Overview 12
- How to View Entries in a Queue 13
- Add Queue Entry Screen 15
- How to Add an Entry to a Queue 17
- How to Delete an Entry From a Queue 19

Maintaining Queues
- Overview 21
- How to Remove a Queue from the Select List 22
- How to Delete a Queue 23

Overview

Introduction

In this chapter

This chapter describes queues and how to use them in Sterling Gentran:Server.

Key terms

This table lists the key terms used in this chapter.

Term	Description
configuration record	A record that describes how a data manager directs the data that it handles for a particular Trading Partnership code or file name. The record: ◗ Specifies the Trading Partnership code or file name that the data manager is to use to identify data ◗ Tells the data manager what to do with the data it has identified.
downstream data manager	A data manager that processes files that a previous (upstream) data manager has placed in its work directory or queue.
initialization file	The configurable file that sets the data manager's personality and processing parameters.
ISAM file	Indexed Sequential Access Method file. A two-part file in which one part of the file contains a list of records and the other part contains an index to the records. **Comment** An ISAM file has the file extensions *.dat* and *.idx*.
priority	A number from 0 through 9 that designates the order in which a downstream data manager is to process files. A 0 represents the highest priority and 9 represents the lowest priority.
queue	An indexed list of files to be processed.

(Contd) Term	Description
upstream data manager	A data manager that processes files and then routes them to the work directory or queue of another data manager known as the downstream data manager.
work directory	The directory or queue in which a data manager looks for the files or file names it is to process.

Queues

Definition

A **queue** is an alternative to a work directory for a data manager. Queues hold a list of information that enables a data manager to find the files they are to process, but not the actual data files.

A queue is a hidden indexed sequential access method (ISAM) file. ISAM files are two-part list files:

- One part of the file (<*queuename*>*/.q.dat*) contains a record of each file written to the queue.
- The second part of an ISAM file (<*queuename*>*q.idx*) contains an index that shows the location of each record in the <*queuename*>*/.q.dat* portion.

Sterling Gentran:Server uses the <*queuename*>*/.q.idx* part of the ISAM file to access the records in the <*queuename*>*/.q.dat* part.

Information in a queue

The information in a queue includes the:

- File name
- Name of the directory in which the file resides
- Name of the resource group to which the file belongs
- Processing priority the file should be given
- File's unique ID that Sterling Gentran:Server assigned.

Reading from and writing to queues

You can configure the source agent to use a queue as its source of files. You can configure the delivery agent to use a queue as the file destination. The PCM wizard creates two queues:

- The queue used as the source agent's destination and the processing agent's source
- The queue used as the processing agent's destination and the delivery agent's source.

Upstream and downstream data managers

The data manager that writes to a queue is called the **upstream data manager**. The data manager that reads from the queue is called the **downstream data manager**.

Queue names

When you follow the procedure in this chapter for creating a queue, you give the queue a name. Sterling Gentran:Server creates the:

- Subdirectory for the queue
- Two parts of the ISAM file.

Example
You create a queue and name it inque1.

Sterling Gentran:Server creates:

- The subdirectory *$EDI_ROOT/inque1*
- The ISAM files *.q.dat* and *.q.idx* (in the *inque1* subdirectory).

Benefits of using queues

There are several benefits to having a data manager read from a queue.

- A queue enables you to have files reside in more than one location. The queue file tells a data manager where to find the files it has to process.
- You can assign processing priority to files. This means that you can process files:
  - From specific trading partners before others
- You can have multiple data managers read from the same queue at the same time.
- Queues help avoid concurrency problems. Once a data manager has picked up an item from a queue, it deletes the entry to prevent another data manager from processing the same file.

The Queue Process

Introduction

In the queue process, one or more upstream data managers write entries to a queue and one or more downstream data managers read from the queue to find files to process.

Stages in the queue process

This table describes the stages in the queue process.

Stage	Description
1	An upstream data manager processes files and writes information about the files to a specified queue.
2	A downstream data manager reads the queue to locate files it is configured to process.
3	The downstream data manager processes the files in the order indicated by the priority assigned to them. **Comment** If multiple entries in the queue have the same priority, the data manager processes the oldest entry first.
4	The downstream data manager deletes the file entries it has processed from the queue.
5	The process starts over with stage 1.

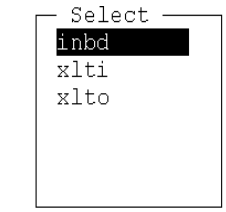
Queue Select Screen

Introduction

The queue Select screen displays the names of the existing queues.

Select screen

This illustration shows an example of the queue Select screen.




```
<CR>: Select
F2: Add a Queue
F3: Del Q Reference
F4: Del Entire Q
F9: Exit
```

Selecting a queue

To select a queue from the list, use the up and down cursor keys.

Select screen function keys

This table describes the function keys on the Select screen.

Key	Function
<CR>	Opens the Queue File screen for the selected queue.
F2	Displays the Add screen for adding a new queue.
F3	Deletes the queue name from the list of queues, but does not delete the queue.
F4	Deletes the queue.
F9	Exits the Queue utility.

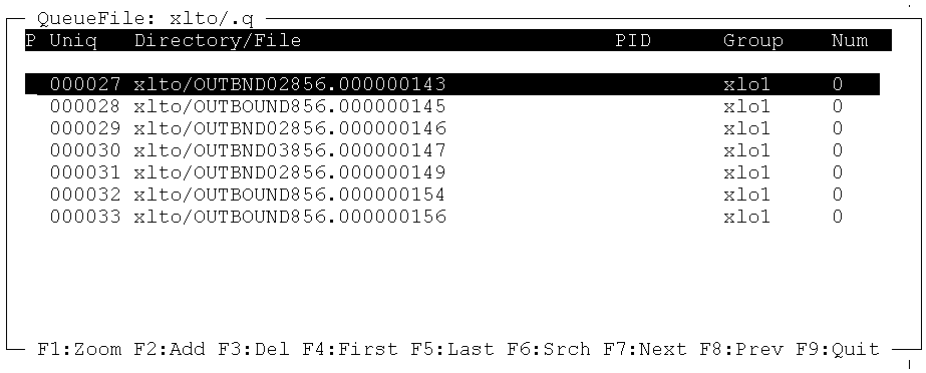
Queue File Screen

Introduction

The Queue File screen displays the entries in a queue.

Queue File screen

This illustration shows an example of the Queue File screen.



Queue File screen fields and functions

This table lists the fields of the Queue File screen and their functions.

Field	Function
P	Defines the processing priority assigned to the file, where 0 represents the highest priority and 9 represents the lowest priority. **Comment** If the processing priority field is blank (as in the above example), the system uses 9 as the default priority value.
Uniq	Displays the unique number Sterling Gentran:Server assigned to this queue entry.
Directory/File	Specifies the name of the directory and the name of the file. **Comment** The screen field size is limited. To see the full path, press F1.
PID	Not used.

(Contd) Field	Function
Group	Defines the name of the resource group to which the file is assigned.
Num	Specifies the maximum number of queue entries allowed for the resource group. Use zero (0) for unlimited entries.
Do you wish to add an entry now? (Y/N)	Displays the Add screen for adding an entry to the queue.

Queue File screen function keys

This table describes the function keys on the Queue File screen.

Key	Function
F1	Displays the entire path and file name for the selected entry at the bottom of the screen.
F2	Displays the Add screen for adding an entry to the queue.
F3	Deletes the selected entry from the queue.
F4	Displays the first page of entries in the queue.
F5	Displays the last page of entries in the queue.
F6	Open the Search screen for starting a search for an entry.
F7	Displays the next page of entries in the queue.
F8	Displays the previous page of entries in the queue.
F9	Exits the screen.

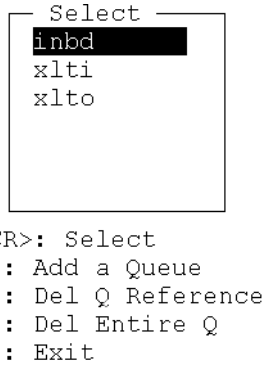
How to Create a Queue

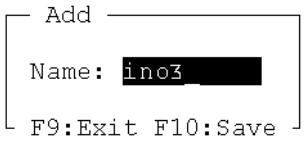
Introduction

Sterling Gentran:Server provides a queue utility that enables you to create a new queue.

Procedure

Use this procedure to create a queue.

Step	Action
1	Go to the host main menu.
2	Select **Queue** from the Util menu. **System Response** Sterling Gentran:Server displays the Select screen. This screen lists the existing queues.
3	Press F2 to add a new queue.  **System Response** Sterling Gentran:Server displays the Add screen.

(Contd) Step	Action
4	Type the name of the queue.  **Comment** You can use up to 9 characters in the queue name. Use only numbers or letters. Do not use symbols or spaces.
5	Press F10. **System Response** Sterling Gentran:Server creates the queue and adds the queue name to the Select list.

Maintaining Queue Entries

Overview

In this section

This section contains procedures for maintaining queue entries. It includes these topics:

- How to view entries in a queue
- Add Queue Entry Screen
- How to add an entry to a queue
- How to delete an entry from a queue.

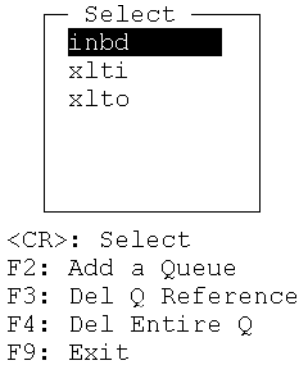
How to View Entries in a Queue

Introduction

If a downstream data manager has not yet processed queued files, the entries remain in the queue. You can view the entries in a queue.

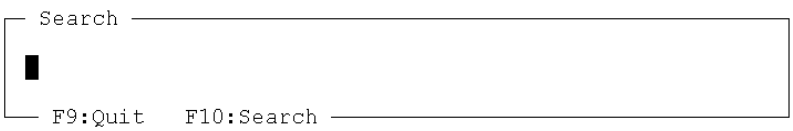
Procedure

Use this procedure to view the entries in a queue.

Step	Action
1	Go to the host main menu.
2	Select **Queue** from the Util menu. **System Response** Sterling Gentran:Server displays the queue Select screen. 
3	Use the cursor keys to select the name of the queue you want to view and then press ENTER (the carriage return key). **System Response** Sterling Gentran:Server displays the Queue File screen for the selected queue.
4	Use the screen's F7 and F8 function keys to navigate through the entries.

Searching for an entry

Use this procedure to search for a specific entry in a queue file.

Step	Action
1	Open the Queue File screen for the queue you want to search.
2	Press F6 to display the entry Search screen. 
3	Complete the **Priority** (P) and **Unique** (Uniq) fields and press F10. **System Response** Sterling Gentran:Server locates and selects the entry that matches the search criteria you entered.

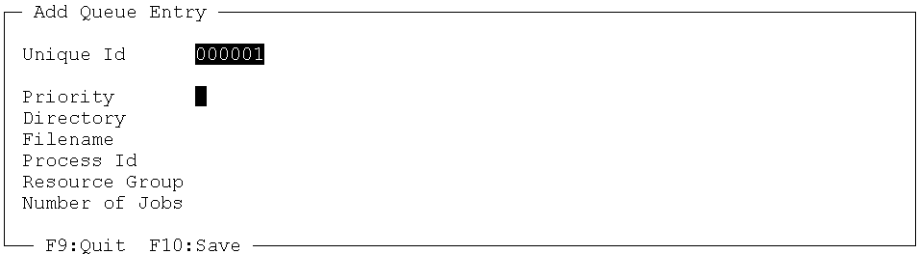
Add Queue Entry Screen

Introduction

The Add Queue Entry screen is used to add entries to a queue.

Add Queue Entry screen

This illustration shows an example of the Add Queue Entry screen.



Add Queue Entry screen fields and functions

This table lists the fields of the Add Queue Entry screen and their functions.

Field	Function
Unique ID	Displays the unique number Sterling Gentran:Server assigned to this file.
Priority	Defines the processing priority for the file, where 0 represents the highest priority and 9 represents the lowest priority.
Directory	Specifies the name of the directory that contains the file.
Filename	Specifies the name of the file.
Process ID	Not used.
Resource Group	Defines the name of the resource group to which the file is assigned.
Number of Jobs	Specifies the maximum number of queue entries allowed for the resource group. Use zero (0) for unlimited entries.

Add Queue Entry screen function keys

This table describes the function keys on the Add Queue Entry screen.

Key	Function
F9	Exits the screen.
F10	Saves the entry.

How to Add an Entry to a Queue

Introduction

There are two ways to manually add an entry to a queue:

- Open the queue and use the add function on the Queue File screen
- Use the **srvr_enq** command line program.

Using the Queue File screen to add a record

Use this procedure to add an entry to a queue.

Step	Action
1	Open the Queue File screen for the queue you want to modify. **Reference** See the How to View Entries in a Queue.
2	**Press F2.** **System Response** Sterling Gentran:Server displays the Add Queue Entry screen.
3	Complete the fields.
4	Press F10 to save the entry.

Using the srvr_enq program to add a record

The **srvr_enq** command line program adds entries to a queue. You can use this command in a Sterling Gentran:Server script.

The command format is:

```
srvr_enq -q<q dir> -g<group> -j<file> -d<file dir>[-p<priority>]
[-n<max num>]
```

This table describes the arguments in the **srvr_enq** command line format.

Argument	Description
-q	The name of the queue.
-g	The name of the resource group to which the file belongs
-j	The name of the file.
-d	The name of the directory that contains the file.
-p	The processing priority (0=highest priority; 9=lowest priority)
-n	The maximum number of files allowed in this queue's resource group. Use zero (0) for unlimited entries. **Note** Data managers do not reference this field.

How to Delete an Entry From a Queue

Introduction

There are two ways to delete a record from a queue:

- Open the queue and use the delete function on the Queue File screen
- Use the **srvr_deq** command line program.

Using the Queue File screen to delete an entry

Use this procedure to delete an entry from a queue.

Step	Action
1	Open the Queue File screen for the queue you want to modify. **Reference** See the How to View Entries in a Queue topic in this chapter.
2	Use the cursor or function keys to locate and select the entry you want to delete.
3	Press F3. **System Response** Sterling Gentran:Server displays a confirmation screen.
4	Type **y** to confirm the deletion. **System Response** Sterling Gentran:Server deletes the entry from the queue.

Using the srvr_deq program to delete an entry

The **srvr_deq** command line program deletes entries from a queue. You can use this command in a Sterling Gentran:Server script.

The command format is:

```
srvr_deq -q<q dir> -j<file> -d<file dir> -g<group>
```

This table describes the arguments in the command line format.

Argument	Description
-q	The name of the queue.
-j	The name of the file.
-d	The name of the directory that contains the file.
-g	The name of the resource group to which the file belongs

Maintaining Queues

Overview

In this section

This section contains procedures for maintaining queues. It includes these topics:

- How to remove a queue from the Select list.
- How to delete a queue.

How to Remove a Queue from the Select List

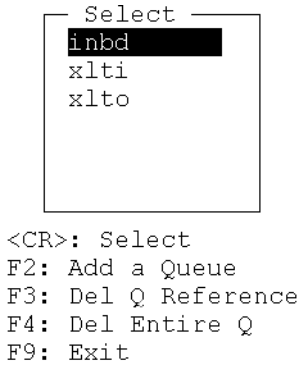
Introduction

You can remove a queue from the Select list. Removing the queue name from the list does not delete the queue.

Reference
If you want to delete the queue, see the How to Delete a Queue.

Procedure

Use this procedure to retain the queue but remove the queue name from the Select list.

Step	Action
1	Go to the host main menu.
2	Select **Queue** from the Util menu. **System Response** Sterling Gentran:Server displays the queue Select screen. 
3	Use the cursor keys to select the name of the queue you want to remove.
4	Press **F3**. **System Response** Sterling Gentran:Server removes the queue name from the Select screen.

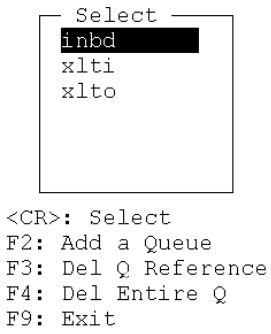
How to Delete a Queue

Introduction

You can delete a queue if you no longer need it.

Procedure

Use this procedure to delete a queue.

Step	Action
1	Go to the host main menu.
2	Select **Queue** from the Util menu. **System Response** Sterling Gentran:Server displays the queue Select screen. 
3	Use the cursor keys to select the name of the queue you want to delete.
4	Press F4. **System Response** Sterling Gentran:Server deletes the queue and removes the queue name from the Select screen.

CHAPTER
5

Working with Scripts

Contents

Overview
- Introduction . . . 3
- Sterling Gentran:Server Scripts . . . 5
- The Script Manager . . . 6
- Ways to Use Scripts . . . 7

Parts of a Sterling Gentran:Server Script
- Introduction . . . 8
- ENVIRON Group . . . 11
- LOCKS Group . . . 13
- DATA Group . . . 17
- DPROCS Group . . . 19
- STEPS Group . . . 20
- PROCS Group . . . 23
- RESULTS Group . . . 27

Creating Sterling Gentran:Server Scripts
- Overview . . . 30
- The Flow of Work . . . 32
- Guidelines for Writing Scripts . . . 33
- The Script Maintenance Screen . . . 36

Procedures
- How to Add a Script with the Script Editor . . . 38
- How to Add a Script with Another Editor . . . 41
- How to Copy a Script . . . 43
- How to Add Lock File Names to the LOCKS Directory . . . 45

Working With Translation Scripts
- Translation Scripts . . . 46
- How to Create a Translation Script . . . 47

Maintaining Sterling Gentran:Server Scripts
- How to Edit a Script with the Script Editor . . . 50

- How to Delete a Sterling Gentran:Server Script 52

Working with UNIX Mail Scripts

- Overview 54
- How to Add or Edit a UNIX Mail Script 55
- How to Delete a UNIX Mail Script 57

Overview

Introduction

In this chapter

This chapter describes the components of Sterling Gentran:Server scripts and explains how to create, maintain, and use scripts to control processes and carry out commands in your data flows.

Reference

For information about running scripts, see the chapter Running Scripts in this guide.

For information about monitoring scripts, see the chapter Monitoring Processes in this guide.

Key terms

This table lists the key terms used in this chapter.

Term	Description
action line	The line in a script that contains the actual commands you want executed.
comment line	A phrase or sentence in a script that explains the purpose or effect of the line of instructions that follows the comment line.
delimiter symbols	Special characters that designate the type of information on a line in the script.
label line	The line in a script that contains the name that you assign to the action that you want performed.
recover script	A script that examines the processing environment of any scripts that were active at the time your machine stopped.
resource pool	Two or more resources of the same type that can be used interchangeably
script	A set of commands that controls processes or performs some action.
script directory	The directory that contains all Sterling Gentran:Server scripts.

(Contd) Term	Description
script editor	The default editor that Sterling Gentran:Server calls when you create or edit a script.
script library	A file that lists and describes all the Sterling Gentran:Server scripts that have been added to the file.
Script Manager	The Sterling Gentran:Server program that directs the script interpreter to execute the commands in a script.
translation script	A special Sterling Gentran:Server for UNIX script associated with and invoked by a translation data manager. The script calls the translator, **lftran**, and other runtime programs.
UNIX mail script	A UNIX script that you can use to send electronic mail messages based on the results of a Sterling Gentran:Server script operation.

Sterling Gentran:Server Scripts

Definition

Sterling Gentran:Server **scripts** are sets of commands that include:

- UNIX commands
- Names of data files you want used in the commands
- Discrete steps with statements that tell Sterling Gentran:Server what to do.

The procedural statements are in "if, then, else" format.

Script language

Sterling Gentran:Server scripts are written in an IBM scripting language that is easy to learn and use.

Script files are in ASCII text. You can edit them or create new ones with the vi editor, the $EDITOR editor, or another ASCII editor.

Identifying Sterling Gentran:Server scripts

Every Sterling Gentran:Server script has the suffix *.scr*.

Where scripts reside

All scripts are stored in the $EDI_ROOT */script* directory.

What you can do with scripts

Sterling Gentran:Server scripts enable you to control processes and carry out commands automatically.

Example
You can have a delivery agent invoke a script after it processes data.

A script can invoke another script.

Translation scripts

Translation scripts are special Sterling Gentran:Server scripts associated with translation data managers (xltr personalities).

The Script Manager

Introduction

Scripts operate under the control of the Script Manager.

Definition

The **Script Manager** is the program that tells the script interpreter to execute the statements in a script. The Script Manager resides in the *./bin* directory.

You can run the Script Manager from:

- The command line
- A data manager
- Another Sterling Gentran:Server script
- A UNIX shell script
- The Permanent Schedule.

Handles script errors

The Sterling Gentran:Server Script Manager returns a non-zero result code if a Sterling Gentran:Server script fails. On success, it returns a zero. You can use the return code to have the Script Manager react in a particular way when a script fails.

Example
You can have the Script Manager invoke a UNIX shell script to send UNIX mail notification when a script fails. You can extend mail notification to include sending beeper messages, routing messages to a printer or terminal, or sending messages to an alternate operating system message facility.

Ways to Use Scripts

Introduction	You can use Sterling Gentran:Server scripts to control many processes and commands. This topic suggests a number of ways to use Sterling Gentran:Server scripts.
Start and terminate programs and processes	Use Sterling Gentran:Server scripts to invoke data managers, the translator, programs, other Sterling Gentran:Server scripts, and shell scripts.
Communicate with hosts	Create a Sterling Gentran:Server script to pull host data through an Ethernet or other host connection.
Communicate with Value Added Networks	The Communications Toolkit contains communication script models to connect with three of the most popular Value Added Networks (VANs) and network interfaces. You can use Sterling Gentran:Server scripts to invoke the communication scripts provided in the Communications Toolkit.
Manage non-shareable resources	You can use scripts to control access to a non-shareable resource such as a modem.
Send notification of an event	You can have scripts automatically generate a notification of an event, such as a communications failure.
Handle files	You can use Sterling Gentran:Server scripts to search for files that match certain conditions or patterns, move files into a directory, convert and copy files, and perform other types of file handling operations.

Parts of a Sterling Gentran:Server Script

Introduction

Common parts

Sterling Gentran:Server scripts have basic parts in common. These are:

- Delimiter definition lines
- Sections
- Groups.

The use of these parts makes creating a script easier.

Delimiter definition line

The first line of every Sterling Gentran:Server script defines special characters, called **delimiters**, used in the script. Delimiters designate the type of information on the line.

This table describes the default delimiter symbols and their functions.

Symbol	Function
#	Designates a comment line, which is a phrase or sentence that explains the purpose or effect of the line that follows the comment. Use as many comment lines as needed in the script.
!	Designates a group, which is an element of a script.
:	Designates a label line, which contains a title for a set of commands or actions in the group.
^	Includes the named file that follows the character.
$	Indicates a reference to a variable defined in the ENVIRON group.

WARNING

All the IBM-supplied model scripts use these delimiters. Changing these delimiters can cause your scripts to fail or operate incorrectly.

Sections

A Sterling Gentran:Server script can have up to four sections:

- Initialization
- Description
- Procedure
- Conclusion.

Groups

Each section in a script contains one or more **groups**, each of which has a specific function. Some groups are required for the script to operate; others are optional.

Section and group functions table

This table lists the groups in each section, indicates whether the group is required or optional, and describes the function of the group.

Section	Group Name	Req./ Opt.	Function
Initialization	ENVIRON	Opt.	Initializes the operating environment (environment variables or parameters) for the rest of the script.
	LOCKS	Opt.	Selects non-shareable resources required for the script to operate.
Description	DATA	Opt.	Describes the text and data files that you want assembled.
	DPROCS	Opt.	Describes the order of assembly of the text and data files in the DATA group. The reserved word build in the PROCS group assembles the elements.
	STEPS	Req.	Describes the commands in the PROCS group that you want executed.
Procedure	PROCS	Req.	Determines the order of execution of the steps in the STEPS group and file construction of DPROCS.
Conclusion	RESULTS	Opt.	Determines the success or failure of the script based upon steps in the STEPS group and the labels within the LOCKS group.

Group general format

This is the general format of a group.

```
!GROUP_NAME
:label_name
action line one
action line two
:another_label
another action line
```

Parts of the general format

This table describes the parts of the general format.

Part	Description	Maximum characters
group name	The name of the group: ENVIRONS, LOCKS, STEPS, DATA, DPROCS, PROCS, RESULTS	-
label name	The title of a set of commands. **Example** This is the label name that represents running the Cleo shell script: :run_cleo	40
action line	The command you want executed or action you want performed. **Example** This is the action line that runs the Cleo shell script: sh $cleoDir/$VAN.run	240

ENVIRON Group

Introduction

The ENVIRON group defines the values of the variables used in the script. These variables are the same types of variables you would include in a shell.

Reference
For information about using variables in a shell, see your UNIX documentation.

Execution order

Sterling Gentran:Server always executes the ENVIRON group first.

General format of the ENVIRON group

This is the general format of the ENVIRON group.

```
!ENVIRON
:var_label
varname=var_value
:another_var_label
varname=var_value
```

Parts of the general format

This table describes the parts of the general format.

Part	Description
ENVIRON	The name of the group.
var_label	The title of a set of variables. **Example** This is the label name for the VAN variables: `:VAN_var`
varname=var_value	The variable and the value to which it is set. **Example** In this example, the variable errorDir, which represents the error directory, is set to commerror. `errorDir=commerror`

ENVIRON group example

This is an example of the ENVIRON group.

```
!ENVIRON
#
#
:VAN_vars
VAN=CommerceNet
#
#
:directory_vars
sendFilesDir=to_CommerceNet
putEdiDir=edii
#
#
:error_vars
errorDir=commerror
scriptName=CommerceNet
#
#
:cleo_vars
cleoDir=cleo/CommerceNet
cleoJob=CommerceNet.job
#
#
```

Explanation of example

In the preceding example:

- The VAN variable is set to **CommerceNet**.
- There are two directory variables. The **sendFilesDir** variable is set to **to_CommerceNet**. The **putEdiDir** variable is set to **edii**.
- There are two error variables. The **errorDir variable** is set to **commerror**. The **scriptName** variable is set to **CommerceNet**.
- There are two cleo variables. The **cleoDir** variable is set to **cleo/CommerceNet**. The **cleoJob** variable is set to **CommerceNet.job**.

Referencing variables in the script

Other parts of the Sterling Gentran:Server script can reference the variables defined in the ENVIRON group.

Referenced variables begin with a "$" prefix, just as in a shell. Variables are case sensitive. Any child process the script spawns also uses these variables.

LOCKS Group

Description

The LOCKS group is used to lock non-shareable resources that are required to operate the script.

When a script with LOCKS is invoked, the script determines whether another Sterling Gentran:Server script is using the resource. If the resource is not available, the script:

- Releases the resources it was able to lock
- Sleeps for a specified time
- Tries again after the sleep period.

If the script cannot lock all the necessary files after a specified number of attempts, the script fails.

Definition of non-shareable resources

Non-shareable resources are files, programs, or hardware devices that only one activity can use at a time.

Example
A modem is an example of a non-shareable resource.

LOCKS group execution order

The LOCKS group executes after the ENVIRON group and before the PROCS group.

Types of LOCKS

There are two types of LOCKS:Type 1 and Type 2.

- Type 1 locks lock the file only.
- Type 2 locks extract information first and then lock the file. If the file is empty, the type 2 lock fails.

How the lock types are used

This table describes how each lock type is used.

Lock	Use

Type 1	Used to lock processes.
	Format
	`:type1lock`
	`1 lock1_filename`
	Example
	Use a type 1 lock to lock a script because you want the script to finish running before another process restarts it.

Type 2	Used to lock devices such as modems, tape drives, and diskette drives.
	Format
	`:type2lock`
	`2 lock2_filename <variable>`
	This format assigns the contents of the first action line in the lock file to the variable name. The variable is set to the contents of the lock2_filename.
	Example
	`:type2lock`
	`2 lock2_filename DEVICE`
	Use a type 2 lock to lock a modem because you want to prevent another Sterling Gentran:Server script from using the same device.

The LOCKS directory and resource file names

The LOCKS directory contains the file names of the devices and files used with locks.

- The type 1 lock files in this directory normally are empty because only the file name is necessary.
- Type 2 lock files usually contain the device name. The path name to a file represents the resource.

To identify a resource in the LOCKS group, you must assign a file name to it. This is because a script can access the resource through the assigned file name only.

Examples

These are examples of LOCKS statements in a script.

```
1 ./LOCKS/PROCESS_1
2 ./LOCKS/MODEM_1 MODEM
```

General format of the LOCKS group

This is the general format for the LOCKS group.

```
!LOCKS -dtime -rretries
:type1lock
1 file1_name
:type2lock
2 file2_name <variable>
```

This table describes the parts in the general format.

Part	Description
LOCKS	The name of the group.
-dtime	This is the time, in seconds, that the script sleeps before it makes another lock attempt if a resource is not available. The default is 5 minutes (300 seconds).
-rretries	The number of times that the script can attempt to lock the resource before the script fails. The default is 12 retries.
type1lock	The title of the set of type 1 lock commands.
1 file1_name	The lock type, 1, followed by the file name that represents the resource.
type2lock	The title of the set of type 2 lock commands.
2 file2_name <*variable*>	The lock type, 2, followed by the file name that represents the resource and the variable that holds the contents of the lock file. **Example** `2 file2_name DEVICE`

LOCKS group example

This is an example of the LOCKS group.

```
!LOCKS -d120 -r5
:modems
## Lock the CLEO modem lines from other Sterling Gentran:Server
scripts.
2 ./LOCKS/MODEM1 MODEM
2 ./LOCKS/MODEM2 MODEM
#
#
:VAN
## Only one session to $VAN running at a time
1 ./LOCKS/CommerceNet
#
```

Resource pools

A **resource pool** consists of two or more resources of the same type that Sterling Gentran:Server can use interchangeably. When you use a resource pool, the script locks the first free file or device available under that label.

Example

A common example of resource pool use is modem access. If you have a limited number of modems available and multiple scripts need those modems, you can control access through a modem LOCKS pool.

This is an example of defining a resource pool for modems.

```
:modems
## Lock the CLEO modem lines from other Sterling Gentran:Server
scripts.
2 ./LOCKS/MODEM1 MODEM
2 ./LOCKS/MODEM2 MODEM
2 ./LOCKS/MODEM3 MODEM
```

Reference

The Sterling Gentran:Server scripts in the script library contains examples of using resource pools.

DATA Group

Description

A **DATA group** describes the text and files that you want assembled from more than one source. It is a way to put files and text together for a specific purpose. The DATA group can include shell commands invoked in later portions of the script.

Examples of use

A DATA group is often used for Value Added Networks that require a file prefix. You can include the prefix in a DATA group so that the script puts the prefix with the file when it processes the file.

You can also have a DATA group use a file as input for some other process.

Relationship to DPROCS and PROCS groups

DATA groups work with DPROCS and PROCS. This table describes the functions of each group.

Group	Function
DATA	Defines the text and files you want assembled.
DPROCS	Describes the order in which you want to assemble the text and files.
PROCS	Issues the assembly command (**build)** to assemble the text and files.

General format of DATA group

This is the general format for a DATA group.

```
!DATA
:data_label_name
Text
^include_file_name
More text
:another_data_label
More text
$script_name
The last text line
```

Parts of the general format

This table describes the parts of the general format.

Part	Description
DATA	The name of the group.
data_label_name	The title of the set of text and files you want assembled.
Text	Text you want included in the file. The caret character (^) signals that this is text to be included. You can include variables in the text.
include_file_name	File you want included. You can use the full path name to the file.
More text	Additional text you want included in the file. You can include variables in the text.
another_data_label	The title of another set of text and files you want assembled.
$script_name	A reference to a variable defined in the ENVIRON group.

DATA example

This is an example of how the DATA group is used to define the text and files you want assembled.

```
!DATA
:ftpscript
## Text that will be built into the $ftpScript
>ScriptName.log 2>/dev/null
ftp -nv $remoteHost<<EOF>./$scriptName.log 2>&1
user $hostDir/$hostFile $toDir/$toFile
bye
EOF
```

DPROCS Group

Description

The **DPROCS group** describes the order in which you want the text and files in the DATA group assembled. The DPROCS group can include shell commands invoked in later portions of the script.

General format of DPROCS group

This is the general format of the DPROCS group.

```
!DPROCS
:build_label
data_label1
data_label2
```

Parts in the general format

This table describes the parts in the general format.

Part	Description
DPROCS	The name of the group.
build_label	The title of the set of assembly instructions.
data_label1	The label name of the item in the DATA group that is to be first in the assembly order.
data_label2	The label name of the item in the DATA group that is to be next in the assembly order.

DPROCS example

This is an example of how DPROCS is used to specify the assembly order of items in the DATA group. The DATA groups **ftpscript** and **build_run** are to be assembled, with **ftpscript** coming first.

```
!DPROCS
#
#
:ftpbld
ftpscript
build_run
#
#
```

STEPS Group

Description

The **STEPS group** contains the actual commands executed during the operation of the script. An action line in the STEPS group can be virtually any command line statement, UNIX shell script, or program.

Relationship to PROCS group

STEPS labels are referenced in the PROCS group. The PROCS group issues each action line under a step label to the operating system. The order of issue is the order in which the actions appear under the label.

General format of STEPS group

The STEPS group consists of step_labels followed by one or more action lines that the script will issue as commands to the operating system.

This is the general format of the STEPS group.

```
!STEPS
:step_label
UNIX commands
More UNIX commands
:step_label
UNIX commands
```

Parts in the general format

This table describes the parts in the general format.

Part	Description
STEPS	The name of the group.
step_label	The title of the set of commands.
UNIX commands	The UNIX command line statement, UNIX shell script, or program you want executed.
More UNIX commands	Additional UNIX command line statements, UNIX shell scripts, or program you want executed.

STEPS group example

This is an example of the commands in a STEPS group.

```
!STEPS
#
```

```
#
:start
date +%m%d%H%M > $cleoDir/.Date
> script/$scriptName.old
#
#
:check_if_need_send
## See if there are any files to send to $VAN.
ls $sendDir/* > $sendFiles/sendlist.$VAN 2>/dev/null
test -s $sendDir/sendlist.$VAN
#
#
:combine_files
cat `cat $sendDir/sendlist.$VAN` > $cleoDir/send.$VAN
#
#
:run_cleo
##  Cleo shell script is executed below.
sh $cleoDir/$VAN.run
#
#
:check_if_tried_send
## See if we tried to send any files.
test -s $sendDir/sendlist.$VAN
#
#
:check_if_send_failed
## See if the send.$VAN file is gone. This means it was
## sent successfully.
test -s $cleoDir/send.$VAN
#
#
:remove_sent_files
## Remove the files that were sent to the VAN.
rm `cat $sendDir/sendlist.$VAN` 2>/dev/null
#
#
:check_if_received_files
## Check to see if we received files from the VAN.
ls $cleoDir/edidata.* > $cleoDir/recvlist.$VAN 2>/dev/null
test -s $cleoDir/recvlist.$VAN
wc -w $cleoDir/edidata.* | grep -v ' 0 ' > $cleoDir/recvcount.$VAN
test -s $cleoDir/recvcount.$VAN
#
#
:handle_files
## Place incoming files from the VAN into the $putEdidir directory.
cat `cat $cleoDir/recvlist.$VAN` > $putEdiDir/recv.$VAN.`cat
$cleoDir/.Date`
cat `cat $cleoDir/recvlist.$VAN` > legal/recv.$VAN.`cat $cleoDir/
.Date`
#
#
:cleanup
## Cleanup all temporary and sent/received files.
```

```
rm $sendDir/sendlist.$VAN 2>/dev/null
rm `cat $cleoDir/recvlist.$VAN ` $cleoDir/recvlist.$VAN 2>/dev/null
rm $cleoDir/edidata.* $cleoDir/recvcount.$VAN 2>/dev/null
```

PROCS Group

Description

The **PROCS group** specifies the order in which the script executes the steps defined in the STEPS group and the building of the DPROCS files.

The PROCS group consists of proc_labels (names), each of which delineates a single simple if - then - else construct:

- The "if" part of the construct is followed by a step_label or a reserved word such as **build**, **release**, or **putenv**. When the proc_label is executed, the action lines included under step_label (as defined in the STEPS group) are executed.
- The "then" and "else" parts of the construct are followed by proc_labels.

PROCS group execution order

The PROCS group is invoked after the LOCKS group.

General format of the PROCS group

This is the general format of the PROCS group.

```
!PROCS
:proc_label
if step_label then zero_proc_label else non_zero_proc_label
```

Parts of the general format

This table describes the parts of the general format.

Part	Description
PROCS	The name of the group.
proc_label	The title of the set of instructions.
if step_label then zero_proc_label else non_zero_proc_label	The if-then-else statement that describes what to do if all the steps in the step_label succeed and what to do if any of the steps fail. If all the steps in the step_label were successful, the zero_proc_label is executed. If any of the steps under the step_label failed, the non_zero_proc_label is executed. **CAUTION** **This scenario assumes the script interpreter finds the proc_labels in PROCS. If the script interpreter is unable to find the label, it terminates the script and notes the failure.**

PROCS reserved words and functions

This table describes the reserved words and functions in the PROCS group.

Reserved Word or Function	Function
if	The part of the PROCS action line syntax that tests the truth of the step_label that follows “if.”
then	The part of the PROCS action line syntax that tells the operating system what to do if the condition in the “if” portion of the syntax is true.
else	The part of the PROCS action line syntax that tells the operating system what to do if the condition in the “if” portion of the syntax is not true.
end	The part of the PROCS action line syntax that tells the operating system to stop the script.

(Contd) Reserved Word or Function	Function
build(DPROC_label,file_name)	Assembles the file_name according to the list under the DPROC_label. **Example** `if build(build_label1,newfile) then...`
release(LOCKS_label)	Unlocks the file or resource locked under LOCKS_label. **Example** `if release(modem_pool) then...`

compare	Compares strings or system variables. If the values are equal, the compare is successful.
	Example
	`if compare($envar,"string value") then...`
	or
	`if compare($envar1,$envar2) then...`
	Note
	In the above example, “string value” represents a literal value.

putenv(ENV_VAR=some_value)	Sets the environment variable ENV_VAR to some_value. **Example** `if putenv(HOST_NAME=edisrv02) then...`

CAUTION

If the build, release, compare, or putenv reserve words do not complete their tasks, the script fails. This failure overrides the analysis generated from the RESULTS section of the script. If one of these functions or reserve words fails, the system sends you mail via the script in the mail_proc directory.

PROCS group example

This is an example of the PROCS group.

```
!PROCS
#
#
:start
if start then check_if_need_send else check_if_need_send
#
```

```
#
:check_if_need_send
if check_if_need_send then combine_files else build_run
#
#
:combine_files
if combine_files then build_run else build_run
#
#
:build_run
if build(build_run,cleo/CommerceNet/CommerceNet.run) then run_cleo
else end
#
#
:run_cleo
if run_cleo then check_if_tried_send else check_if_tried_send
#
#
:check_if_tried_send
if check_if_tried_send then check_if_send_failed else
check_if_received_files
#
#
:check_if_send_failed
if check_if_send_failed then check_if_received_files else
remove_sent_files
#
#
:remove_sent_files
if remove_sent_files then check_if_received_files else
check_if_received_files
#
#
:check_if_received_files
if check_if_received_files then handle_files else cleanup
#
#
:handle_files
if handle_files then cleanup else end
#
#
:cleanup
if cleanup then end else end
#
#
```

RESULTS Group

Description

The labels in the **RESULTS group** and any function invocations such as **putenv** or **build** determine the final result of the script. The RESULTS group determines how to report the final result of the script.

The RESULTS group performs these functions:

- Determines the overall success or failure of the script
- Defines the mail message sent for step_labels in STEPS, lock labels (type1lock or type2lock) in LOCKS, or for var_labels in ENVIRON when the script fails.

If the script fails, the UNIX mail script routes the mail message to the mail ID specified in the UNIX mail script.

General format of the RESULTS group

This is the general format of the RESULTS group.

```
!RESULTS
:steplabel
Mail message
```

Parts in the general format

This table describes the parts in the general format.

Part	Description
RESULTS	The name of the group.
steplabel	The step_label, type1lock, type2lock, or var_label.
Mail message	The text of the mail message that you want sent if the script fails. **CAUTION** **The system combines the text messages that follow failed labels and sends them out via the UNIX mail script. The messages are also written to the script log.**

Choosing step_labels for the RESULTS group

You select labels from the ENVIRON, STEPS, and LOCKS groups that are significant to the operation of the script. The results of the actions in these labels contribute to the final result of the script. Any labels not included in RESULTS do not contribute to the overall success or failure of the script.

Example

Failure to lock a modem is a significant event for the script, so you should include a modem lock or modem lock pool label from the LOCKS group in the RESULTS section. This example shows the label and the mail message text sent to alert you of the failure.

```
:modem_pool
Could not obtain a modem to lock in $scriptName.
Check all modems and permissions on ports.
```

RESULTS group example

This is an example of the RESULTS group.

```
!RESULTS
#
#
:modems
Could not obtain a modem to lock in $scriptName.
Check all modems and permissions on ports.
#
#
:VAN
Could not obtain a lock on $VAN file. See if modem is
stuck talking to $VAN or if another script is locked up
while talking to $VAN.
#
#
:run_cleo
Cleo failed to communicate with $VAN.
Please see files in $EDI_ROOT/commerror directory.
#
#
:combine_files
Could not combine files to send to the VAN.
Please check the permissions on the
to_$VAN directory and the files in the directory.
#
#
build_run
Could not create the $cleoDir/$VAN.run file.
Please check the permissions on the $cleoDir
directory and on the $cleoDir/$VAN.run file.
#
#
:remove_sent_files
Could not remove the files sent to the VAN. Check permissions
on $sendFilesDir.
#
```

```
#
:handle_files
Could not move files into $putEdiDir. Check permissions
on the $putEdiDir directory.
```

Creating Sterling Gentran:Server Scripts

Overview

Two ways to create scripts

There are two basic ways to create a script:

- Write a totally new one
- Copy an existing one and modify it to suit your needs.

Copying an existing script that does most or all of the things you want the new script to do is the easiest way to create a new script.

Sample scripts

The *$EDI_ROOT/script* directory contains several sample scripts. You can copy and modify these samples to develop your own scripts.

Note
If you cannot find an appropriate script to copy and do not have the resources to develop a new script, contact IBM Customer Support.

Permissions

The file permissions for a script are normally rwx rwx r-x.

- The person who is likely to run the script should own the script.
- The root user should not own scripts, because no one else is in the root user's group.

Selecting an editor

You can use the editor set in the $EDITOR environment variable to create or edit scripts. If the $EDITOR variable is not set, the system uses the vi editor.

You can use an editor other than the one invoked through Sterling Gentran:Server because script files are ASCII text.

The ./script directory

All Sterling Gentran:Server scripts must reside in the **script directory**, *./script.* If the script is not in the *./script* directory, Sterling Gentran:Server cannot execute it.

- When you use the editor accessed through Sterling Gentran:Server, Sterling Gentran:Server stores the file in the *./script* directory.

- If you prefer to edit scripts in another editor, you can manually copy the scripts in and out of the *./script* directory to work on them.

Script library

The **script library** contains the name and description of every Sterling Gentran:Server script. Regardless of how you create a script, you must add the script's name and description to the library list before or after you create the script.

Note
If you create a new script by copying an existing one from the Script Maintenance screen, Sterling Gentran:Server adds the new script name to the library list for you.

The Flow of Work

Script creation process

This table describes the process of creating a script.

Stage	Description
1	Break down the job into steps. **Example** 1. Find a file to process. 2. Translate the file. 3. Transfer the file.
4	Break the steps into executable commands.
5	Identify error conditions.
6	Create any directories or files that the script will use.
7	Use the shell to test commands and error conditions. **Comment** You can use CTRL+A to access the shell.
8	Write the script. You can use the script editor in the Sterling Gentran:Server **Script Maintenance** facility, or write the script with another editor and then move the script into the *./script* directory. **Reference** See your UNIX documentation for UNIX commands to use in your script. **Comment** If you use another editor, make sure that you: ◗ Add the *.scr* suffix to the script file name ◗ Use the Script Maintenance screen to add the script to the script library.
9	Test the script. Execute it from the Script Maintenance screen, or run the Script Manager from the shell with the command **smgr -s<*scriptname*>**. (Omit the *.scr* suffix in the file name. DO NOT leave a space between -s and the file name.)

Guidelines for Writing Scripts

Introduction

This topic contains rules and tips for writing scripts.

Delimiters

Make sure that the comment delimiter (#) appears in the first position (column one) of the script line.

Do not change the standard delimiters. Doing so can cause your scripts to fail or operate incorrectly.

Blank lines

Do not leave any blank lines except in the RESULTS or DATA group.

Groups

Use each group name only once in a script.

After the delimiter line, place the groups in any order. The order in which the groups appear in the script does not affect the execution order of the groups.

Labels

A group can have one or more label names.

- Each label name within a group must be unique.
- Label names cannot contain spaces.
- The maximum number of characters for the label name is 39.

Variables

To refer to a variable in the script, you must begin it with a "$" prefix, just as you would in a shell.

To use the variable reserve character ($) and the include file character (^) literally in the text, begin them with a backslash (\) prefix. To include a backslash, enter two backslashes (\\).

Format

```
:example_label
This text will include the contents of $VARIABLE_NAME.
This text will include the dollar sign \$ and the caret sign \^
This line will have just one backslash \\.
```

Example 1

```
!DATA
:build_run
cd $cleoDir
./3780Plus -j $cleoJob -S -B 9600 -D $MODEM -LA 1>/dev/null 2>/dev/
null
##
##
```

Example 2

```
!DATA
:ftpscript
## Text that will be built into the $ftpScript.
>$scriptName.log 2>/dev/null
ftp -nv $remoteHost << EOF >./$scriptName.log 2>&1
user $hostLogin $hostPassword
runique
get $hostDir/$hostFile $toDir/$toFile
bye
EOF
```

Tips

This table lists several tips for having your script perform specific activities.

IF you want to have the script…	Then…

Remove input files and temporary files	Include a cleanup step.
	Example
	`:cleanup`
	`rm $tempdir/xlcntl.err`

Assign a time stamp to the files it processes	Use a UNIX regular expression with the date command. **Example** `mv udf bad. `date + %y%m%d%H%M%S``
Release all locks during the execution of the script	Use the reserved word “release” to unlock the file that the script locked. **Example** `if release(modem_pool) then...`
Run through **cron** (the UNIX daemon that starts other processes at scheduled times) with the appropriate environment	Create a hidden file named *.edi_env* in EDI_ROOT. Include in the file all the necessary environment settings required for Sterling Gentran:Server processing. When you add the script to the Permanent Schedule, the script will run under the parameters defined in the *.edi_env* file.

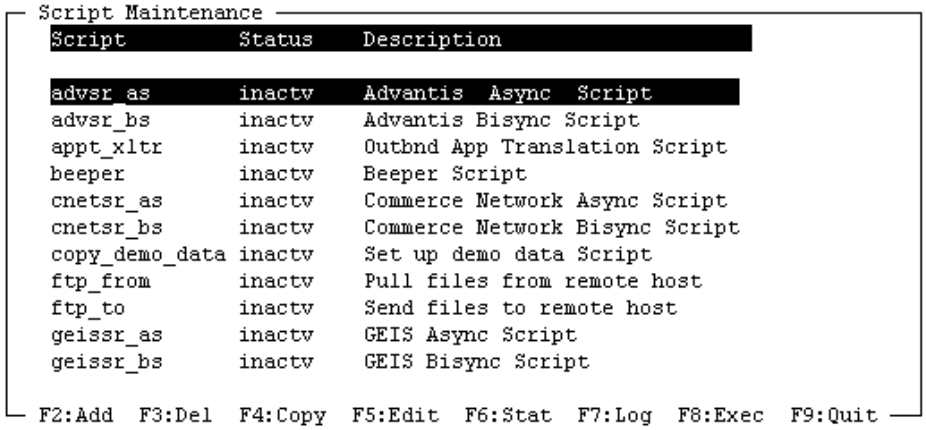
The Script Maintenance Screen

Introduction

The Script Maintenance screen is the starting point for script creation and maintenance activities.

Script Maintenance screen

This illustration shows an example of the Script Maintenance screen.



Script Maintenance screen fields and functions

This table lists the fields of the Script Maintenance screen and their functions.

Field	Function
Script	Defines the name of the script.
Status	Displays the current status of the script: ◗ Inactv means the script is not running ◗ Active means that the script is running.
Description	Describes the script.

Script Maintenance screen function keys

This table describes the function keys on the Script Maintenance screen.

Key	Function
F2	Displays the screen used to add the script name and description to the script library.
F3	Deletes the selected script.
F4	Copies the selected script and displays a screen that enables you to name the copy.
F5	Starts the edit function to enable you to modify the selected script or the script's UNIX mail script.
F6	Updates the value in the Status field of each script.
F7	Displays the log file of the selected script.
F8	Executes the selected script.
F9	Exits the Script Maintenance screen.

Procedures

How to Add a Script with the Script Editor

Introduction

This topic explains how to add a script to the script library and then create the script with the script editor. The script editor is set in the $EDITOR environment variable. If the variable is not set, Sterling Gentran:Server uses **vi** as the default editor.

Reference
See the *IBM® Sterling Gentran:Server® for UNIX Getting Started Guide* for information about setting environment variables.

WARNING

Do not use this method to create translation scripts. See the How to Create a Translation Script topic in this chapter for the correct procedure.

Creating a script

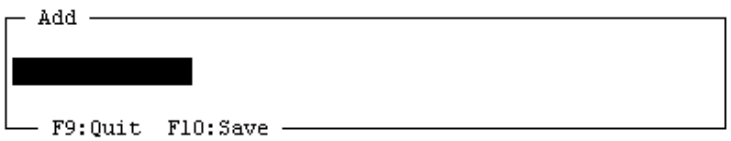
Use this procedure to create a new script.

Step	Action
1	Select **Script** from the host main menu. **System Response** Sterling Gentran:Server displays the Script Maintenance screen, which lists all the scripts in the script library.

```
Script Maintenance
Script          Status    Description

advsr_as        inactv    Advantis  Async  Script
advsr_bs        inactv    Advantis Bisync Script
appt_xltr       inactv    Outbnd App Translation Script
beeper          inactv    Beeper Script
cnetsr_as       inactv    Commerce Network Async Script
cnetsr_bs       inactv    Commerce Network Bisync Script
copy_demo_data  inactv    Set up demo data Script
ftp_from        inactv    Pull files from remote host
ftp_to          inactv    Send files to remote host
geissr_as       inactv    GEIS Async Script
geissr_bs       inactv    GEIS Bisync Script

F2:Add  F3:Del  F4:Copy  F5:Edit  F6:Stat  F7:Log  F8:Exec  F9:Quit
```

(Contd) Step	Action
2	Press F2 to add a script to the script library. **System Response** Sterling Gentran:Server displays the Add screen. 
3	Type the script name (without the *.scr* suffix) in the selected field. Type the script's description in the second field. **Note** The script name field accepts a maximum of 14 characters. The description field accepts a maximum of 30 characters.
4	Press F10 to save the script name and add it to the script library. **System Response** Sterling Gentran:Server displays the Script Maintenance screen.
5	Select the script name you just added and press F5 to continue. **System Response** Sterling Gentran:Server displays a Modify screen with additional function keys.
6	Press F5 to display the script editor. **System Response** Sterling Gentran:Server displays the editor set in the $EDITOR variable (the vi editor by default). **Note** Because the script does not exist, the screen is blank.
7	Write the script.
8	Save the script.

(Contd) Step	Action
9	Exit the editor.
10	Does your script include LOCKS? ◗ If YES, see the How to Add Lock File Names to the LOCKS Directory topic. ◗ If NO, you are finished.

How to Add a Script with Another Editor

Introduction

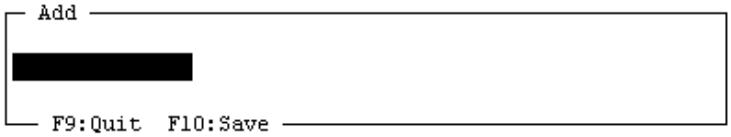
When you create a script with an editor other than the script editor, you must move or copy the script to the *./script* directory and add it to the Sterling Gentran:Server script library list.

WARNING

Do not use this method to create translation scripts. See the How to Create a Translation Script topic in this chapter for the correct procedure.

Creating a script

Use this procedure to create a new script.

Step	Action
1	Open the editor and write the script.
2	Name the script with a *.scr* suffix and save it.
3	Copy or move the script into the *./script* directory.
4	Start Sterling Gentran:Server and access the host menu.
5	Select **Script** from the host main menu. **System Response** Sterling Gentran:Server displays the Script Maintenance screen.
6	Press F2 to add the script to the library list. **System Response** Sterling Gentran:Server displays a screen for the script name and description. 

(Contd) Step	Action
7	Type the script name (without the *.scr* suffix) in the selected field. Type the script's description in the second field. **Note** The script name field accepts a maximum of 14 characters. The description field accepts a maximum of 30 characters.
8	Press F10 to save the detail you added**.**
9	Does your script include LOCKS? ◗ If YES, see the How to Add Lock File Names to the LOCKS Directory. ◗ If NO, you are finished.

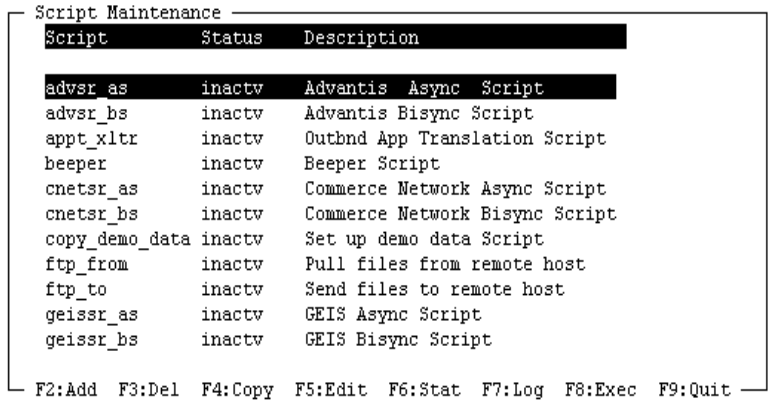
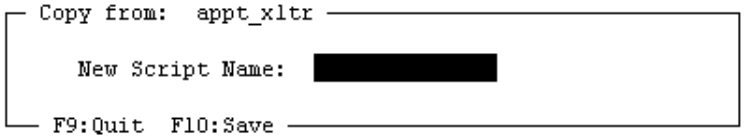
How to Copy a Script

Introduction

A quick way to create a new script is to start with a copy of an existing script. You can copy any script in the library list. When you copy a script on the list and name the new copy, Sterling Gentran:Server adds the new copy to the library list and copies the file contents in the *./script* directory. You can then edit the copy to change the description and the script contents.

Copying a script

Use this procedure to copy a script.

Step	Action
1	Select **Script** from the host main menu. **System Response** Sterling Gentran:Server displays a list of the scripts in the script library and a list of function choices. 
2	Select the name of the script you want to copy.
3	Press F4 to copy the script. **System Response** Sterling Gentran:Server displays a Copy screen. 

(Contd) Step	Action
4	Type the name of the new script, without the *.scr* suffix, into the **Script** field.
5	Press F10 to save the detail you added.

Reference

To edit the script, see How to Edit a Script with the Script Editor.

How to Add Lock File Names to the LOCKS Directory

Introduction

To identify a resource (such as a modem) in the LOCKS group, you must assign a file name to it and enter the file name into the LOCKS directory. A script can access the resource only through the assigned file name. The LOCKS directory holds the file names of the devices and files you want to lock.

File contents

The type 1 lock files in the LOCKS directory are normally empty because only the file name is necessary.

Type 2 lock files usually contain the device name. The path name to a file represents the resource.

Adding lock file names to the LOCKS directory

This table describes how to add lock file names to the LOCKS directory.

Lock	Procedure	Example
Type 1	Use the UNIX touch command at the command line.	`touch LOCKS/MODEM_10` This command creates an empty file in the LOCKS directory with the name MODEM_10.
Type 2	Use the UNIX command echo at the command line.	`echo "/dev/tty0d" > ./LOCKS/MODEM_1` This command sends the device name, /dev/tty0d, to the file MODEM_1 in the LOCKS directory.

Working With Translation Scripts

Translation Scripts

Definition

A **translation script** is a special Sterling Gentran:Server for UNIX script associated with a translation data manager. Its function is to:

- Call the translator, **lftran**
- Call other runtime programs, such as **ediarc, envelope,** and **xlld**
- Ensure that the script runs under the same environment variables as the translation data manager that invoked it
- Clean up if commands fail.

Model translation scripts

Your Sterling Gentran:Server software includes model translation scripts. Each model translation data manager you received (appt, xli1, xli2, xlo1, xlo2) has its own translation script.

How the script is called

The translation data manager starts a process that signals the Script Manager to execute the translation script.

Naming convention

This is the naming convention for a translation script:

`<datamgr_name>_xltr.scr`

Where <*datamgr_name*> is the name of the translation data manager.

Examples

```
xli1_xltr.scr
appt_xltr.scr
```

Specifying the translation script

You specify the name of the translation script in the XL_MODEL_SCR parameter of the translation data manager's initialization file.

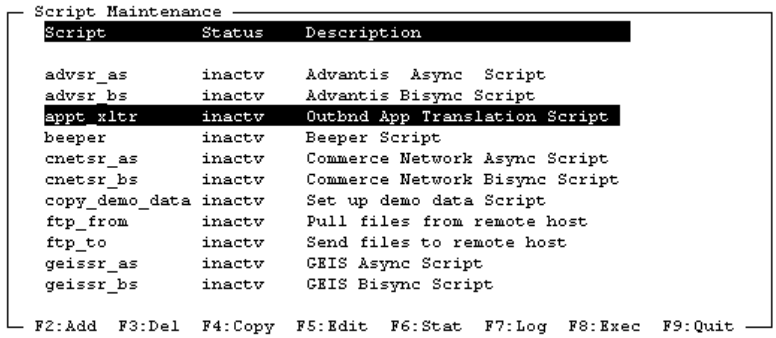
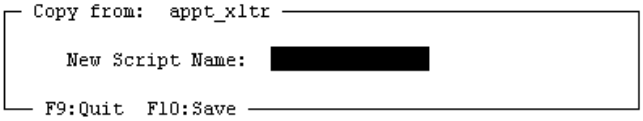
How to Create a Translation Script

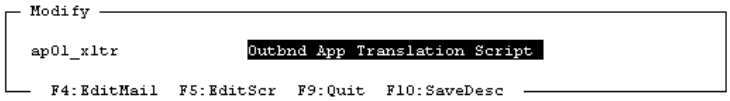
Introduction

To create a new translation script, you must copy an existing translation script and rename it for the new translator data manager. You can then modify the new script to meet your needs.

Creating a translation script

Use this procedure to create a translation script.

Step	Action
1	Select **Script** from the host main menu. **System Response** Sterling Gentran:Server displays a list of the scripts in the script library and a list of function choices.
2	Select the translation script you want to copy. 
3	Press F4 to copy the script. **System Response** Sterling Gentran:Server displays a Copy screen. 

(Contd) Step	Action
4	Type the name of the new script, without the *.scr* suffix, into the New Script Name field.
5	Press F10 to save the new script. **System Response** Sterling Gentran:Server adds the script name to the Script Maintenance screen.
6	Select the name of the new translation script and press F5. **System Response** Sterling Gentran:Server displays a Modify screen with additional function keys. 
7	Modify the script's description and then press F10 to save the description.
8	Press F5 to edit the new script. **System Response** Sterling Gentran:Server displays the script editor.
9	Use the standard editor keys to edit the script. **Comment** Here are some items you may want to modify: ◗ Name of the translation data manager that invokes the script (dmName line) ◗ Translation options in the **lftran** command line ◗ Addition of the **xlld** command to run the Life Cycle update program from the translation script. ◗ Messages in the RESULTS section.

(Contd) Step	Action
10	Press F10 to save your changes.
11	Check all the parameter settings in the translation data manager's initialization file. You may need to change the values for these parameters: ◗ XL_MODEL_SCR ◗ XL_RUN_DIR ◗ WORK_DIRECTORY

Maintaining Sterling Gentran:Server Scripts

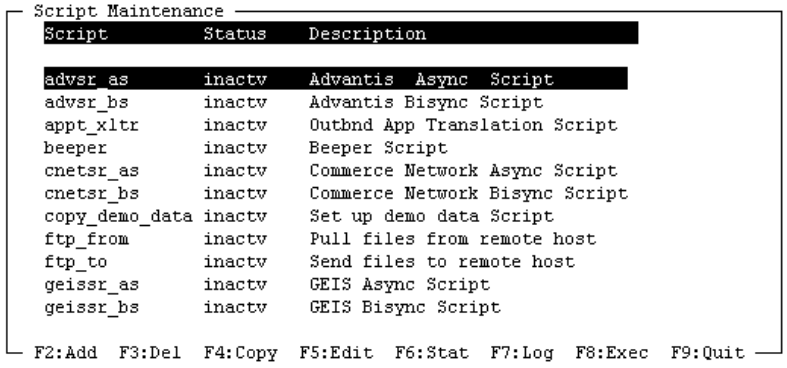
How to Edit a Script with the Script Editor

Introduction

This topic explains how to edit a script with the script editor. The script editor is set in the $EDITOR environment variable. If the variable is not set, Sterling Gentran:Server uses vi as the default editor.

Editing a script

Use this procedure to edit the description or contents of a script.

Step	Action
1	Select **Script** from the host main menu. **System Response** Sterling Gentran:Server displays the Script Maintenance screen. 

<table>
<tr><th>(Contd) Step</th><th>Action</th></tr>
<tr><td>2</td><td>Select the name of the script you want to edit and then press F5.

System Response
Sterling Gentran:Server displays a Modify screen with additional function keys.

</td></tr>
<tr><td>3</td><td>Use this table to determine your next action.

<table><tr><th>IF you want to change…</th><th>THEN…</th></tr><tr><td>The description of the script</td><td>Move the cursor to the character or characters you want to change.
◗ Type the new characters over the existing ones. To erase a character or insert a blank space, press the space bar.
◗ Press F10 to save the changes.</td></tr><tr><td>The contents of the script</td><td>Press F5 to access the script editor and continue with Step 4.

System Response
Sterling Gentran:Server displays the script editor.</td></tr></table></td></tr>
<tr><td>4</td><td>Use the standard editor keys to edit the script.</td></tr>
<tr><td>5</td><td>Press F10 to save your changes.</td></tr>
<tr><td>6</td><td>Does your script include LOCKS?
◗ If YES, see the How to Add a Script with Another Editor topic to add the lock names to the LOCKS directory.
◗ If NO, you are finished.</td></tr>
</table>

How to Delete a Sterling Gentran:Server Script

Introduction

You have three options when you delete a script. You can remove:

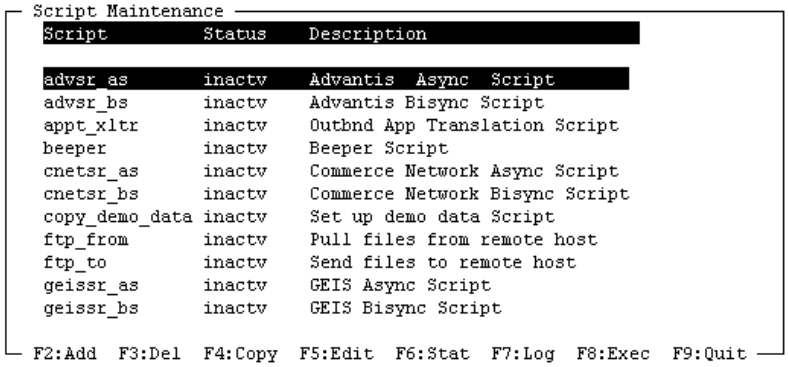
- Only the script name from the library list
- Both the script name and the actual script
- (With a command) only the script itself from the *./script* directory.

Reason to delete only the script name

When you delete only the script name from the library list, the contents of the script remain in the *./script* directory. This allows you to add the script back later by simply adding the script name to the library list.

Deleting the script name only or both the name and the script

Use this procedure to delete just the script name or both the name and the script.

Step	Action
1	Select **Script** from the host main menu. **System Response** Sterling Gentran:Server displays the Script Maintenance screen. 
2	Select the script name you want to delete.

(Contd) Step	Action
3	Press F3 to delete the script. **System Response** Sterling Gentran:Server displays a confirmation prompt for deleting the script name. Delete: appg_xltr Do you want to delete this entry? F9:Quit
4	Type **y** at the prompt to delete the script name. **System Response** Sterling Gentran:Server displays a confirmation prompt for deleting the script. Confirmation Delete the actual Script File (y/n) ?
5	Do you want to delete the script? ◗ If YES, enter **y** for at the prompt ◗ If NO, enter **n** at the prompt

Working with UNIX Mail Scripts

Overview

Description

UNIX mail scripts are Korn shell scripts.

Mail scripts send mail messages

You can use UNIX mail scripts with Sterling Gentran:Server scripts to send messages based on the consequences of the script operations.

A script can have its own UNIX mail script (named for the script it is associated with), or it can use the default UNIX mail script. Your organization can set up the script's UNIX mail script or the default mail script to meet whatever common notification messages you require.

Example
Your organization may want a certain individual or group to be notified if a script fails. You can use UNIX mail scripts to accomplish this.

Mail script location

All UNIX mail scripts reside in the *./mail_proc* directory.

Example UNIX mail script

This is an example of a UNIX mail script.

```
cat $1 | rmail $LOGNAME
~
~
~
~
~
~
~
~
~
~
```

This script sends the file specified in a UNIX mail message to the operator who is logged on.

Reference
See your UNIX documentation for information about writing Korn shell scripts.

How to Add or Edit a UNIX Mail Script

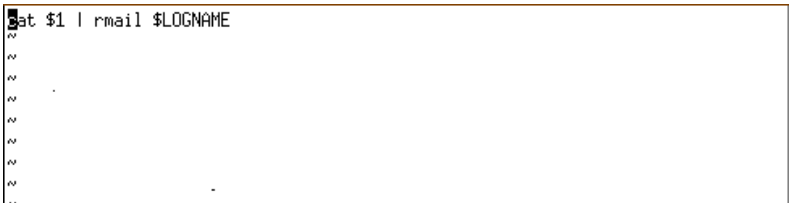
Introduction

You can add a UNIX mail script to a Sterling Gentran:Server script and edit a mail script already created for a Sterling Gentran:Server script.

Adding or editing a UNIX mail script

Use this procedure to add or edit a UNIX mail script for use with a Sterling Gentran:Server script.

Step	Action
1	Select **Script** from the host main menu. **System Response** Sterling Gentran:Server displays the Script Maintenance screen. Script Maintenance Script Status Description advsr_as inactv Advantis Async Script advsr_bs inactv Advantis Bisync Script appt_xltr inactv Outbnd App Translation Script beeper inactv Beeper Script cnetsr_as inactv Commerce Network Async Script cnetsr_bs inactv Commerce Network Bisync Script copy_demo_data inactv Set up demo data Script ftp_from inactv Pull files from remote host ftp_to inactv Send files to remote host geissr_as inactv GEIS Async Script geissr_bs inactv GEIS Bisync Script F2:Add F3:Del F4:Copy F5:Edit F6:Stat F7:Log F8:Exec F9:Quit
2	Select the script associated with the UNIX mail script that you want to add or edit.
3	Press F5 to edit the script. **System Response** Sterling Gentran:Server displays a Modify screen with additional function keys. Modify ap01_xltr Outbnd App Translation Script F4:EditMail F5:EditScr F9:Quit F10:SaveDesc

(Contd) Step	Action
4	Press F4 to add or edit the mail script. **System Response** Sterling Gentran:Server opens the editor. If a mail script exists for the selected script, Sterling Gentran:Server displays it. If a script does not have its own UNIX mail script, the system copies the one named “default” and renames it for the script. 
5	Use the standard editor keys to modify the mail script.
6	Save your changes.
7	Exit the editor.

How to Delete a UNIX Mail Script

Introduction

If you no longer need a UNIX mail script to send messages based on script operations, you can delete the UNIX mail script.

Deleting a UNIX mail script

Use this procedure to delete a UNIX mail script.

Step	Action
1	Go to the UNIX command line.
2	Use a UNIX delete command to remove the mail script. **Example** `rm ./mail_proc/<mail_script_name>`

CHAPTER
6

Running Scripts

Contents

Overview
- Introduction 2

Running Scripts
- Overview 3
- How to Run a Script from the Command Line 5
- How to Run a Script from Another Script 6
- How to Run a Script from the Script Maintenance Screen 8

Running Scripts on a Schedule
- Overview 10
- The Permanent Schedule 11
- The Permanent Schedule Maintenance Screen 12
- The Permanent Schedule Screen 15
- How to Create an Environment File 18
- How to Add a Script to the Permanent Schedule 19
- How to Copy a Schedule 21
- How to Change a Script's Processing Schedule 24
- How to Remove a Script From the Permanent Schedule 26

Restarting Scripts Automatically
- Overview 28
- The Recover Script 29
- How to Make a Script Restart Automatically 30

Overview

Introduction

In this chapter

This chapter explains how to run scripts, monitor the results, and restart scripts automatically after a machine halt.

Key terms

This table lists the key terms used in this chapter.

Term	Description
cleanlog	The command line tool that enables you to purge log file entries.
cron	The UNIX system daemon that starts programs identified in the system's **crontab** at scheduled times.
crontab	A UNIX system file that contains the files listing all the programs to be run by the **cron** daemon. Sterling Gentran:Server submits entries in the permanent schedule to **crontab**.
Permanent Schedule	The Sterling Gentran:Server feature that enables you to run scripts on a specified schedule.
script journal	The file that contains a record of a script's previous activity. When a script finishes, it appends the data in its log file to its journal.
script log	The file in which a script records its process activity while it is active. The script creates a new log file each time it starts.
status	The running state (active or inactive) of a script.

Running Scripts

Overview

Introduction

You can run scripts in a number of ways:

- Have the delivery agent run a post-processing script
- Run the script from another script
- Run the script from the Script Maintenance screen
- Run the script from the command line
- Run the script from the Permanent Schedule.

Selecting a method

Use this decision table to help determine how to start a script.

IF you want to…	THEN…
Run a script after the delivery agent in the flow completes processing	Enter the name of the script in the Post Processing Script Name text box on the Delivery agent dialog box. **Reference** See the Creating a Flow with the PCM Wizard chapter in this guide.
Run or not run a script, depending on the logic defined in another script	Invoke the script from another script. **Reference** See the How to Run a Script from Another Script topic in this chapter.
Run a script on a one-time basis	Start the script from the Script Maintenance screen. **Reference** See the How to Run a Script from the Script Maintenance Screen topic in this chapter.
Run a script without exiting the Sterling Gentran:Server menu system	
Test a new script	Run the script from the command line. **Reference** See the How to Run a Script from the Command Line topic in this chapter.

(Contd) IF you want to…	THEN…
Perform a one-time housekeeping operation	
Run a script at regular intervals **Example** You may want to run a communications script on a schedule.	Start the script from the Permanent Schedule. **Reference** See the Running Scripts on a Schedule section in this chapter.
Run a script to perform housekeeping operations on a schedule **Example** Run the longterm script to handle long-term archiving	

In this section

This section contains these topics:

- How to Run a Script from the Command Line
- How to Run a Script from Another Script
- How to Run a Script from the Script Maintenance Menu.

How to Run a Script from the Command Line

Introduction

You can invoke a Sterling Gentran:Server script from the command line just like any other UNIX command.

When to use

Run a script from the command line when you want to:

- Test the script
- Run a script before its next scheduled time
- Run a script only occasionally.

Procedure

Use this procedure to run a script from the command line.

Step	Action
1	Go to the UNIX command line.

2	Type this command:
	`smgr -s<scriptname>`
	Where <*scriptname*> is the name of the script.
	Example
	To run the script named *mvedi.scr*, type the following at the command line:
	`smgr -smvedi`
	Note
	Do not include the *.scr* extension in the script name.

How to Run a Script from Another Script

Introduction

You may have a Sterling Gentran:Server script invoke a UNIX shell script or a Sterling Gentran:Server script.

When to use

Use this procedure when you want to:

- Invoke UNIX shell script or Sterling Gentran:Server scripts from a Sterling Gentran:Server script
- Run or not run a UNIX shell script or a Sterling Gentran:Server script based on logic statements in a Sterling Gentran:Server script
- Use a Sterling Gentran:Server script to indirectly run a UNIX shell script from the Permanent Scheduler or as a post process to a data manager.

Procedure

Use this procedure when you want to run a script from a Sterling Gentran:Server script.

Step	Action
1	Does the script you want to invoke exist? ◗ If YES, continue with Step 2. ◗ If NO, create the script that you want to invoke. **Reference** See the Working with Scripts chapter in this guide for instructions.
2	Does the script that will start the script exist? ◗ If YES, continue with Step 3. ◗ If NO, create the script and then continue with Step 3.

(Contd) Step	Action

3	Modify the script from Step 2 to include a step in the STEPS group that starts the script that you want to invoke.
	
	Example 1
	This example shows a step to invoke the Cleo UNIX shell script.
	
	`:run_cleo`
	`## Execute the Cleo shell script`
	`sh $cleoDir/$VAN.run`
	
	Example 2
	This example shows a step to invoke a Sterling Gentran:Server script named check_ftp.
	
	`:run_check`
	`## Start the Server check_ftp script`
	`smgr -scheck_ftp`

4	Add a PROC to the PROCS group to execute the new step.
	
	Example
	
	`:run_cleo`
	`if run_cleo then check_if_tried_send else`
	`check_if_tried_send`

5	Save your changes and exit the editor.

How to Run a Script from the Script Maintenance Screen

Introduction

The Sterling Gentran:Server Script Maintenance screen enables you to start a Sterling Gentran:Server script that is in the script library.

Reference
See the Working with Scripts chapter in this guide for information about adding a script to the script library.

When to use

Use this procedure when you want to:

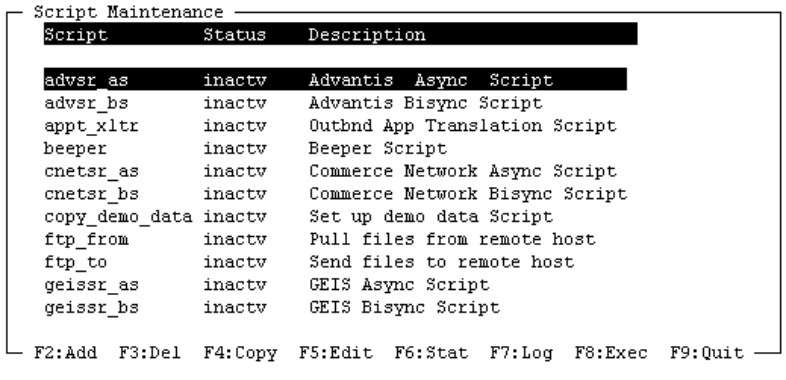
- Test a script
- Run a script outside its scheduled time.

Before you begin

Before you attempt to run a script from the Script Maintenance menu, make sure you have *at.allow* and *cron.allow* privileges. Scripts are queued or scheduled through the UNIX batch facility, which requires *at.allow* and *cron.allow* privileges.

Procedure

Use this procedure to run a script from the Script Maintenance menu.

Step	Action
1	Select **Script** from the host main menu. **System Response** Sterling Gentran:Server displays the Script Maintenance screen. 

(Contd) Step	Action
2	Check the Status field of the script you want to run to make sure that the script is inactive.
3	Select the script name you want to run.
4	Press F8 to run the script. **System Response** Sterling Gentran:Server runs the script. The script's status is active while the script is running.

Running Scripts on a Schedule

Overview

Introduction

Sterling Gentran:Server enables you to run Sterling Gentran:Server scripts on a schedule.

In this section

This section contains these topics:

- The Permanent Schedule
- The Permanent Schedule Maintenance Screen
- The Permanent Schedule Screen
- How to Create an Environment File
- How to Add a Script to the Permanent Schedule
- How to Copy a Schedule
- How to Change a Script's Processing Schedule
- How to Remove a Script From the Permanent Schedule

The Permanent Schedule

Introduction

The Permanent Schedule feature enables you to specify the names of Sterling Gentran:Server scripts (but not UNIX shell scripts) you want to run on a schedule as well as the schedule itself.

Reference
To run a UNIX shell script indirectly from a Sterling Gentran:Server script that you add to the Permanent Schedule, see the How to Run a Script from Another Script topic in this chapter.

How the Permanent Schedule works

Sterling Gentran:Server submits the Permanent Schedule entries to **crontab**, which is the list of programs that the UNIX **cron** daemon runs at specified times.

Reference
For more information about **cron** and **crontab**, see your UNIX reference books.

Process environments

Processes scheduled from within Sterling Gentran:Server overwrite the user's existing **crontab** files. You may want to use the ediadmin user ID to schedule scripts in the Permanent Schedule.

Permanent Schedule entries are run from **cron**, which uses different environment variables than the ones set for the Sterling Gentran:Server user. For this reason, you may need to create an environment file to enable **cron** to find executable files and spawned executable files.

Reference
See the How to Create an Environment File for instructions.

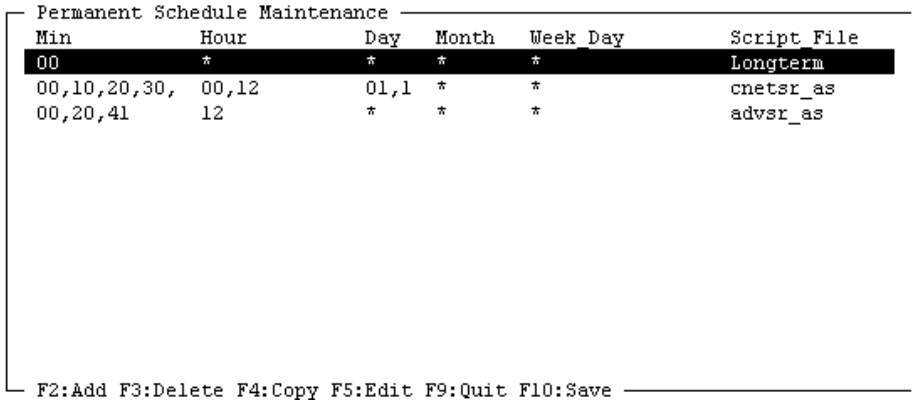
The Permanent Schedule Maintenance Screen

Introduction

You start Permanent Schedule activities from the Permanent Schedule Maintenance screen.

Sample Permanent Schedule Maintenance screen

This illustration shows a sample Permanent Schedule Maintenance screen.



Permanent Schedule Maintenance screen fields and functions

This table lists the fields of the Permanent Schedule Maintenance screen and their functions.

Field	Function
Min	Displays the time within the hour that the script is to run. **Examples** The number 00 in this field means that the script runs on the hour. The number 30 in this field means that the script runs at 30 minutes past the hour.

(Contd) Field	Function
Hour	Displays the hours that the script runs, based on a 24-hour clock. **Examples** The number 06 means that the script runs at 6 a.m. The number 18 means that the script runs at 6 p.m. An asterisk (*) means that the script runs every hour.
Day	Displays the days of the month that the script is run, based on a 31-day calendar. **Examples** The number 01 means that the script runs on the first day of the month. An asterisk (*) means that the script runs every day of the month.
Month	Displays the months the script is to run, January through December. **Example** The number 04 means that the script runs in April. An asterisk (*) means that the script runs every month.
Week Day	Displays the days of the week the script is to run, Sunday through Saturday. **Examples** The number 01 means that the script runs on the first day of the week (Sunday). An asterisk (*) means that the script runs every day of the week.
Script_File	Displays the name of the script run under the schedule.

Permanent Schedule Maintenance screen function keys

This table describes the function keys on the Permanent Schedule Maintenance screen and their functions.

Key	Function
F2	Displays an add screen for adding a script to the Permanent Schedule.
F3	Deletes the selected script from the Permanent Schedule.
F4	Copies the schedule of the selected script.
F5	Displays the edit screen for changing the schedule of the selected script.
F9	Exits the screen.
F10	Saves your changes to the Permanent Schedule.

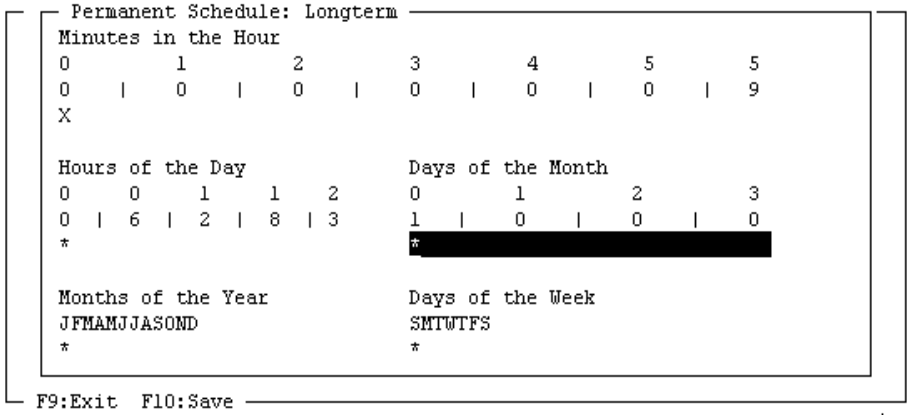
The Permanent Schedule Screen

Introduction

To add or edit the run schedule for a Sterling Gentran:Server script, you complete the Permanent Schedule screen.

Permanent Schedule screen

This illustration shows the Permanent Schedule screen.



Permanent Schedule screen fields and functions

This table lists the fields of the Permanent Schedule Maintenance screen and their functions.

Field	Function
Permanent Schedule	Displays the name of the script to be run under the schedule.
Minutes in the Hour	Defines the time within the hour that the script is to run. **Examples** To run the script on the half hour, type **X** at 30. To run the script at 45 minutes past the hour, type **X** at the halfway point between 40 and 50.

(Contd) Field	Function
Hours of the Day	Defines the hours that the script runs, based on a 24-hour clock. **Examples** Type an **X** at 06 to run the script at 6 a.m. Type an **X** at 18 to run the script at 6 p.m. To run the script every hour, type an asterisk (*) in the first position.
Days of the Month	Defines the days of the month that the script is run, based on a 31-day calendar. **Examples** Type an **X** at 01 to run the script on the first day of the month. Type an **X** at 10 to run the script on the 10th of the month. To run the script on every day of the month, type an asterisk (*) in the first position.
Months of the Year	Defines the months the script is to run, January through December. **Examples** Type an **X** under F and S to run the script in February and September. To run the script every month, type an asterisk (*) in the first position.
Days of the Week	Displays the days of the week the script is to run, Sunday through Saturday. **Examples** Type an **X** under M, T, W, T, and F to run the script Monday through Friday. To run the script every day of the week, type an asterisk (*) in the first position.

Using the Permanent Schedule screen

This table describes how to enter the run schedule on the Permanent Schedule screen.

IF you want to…	THEN…
Select the minutes, hours, days, or months	Type **X** in each field that applies.
Select all values for the **Hours of the Day**, **Days of the Month**, **Months of the Year**, or **Days of the Week** field	Type an asterisk (*****) in the first column of the field. The system clears the field. You cannot use this option in the **Minutes in the Hour** field.
Remove an X or asterisk	Select the character and press the space bar.

How to Create an Environment File

Introduction

An **environment file** enables you to set the environment variables for the Sterling Gentran:Server scripts in the Permanent Schedule. If you do not create an environment file, **cron** runs the scripts under the environment set for **cron**. The **cron** program does not use the environment of the user who submitted the schedule.

When to use

Use this procedure when you want to specify the environment for the scripts run from the Permanent Schedule rather than run the scripts under the environment that **cron** uses.

Procedure

Use this procedure to create an environment file.

Step	Action
1	At the UNIX command line, create a hidden file named *$EDI_ROOT/.edi_env*.
2	Type into the *.edi_env* file the following: ◗ The environment variables in which you want the process to run ◗ The location of EDI_ROOT and its path.

Sample edi_env file

This is a sample *.edi_env* file.

```
export EDI_ROOT=/USR/EDI2/SRVR22/QA
export VVTERMCAP=/usr/edi2/srvr22/qa/bin/vvtermcap
export EDI_AUDIT=$EDI_ROOT/bin/audit.sh
export TERM=vt100
export PATH=/usr/bin:/etc:/usr/sbin:$EDI_ROOT/bin
```

How to Add a Script to the Permanent Schedule

Introduction

If you want to run a Sterling Gentran:Server script on a specified schedule, add the script to the Permanent Schedule.

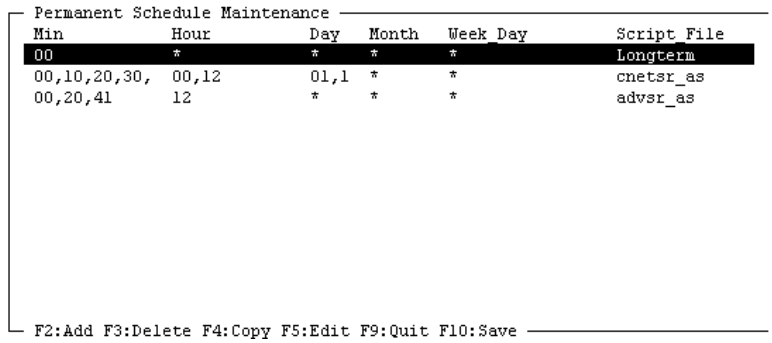
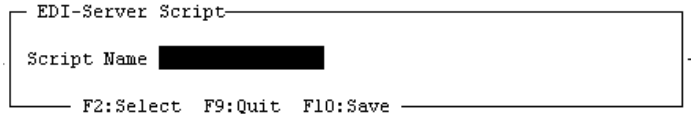
Before you begin

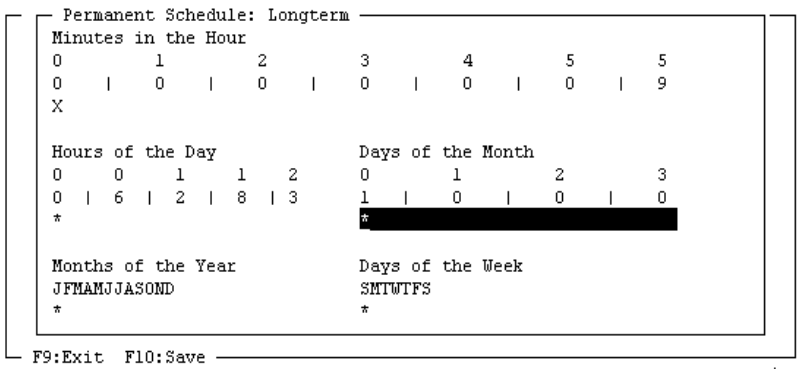
Before you can add a script to the Permanent Schedule, you must create the script and add its name to the script library.

Reference
See the Working with Scripts chapter in this guide for instructions.

Procedure

Use this procedure to add a script to the Permanent Schedule.

Step	Action
1	Select **Sched** from the host main menu. **System Response** Sterling Gentran:Server displays the Permanent Schedule Maintenance screen. 
2	Press F2 to add the script to the schedule. **System Response** Sterling Gentran:Server displays a screen to add the script name. 

(Contd) Step	Action
3	Press F2 to display a list of script names that were added to the script library. Select the script name and then press ENTER. **System Response** Sterling Gentran:Server displays a Permanent Schedule screen, which is the screen you use to set up the schedule for the script. 
4	Type **X**s and asterisks in the fields to select the schedule. **Reference** See The Permanent Schedule for information.
5	Press F10 to save the changes to the Permanent Schedule.
6	Press F10 again to submit the schedule to **crontab**.

How to Copy a Schedule

Introduction

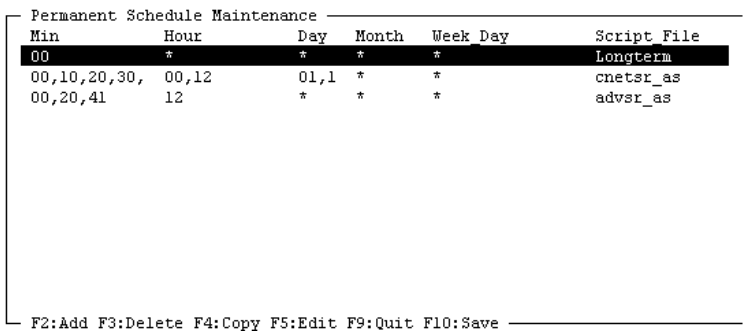
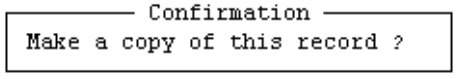
There are two ways to copy a schedule:

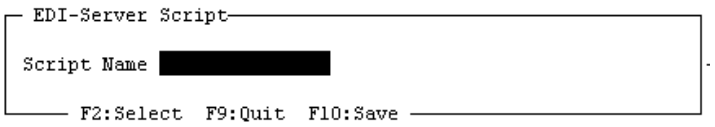
- Copy an entry and rename it to apply the schedule to a new script
- Copy an entry, use the same script name, and then modify the schedule to run the same script on a different schedule.

This topic includes procedures for both options.

Applying a schedule to a different script

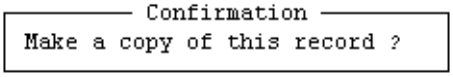
Use this procedure to copy a schedule.

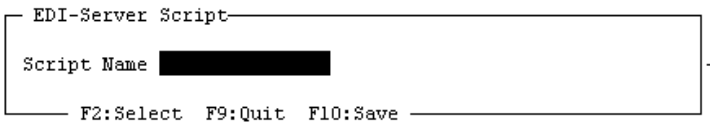
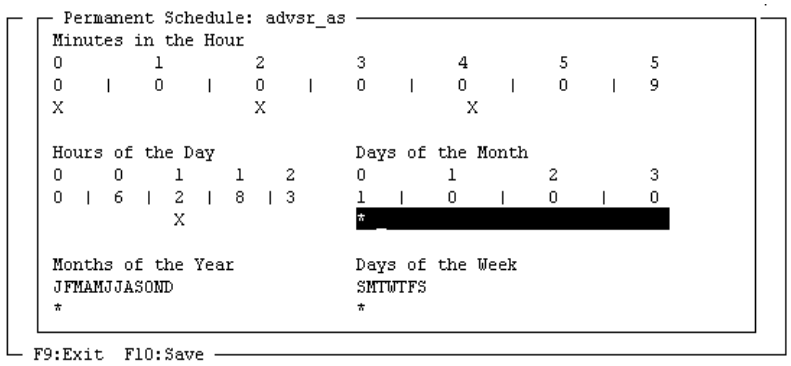
Step	Action
1	Select **Sched** from the host main menu. **System Response** Sterling Gentran:Server displays the Permanent Schedule Maintenance screen. 
2	Select the line of the schedule you want to copy and then press F4. **System Response** Sterling Gentran:Server copies the line and displays a Confirmation prompt. 

(Contd) Step	Action
3	Type **y** to confirm the copy. **System Response** Sterling Gentran:Server displays a screen for the new script name. 
4	Press F2 to display a list of available script names.
5	Select the script name and press ENTER.
6	Press F10 to submit the schedule to **crontab**.

Running a script on another schedule

Use this procedure to copy an entry and set up another schedule for the same script.

Step	Action
1	Select **Sched** from the host main menu. **System Response** Sterling Gentran:Server displays the Permanent Schedule Maintenance screen.
2	Select the line of the schedule you want to copy and then press F4. **System Response** Sterling Gentran:Server copies the line and displays a Confirmation prompt. 

(Contd) Step	Action
3	Type **y** to confirm the copy. **System Response** Sterling Gentran:Server displays a screen for the new script name. 
4	Type the name of the script that you copied.
5	Press ENTER. **System Response** Sterling Gentran:Server displays the Permanent Schedule screen for the script. 
6	Change the schedule for the duplicate entry. **Reference** To change the processing schedule, see the How to Change a Script's Processing Schedule topic in this chapter.

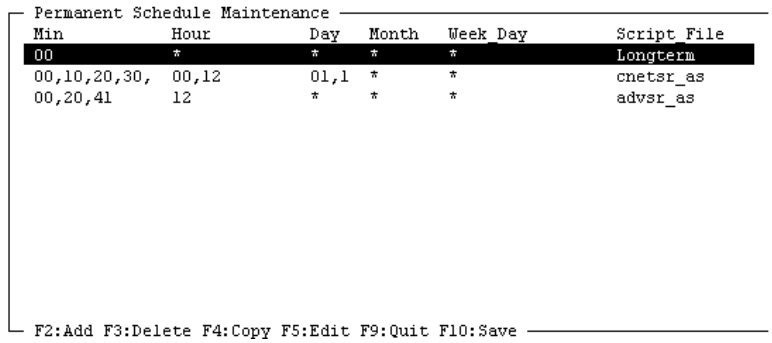
How to Change a Script's Processing Schedule

Introduction

You may change any part of a Sterling Gentran:Server script processing schedule.

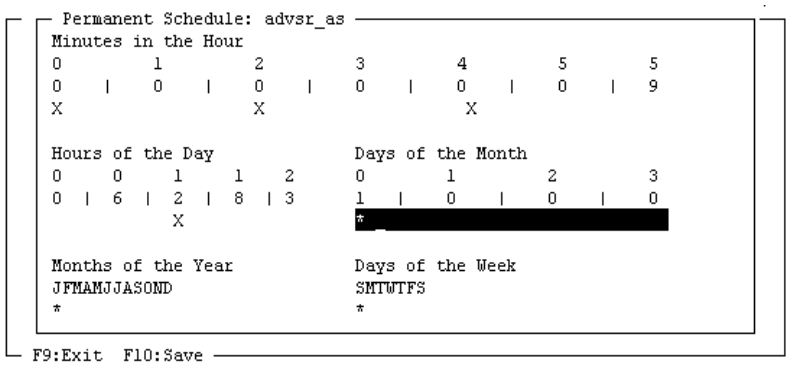
Procedure

Use this procedure to change a script processing schedule.

Step	Action
1	Select **Sched** from the host main menu. **System Response** Sterling Gentran:Server displays the Permanent Schedule Maintenance screen. 
2	Select the script line that you want to change.

<table>
<tr><th>(Contd) Step</th><th>Action</th></tr>
<tr><td>3</td><td>Press F5 to edit the processing schedule.

System Response
Sterling Gentran:Server displays the Permanent Schedule screen for the script.

 
</td></tr>
<tr><td>4</td><td>Use this table to change the schedule.
<table>
<tr><th>IF you want to…</th><th>THEN…</th></tr>
<tr><td>Select a value</td><td>Type an X.</td></tr>
<tr><td>Select all values in any field (except the Minutes field)</td><td>Type an asterisk (*) in the first position.</td></tr>
<tr><td>Remove a character</td><td>Press the space bar.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Press F10 to save your changes.

System Response
Sterling Gentran:Server displays the Permanent Schedule menu and a message that asks if you would like to change the Sterling Gentran:Server script.</td></tr>
<tr><td>6</td><td>Type n at the prompt and continue with Step 7.</td></tr>
<tr><td>7</td><td>Press F10 to submit the modified schedule to crontab.</td></tr>
</table>

How to Remove a Script From the Permanent Schedule

Introduction

You can remove a Sterling Gentran:Server script from the Permanent Schedule at any time, even if the script is running. Removing the script does not affect current processing.

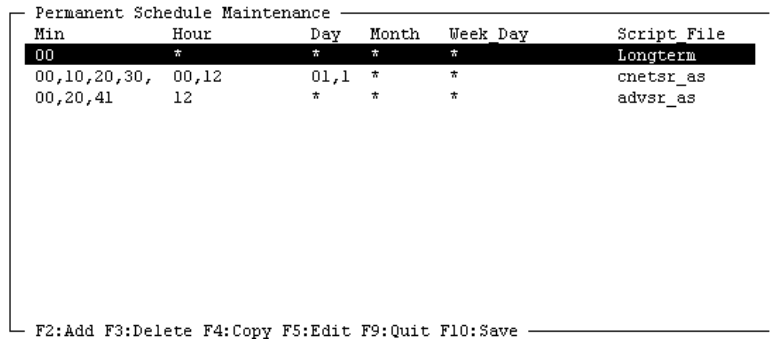
When to use

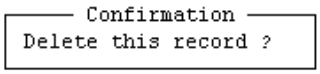
Use this procedure when you:

- Want to delete from your system a script that is run on a schedule
- No longer want to run the script on a schedule.

Procedure

Use this procedure to delete a script from the Permanent Schedule.

Step	Action
1	Select **Sched** from the host main menu. **System Response** Sterling Gentran:Server displays the Permanent Schedule Maintenance screen. 
2	Select the script.

(Contd) Step	Action
3	Press F3 to delete the script from the schedule. **System Response** Sterling Gentran:Server displays a confirmation prompt. 
4	Type **y** at the prompt to confirm the deletion. **System Response** Sterling Gentran:Server displays the Permanent Schedule menu.
5	Press F10 to save the new schedule and submit it to **crontab**. **Comment** If you delete all the scripts, you must still press F10.
6	Press F9 to exit the screen.

Restarting Scripts Automatically

Overview

Introduction

Unexpected machine halts disrupt script processing. To restart Sterling Gentran:Server scripts after a machine halt, Sterling Gentran:Server provides a special script, *recover.scr.*

In this section

This section contains these topics:

- The Recover Script
- How to Make a Script Restart Automatically.

The Recover Script

Introduction

The **recover script**, *recover.scr,* is used to restart Sterling Gentran:Server scripts after a machine halt.

The recover script restarts scripts

In the recover script, you specify the scripts that you want the recover script to restart. The recover script restarts these scripts at machine start-up if the scripts were active when the machine stopped.

CAUTION

The scripts restart from the beginning, not at the point they stopped.

The recover.scr script generates mail notices

You can also use the recover script to generate mail notices about the scripts that the machine halt affected.

You can modify the recover script to have it generate mail messages that:

- Let you know which scripts were running when the machine halted
- List the scripts that still have logs and appear active, even though they aren't active currently.

Where to find the recover script

The recover script is located in the *$EDI_ROOT/script* directory.

How to Make a Script Restart Automatically

Introduction

To automatically restart a Sterling Gentran:Server script that was active when the machine stopped, you must:

- Modify the recover script to include the script name
- Include the recover script in the */etc/inittab* or */etc/rc* initialization file so that it starts up when the operating system is restarted.

This topic covers procedures for both tasks and explains how to include mail notification steps.

Modifying the recover script

Use this procedure to modify the recover script.

Step	Action
1	Select **Script** from the host main menu. **System Response** Sterling Gentran:Server displays the Script Maintenance screen.
2	Select the recover script.
3	Press F5 to display the Modify screen.
4	Press F5 to access the script editor.

5	Add to the script a STEP that defines a list of all scripts to be run. Include the script manager command line option to execute the scripts. You may need to add a PROC to execute the STEP.
	
	Example
	
	`:Run_script`
	`STEP Section Commands:`
	`smgr -s ‘head -1 scriptlist‘`
	`sed -e ‘1d’ <scriptlist >scriptlist.tmp`
	`mv scriptlist.tmp scriptlist`
	`:Check_list`
	`test -s scriptlist`

(Contd) Step	Action

6	Do you want the recover script to generate a notice to let you know which scripts were running when the machine halted?
	◗ If YES, add a STEP that checks the script directory for files with a *.l extension* and then checks to see if the script manager is running the script that created the log.
	
	Example
	`:Find_Scripts`
	`ls script/*.l | cut -c7- > filename`
	`cut -d -f1 <filename>scriptname`
	`ps -ef | grep 'head -1 scriptname"`
	
	◗ If NO, continue with Step 7.

7	Do you want the recover script to tell you which scripts still have logs and appear active, even though they aren't active currently?
	◗ If YES, add a STEP that calls the mail script, includes the notification message, and creates a list of scripts listed as active, but not running.
	
	Example
	`:Mail_list`
	`echo "Recover scripts found these scripts not running" >mailfile`
	`cat scriptname >>mailfile`
	`sh mail_proc/default mailfile`
	`If NO, continue with Step 8.`

8	Press F10 to save the script.

Adding the recover script to the */etc/inittab* or */etc/rc* initialization file

Use this procedure to add the recover script to the initialization file.

Note

Only a system administrator or a user with root privileges can perform this procedure.

Step	Action
1	Exit Sterling Gentran:Server and access the UNIX command line. **WARNING** **From Sterling Gentran:Server, press CTRL+A to access the UNIX command line.**
2	Open the */etc/inittab* or */etc/rc* file. **Reference** See your UNIX documentation for instructions.
3	Add the recover script to the file. **Example** `smgr -srecover`

CHAPTER
7

Defining the Document Reference Number

Contents

Overview
- Introduction 3
- The Document Reference Number 5
- How the Document Reference Number is Set 6
- Document Specifier Tables 9

Defining Document Reference Numbers
- Overview 11
- Document Reference Number Specifier Screen 12

Creating a Document Specifier Table
- Overview 14
- How to Add a Document Specifier Table 15
- How to Display a Document Specifier Table 18
- How to Copy a Document Specifier Table 21

Mapping Document Specifier Tables
- Overview 23
- EDI Add Screen 27
- APP Add Screen 32
- XML Add Screen 36
- NCPDP Add Screen 39
- How to Add a Map to a Document Specifier Table 42
- Using the Qualifier and Occurrence Fields 45
- How to Display a Map Picture 47

Attaching a Trading Partnership Code to a Table
- How to Attach a Trading Partnership Code to a Table 50
- How to Verify Trading Partnership Code Attachments 54

Maintaining Document Reference Number Tables

- Overview 56
- How to Change a Trading Partnership Code and Table Attachment . 57
- How to Remove a Trading Partnership Code from a Table 58
- How to Delete a Document Specifier Table 60

Maintaining Document Specifier Maps

- Overview 61
- EDI Mapping Screen 62
- APP Mapping Screen 65
- XML Mapping Screen 68
- NCPDP Mapping Screen 71
- How to Display a Document Specifier Map 73
- How to Change a Document Specifier Map 75
- How to Delete a Map from a Table 78

Overview

Introduction

In this chapter

This chapter describes how to specify the characters in a document that the inbound data manager, the application data manager, the XML data manager (if you have the XML translation option), the NCPDP data manager (If you use the NCPDP EDI standard), and the translator use to determine the document reference number.

Key terms

This table lists the key terms used in this chapter.

Term	Description
application name map	A record that identifies the records and fields that the data manager extracts from the application transaction to build the document reference number.
date-time stamp	The label that Sterling Gentran:Server attaches to a document to identify the date and time the document was received.
document reference number	The unique number that Sterling Gentran:Server assigns to each data set or document to track the movement of the data set or document through the system.
document specifier table	A collection of set ID or application name maps that specifies the places in a trading partner's document that certain Sterling Gentran:Server processes reference to construct the document reference number.
document specifier utility	The Sterling Gentran:Server tool that enables you to specify the characters in a document that the inbound data manager, application (appm) data manager, NCPDP data manager, and the translator extract to derive the document reference number.
mailbag ID	The 6-character, base-32 code that Sterling Gentran:Server generates to identify a session in which files were received and passed.

(Contd) Term	Description
map picture	A pictorial representation of the map for the document reference number. Each line in the map is represented with a unique symbol.
set ID map	A record that identifies the segments, elements, and sub-elements that the data manager or translator extracts from the transaction set or EDI document to build the document reference number.
transaction register	An indexed file used to keep track of documents that Sterling Gentran:Server handles.

The Document Reference Number

Introduction

To track the movement of a document through the system, Sterling Gentran:Server assigns a unique **document reference number** to each data set or document. This number distinguishes the document from all others throughout the document's life in Sterling Gentran:Server.

The document reference number is used in the Life Cycle event record to identify the document.

The document reference number is used in:

- Transaction registers
- Archive logs
- The Life Cycle database (if your organization uses a database).

Role in duplicate checking

Sterling Gentran:Server also uses the document reference number to detect duplicate documents, if you structured your data manager initialization files to check for duplicates.

To check for duplicates, a data manager examines the transaction register and compares the derived document reference number to those in the transaction register to see if it already exists.

This table describes what happens if you structure your system to not allow duplicates.

IF the document reference number…	**THEN…**
Is already in the transaction register	Sterling Gentran:Server processes the data as a duplicate type error.
Is not in the transaction register	The data manager creates an entry and adds it to the register.

How the Document Reference Number is Set

Introduction

The Sterling Gentran:Server data managers and the translator either:

- Extract the document reference number from the document itself, or
- Create the document reference number from the mailbag ID and date-time stamp.

The method depends on the type of data manager and the type of document the data manager or translator is designed to handle.

Method table

This table describes how data manager or process either extracts or produces the document reference number.

Data Manager or Process	Method	
translation data manager (xltr personality)	**IF the ADD_MBAG_TO_DOC_REF_NUM parameter in the data manager's initialization file is set to…**	**THEN the data manager…**
	0	Creates the document reference number from the date-time stamp (DATETIME) that indicates when the document was processed.
	1	Uses the mailbag ID and the date-time stamp (MBAGID.DATETIME) for the document reference number.

(Contd) Data Manager or Process	Method	
inbound data manager (inbd personality)	**IF the Trading Partnership code is…**	**THEN the data manager…**
	Attached to an EDI document specifier table	Derives the document reference number from the instructions in the EDI document specifier table.
	Attached to an EDI document specifier table, but no entry for the set ID exists in the table	Uses the default EDI document specifier table.
	Not attached to a document specifier table	Uses the default EDI document specifier table.
application data manager (appm personality)	**IF the Trading Partnership code is…**	**THEN the data manager…**
	Attached to a document specifier table	Derives the document reference number from the instructions in an APP document specifier table.
	Attached to a document specifier table, but no entry for the application exists in the table	Does not create a document reference number. ◗ If you use the Life Cycle facility, the Life Cycle update process fails.
	Not attached to a document specifier table	Does not create a document reference number. This means that: ◗ If you use the Life Cycle facility, the Life Cycle update process fails.

(Contd) Data Manager or Process	Method	
translator (lftran)	**IF the document type is…**	**THEN the translator …**
	Inbound EDI	Derives the document reference number used in the temporary audit file from the instructions in a document specifier table.
	Outbound EDI	Derives the document reference number from the instructions in a document specifier table.

Repeating segments or records

This table describes how Sterling Gentran:Server handles repeating segments or records.

IF a segment or record is …	THEN the data manager or translator…
Repeated in the document	◗ Checks the map to determine which occurrence to use. ◗ If the map does not specify the occurrence, uses the last occurrence of the segment or record. ◗ Extracts the document reference number.
Not repeated in the document	Extracts the document reference number.

Document Specifier Tables

Introduction

The inbound data manager, application data manager, XML data manager, NCPDP data manager, and the translator follow the instructions in document specifier tables to determine the values in the document reference number.

Definition

A document specifier table is a set of instructions that specifies the characters in a document that Sterling Gentran:Server processes extract for the document reference number. You create document specifier tables for:

- Inbound data managers
- Application data managers
- XML data managers (if you have the Sterling Gentran:Server XML translation option).
- NCPDP data managers (if you use the NCPDP EDI standard)

Each document specifier table has one or more set ID, application name, or file definition (DDF) name maps linked to it. Think of a table and its associated maps as a set of rules for extracting the document reference number.

Example

You can build a document reference number table to extract the Purchase Order (PO) number and the release number from a document. Together, these values uniquely identify the document.

Types of document specifier tables

There are four types of document specifier tables:

- EDI - A group of set ID maps that identify the segments, elements, and sub-elements you want used for the document reference number of EDI transaction sets.
- APP - A set of application name (<filename>.app) or file definition name (<filename>.ddf) maps that specify the record IDs and fields you want used for the document reference number of application transaction sets.
- XML - A set of file definition name maps that specify the paths to the target nodes you want used for the document reference number of XML transaction sets.
- NCPDP - A set of file definition (DDF) name maps that specify the transaction code segments and fields you want used for the document reference number of NCPDP transactions sets.

Default table for EDI documents

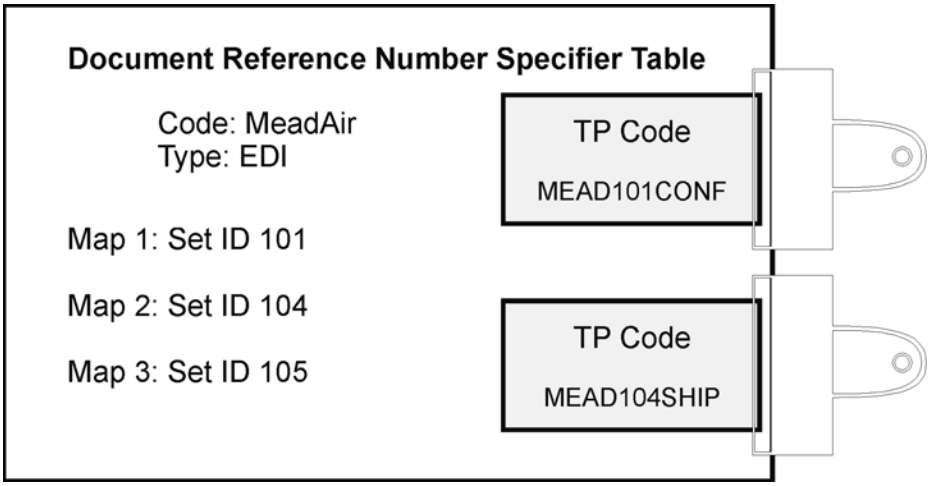
Sterling Gentran:Server has a default document specifier table for EDI documents. This table defines a basic way to extract the document reference number. You can use the default table or create your own to override the default settings.

Trading Partnership codes linked to document specifier tables

You attach one or more Trading Partnership codes to a document specifier table. This enables the data managers and translator to extract the document reference number according to the rules you establish for a specific trading partner's documents.

Diagram

The following illustration shows the relationship between the document specifier table, the set IDs, the maps, and the Trading Partnership codes in an EDI table. This table has three maps. Two Trading Partnership codes are attached to it.



Defining Document Reference Numbers

Overview

Introduction

You perform three tasks to define the values that your inbound data manager, application data manager, XML data manager, NCPDP data manager, or translator uses to make up the document reference number.

Task overview

This table describes the tasks you must perform to create a document specifier table and its maps.

Task	Description
1	Create an EDI, APP, XML, or NCPDP document specifier table. **Reference** See Creating a Document Specifier Table.
2	Construct one or more set ID, application name, XML name, or NCPDP transaction code maps for the document specifier table. **Reference** See Mapping Document Specifier Tables.
3	Attach one or more Trading Partnership codes to the table. **Reference** See Attaching a Trading Partnership Code to a Table.

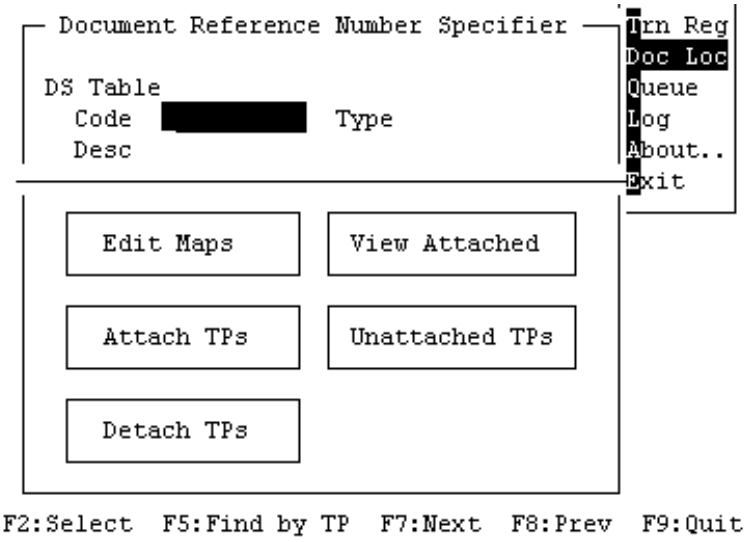
Document Reference Number Specifier Screen

Introduction

All the tasks you need to complete to define document reference numbers begin on the Document Reference Number Specifier screen.

Document Reference Number Specifier screen

This illustration shows the Document Reference Number Specifier screen.



Document Reference Number Specifier screen fields and functions

This table describes the fields of the Document Reference Number Specifier screen and their functions.

Field	Function
DS Table Code	Defines the name of the document specifier table.
DS Table Type	Defines the type of document specifier table: ◗ EDI ◗ APP ◗ XML (if you have the XML translation option) ◗ NCP (if you use the NCPDP standard)
DS Table Description	Describes the document specifier table.

(Contd) Field	Function
Edit Maps	Displays the mapping screen for the specified table type. Use the mapping screen to create or edit an EDI, APP, XML, or NCPDP document specifier map table.
View Attached	Displays a list of Trading Partnership codes attached to the document specifier table.
Attach TPs	Displays the Trading Partnership Search screen. Use this screen to search for the Trading Partnerships that you want to attach to the document specifier table.
Unattached TPs	Lists all the Trading Partnership codes that are not linked to a document specifier table.

Document Reference Number Specifier screen function keys

This table describes the function keys on the Document Reference Number Specifier screen and their functions.

Key	Function
F2	When the DS Table Code field is selected, displays a list of the existing document specifier tables.
F5	When the DS Table Code field is selected, displays the Trading Partnership search screen, which enables you to search for a table attached to a Trading Partnership code.
F7	Displays information about the next document specifier table in your system.
F8	Displays information about the previous document specifier table in your system.
F9	Exits the screen.

Creating a Document Specifier Table

Overview

In this section

This section describes how to create document specifier tables.

Two ways to create a new document specifier table

There are two ways to create a new table:

- Add a new table on the Document Number Specifier screen.

  When you use this method, you must map the table entirely and then attach one or more Trading Partnership codes to the table.

- Display a table similar to the one you want to create and then copy it to a new table.

  This method copies the table's set ID or application name maps, but not the Trading Partnership code attachments. If you use this method, you can modify the new table with the Edit Maps function and then attach Trading Partnership codes to it.

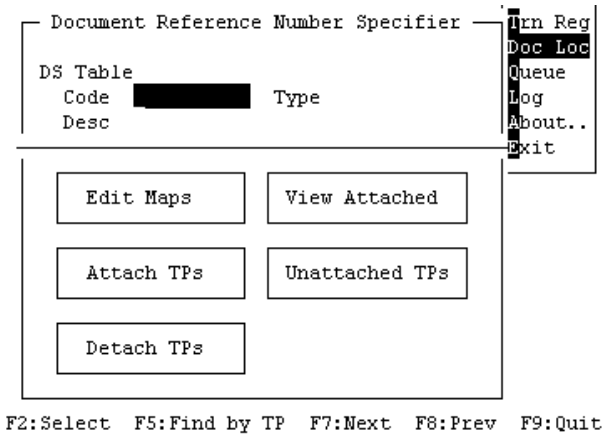
How to Add a Document Specifier Table

Introduction

Creating a document specifier table is the first task in defining the values in a document that will comprise the document reference number.

Adding a document specifier table

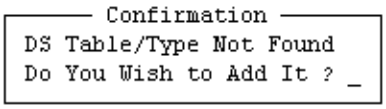
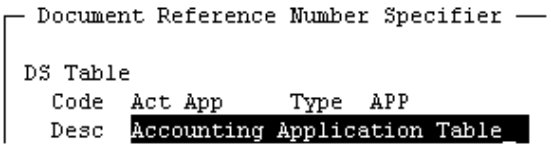
Use this procedure to add a table.

Step	Action
1	Select **Doc Loc** from the host Util menu. **System Response** Sterling Gentran:Server displays the Document Reference Number Specifier screen. 

<table>
<tr><th>(Contd) Step</th><th colspan="3">Action</th></tr>
<tr><td rowspan="5">2</td><td colspan="3">Type the name of the new table in the DS Table Code field and press ENTER.

System Response
The utility searches a list of table names for a match.</td></tr>
<tr><td>IF the DS Table Code…</td><td>THEN the system displays…</td><td>AND you should…</td></tr>
<tr><td>Exists</td><td>The table description and type on the Document Reference Number Specifier screen</td><td>Enter a different name for your table.</td></tr>
<tr><td>Does not exist</td><td>A “Confirmation Partial Key Lookup” prompt to have Sterling Gentran:Server use the characters you entered as a table search key</td><td>Continue with Step 3.</td></tr>
<tr><td rowspan="4">3</td><td colspan="3">Type y for yes and press ENTER.

System Response
The system uses the characters as a search key for a table name.</td></tr>
<tr><td>IF…</td><td>THEN the system…</td><td>AND you should…</td></tr>
<tr><td>No table names match the search characters</td><td>Displays the message “No records matched” in the message field at the bottom of the screen. It then moves the cursor to the Type field</td><td>Continue with Step 4.</td></tr>
<tr><td>One or more table names match the search characters</td><td>Displays the first table type and description that matches</td><td>Quit, start again, and give the table a code name that does not exist.</td></tr>
</table>

(Contd) Step	Action
4	Type **EDI, APP, XML,** or **NCP** in the Type field to specify the table type and press ENTER. **System Response** The system displays a Confirmation prompt. 
5	Type **y** for yes and press ENTER. **System Response** The system moves the cursor to the Desc field.
6	Type the description of the new table in the Desc field. 
7	Press F10 to save the new table.
8	GO TO Mapping Document Specifier Tables.

How to Display a Document Specifier Table

Introduction

To create a new document specifier table, you can display an existing table and then copy it. This topic explains how to display a table.

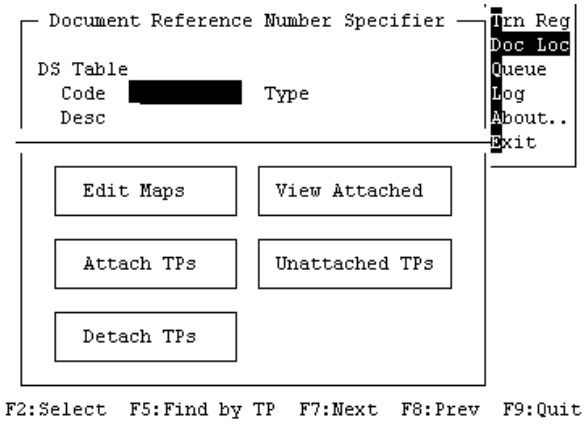
Two ways to search for a table

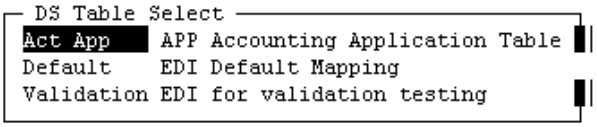
There are two ways to search for and display a document specifier table from the Document Reference Number Specifier screen:

- Press F2, the select key, to display a list of all the existing tables.
- Press F5 to specify a Trading Partnership code that is attached to the table you want.

Selecting a table with the select key

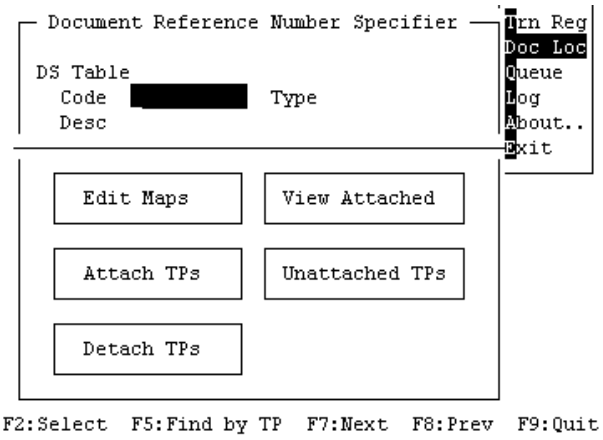
Use this procedure to select a table with F2, the select key.

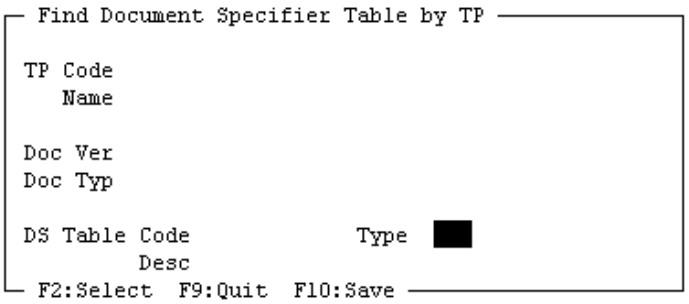
Step	Action
1	Select **Doc Loc** from the host Util menu. **System Response** Sterling Gentran:Server displays the Document Reference Number Specifier screen. 
2	Select the DS Table Code field.

(Contd) Step	Action
3	Press F2 to display a list of document specifier tables. 
4	Select the name of the table and press ENTER. **System Response** The system enters the table code, type, and description in the DS Table fields.
5	GO TO How to Copy a Document Specifier Table.

Locating a table by the Trading Partnership code

Use this procedure to find a table by the Trading Partnership code.

Step	Action
1	Select **Doc Loc** from the host Util menu. **System Response** Sterling Gentran:Server displays the Document Reference Number Specifier screen. 
2	Select the DS Table Code field.

(Contd) Step	Action
3	Press F5 to use the Trading Partnership code to locate the table. **System Response** The system displays the Find Document Specifier Table by TP Search screen. 
4	Complete the **Type** field with the kind of document specifier table (EDI, APP, XML, NCP). This is a required field.
5	Select the TP Code field and press F2 to display a list of Trading Partnership codes attached to the table type you indicated.
6	Select the Trading Partnership code from the list. **WARNING** **You can select only one Trading Partnership code.** **System Response** Sterling Gentran:Server displays the table values in the Find Document Specifier Table by TP screen.
7	Note the DS Table name displayed. Is this the table you want to copy? ◗ If YES, press F10 to save the selection and continue. ◗ If NO, press F9 to clear the fields and start over. **System Response** When you press F10, the system enters the DS Table Code and Description in the Document Reference Number Specifier screen.
8	GO TO How to Copy a Document Specifier Table.

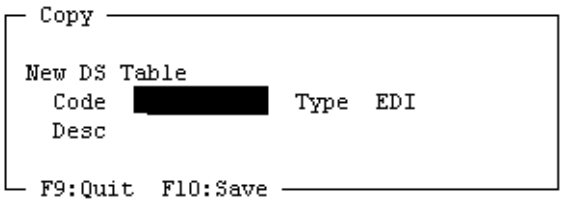
How to Copy a Document Specifier Table

Introduction

When you copy a document specifier table, you also copy the table's set ID, application name, XML name, or NCPDP transaction code maps. The copy function does not copy the Trading Partnership code attachment, so you should attach the Trading Partnership code after copying the table.

Copying a document specifier table

Use this procedure to copy a document specifier table.

Step	Action
1	Display the table you want to copy by following the instructions in How to Display a Document Specifier Table.
2	Press F4 to copy the table. **System Response** The system displays the Copy screen. 
3	Enter the code and description for the new table in the appropriate fields. **System Response** The system enters the new table code and description in the Document Reference Specifier screen.
4	Press F10 to save the table.
5	Do you want to edit the new table's maps? ◗ If YES, see Maintaining Document Specifier Maps. ◗ If NO, continue with Step 6.

(Contd) Step	Action
6	Do you want to delete any of the new table's maps? ◗ If YES, see How to Delete a Map from a Table. ◗ If NO, continue with Step 7.
7	GO TO the Attaching a Trading Partnership Code to a Table section in this chapter.

Mapping Document Specifier Tables

Overview

Introduction

After you create a document specifier table, you build set ID, application name, XML name, or NCPDP transaction code maps for the table. You can have one or more maps in a document specifier table.A table can have any number of maps.

Map functions

Set ID, application name, XML name, and NCPDP transaction code maps show the data managers and the translator where in a document they should look for the unique values to use in constructing the document reference number.

Set ID maps

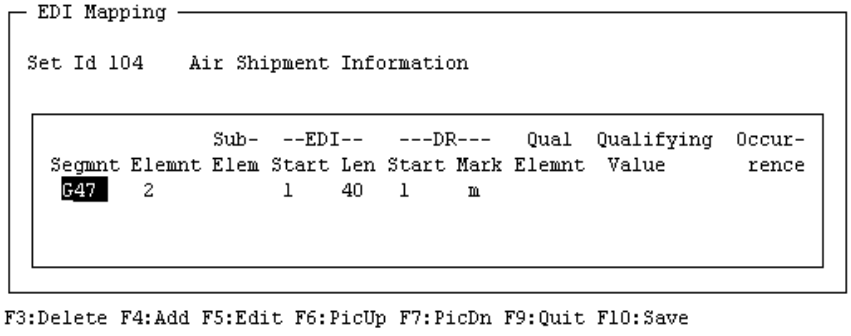
In a set ID map, the set ID represents the document type (for example, 850, 810, 864) for EDI data.

A set ID map contains one or more lines, each of which specifies:

- Segment name and element or sub-element in the document that contains the value you want extracted.
- Length (in number of characters) of the value. The document reference number can be up to 40 characters in length. If your map contains multiple segment and element lines, **the total length cannot exceed 40 characters**.
- Position in the document reference number that the extracted value occupies.
- Conditions, if any, under which the value is extracted.

Example

This is an example of a set ID map.



Application name maps

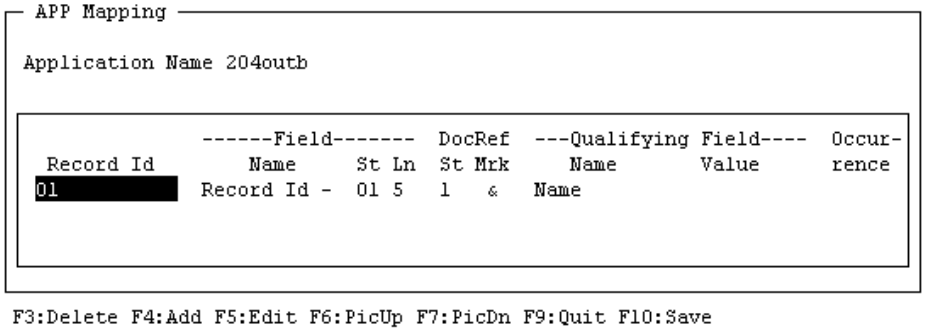
An application name map is a record that identifies the records and fields that the data manager extracts from the application transaction to make the document reference number. The application name is a file definition (<filename>.ddf) or an application description (<filename>.app) you defined to Sterling Gentran:Server.

An application name map contains one or more lines, each of which specifies:

- Record ID and field in the document from which to extract a value.
- Length (in number of characters) of the value. The document reference number can be up to 40 characters in length. If your map contains multiple segment and element lines, **the total length cannot exceed 40 characters**.
- Position in the document reference number that the extracted value occupies.
- Conditions, if any, under which the value is extracted.

Example

This is an example of an application name map.



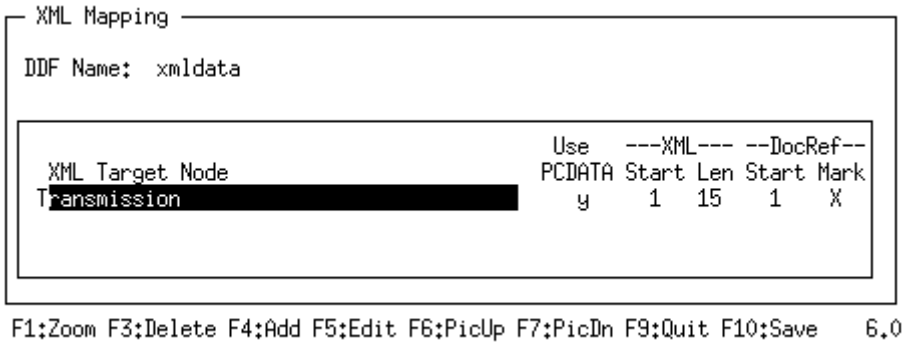
XML name maps

If you have the Sterling Gentran:Server XML translation option, you can create XML name maps from a DDF file. An XML name map identifies the series of XML tags representing the path to specific (target) nodes in the DDF document. The data manager extracts either the tag names or the PCDATA in the tags to make the document reference number.

An XML name map contains one or more lines, each of which specifies:

- XML node that defines the path to the tag name. Each line in the map must have a unique “target” node.
- Whether to use PCDATA instead of the tag name
- Length (in number of characters) of the value
- Position in the document reference number that the extracted value occupies
- The symbol used to mark this part of the document reference number in the map picture.

Example
This is an example of an XML name map.



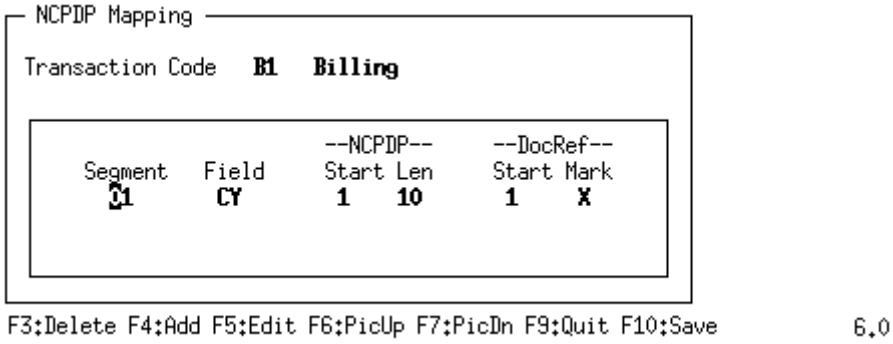
NCPDP transaction code maps

In an NCPDP transaction code map, the transaction code represents the document type (for example, B1, B2, B3) for NCPDP data.

A transaction code map contains one or more lines, each of which specifies:

- Segment name and field in the document that contains the value you want extracted.
- Length (in number of characters) of the value. The document reference number can be up to 40 characters in length. If your map contains multiple segment and element lines, the total length cannot exceed 40 characters.
- Position in the document reference number that the extracted value occupies.

Example
This is an example of an NCPDP transaction code map.



Maximum lines in a map

You can specify up to 99 segment or record lines in a document reference number map, including conditional statements. The total length of the document reference number cannot exceed 40 characters. Each segment or record line (excluding conditional statements) in the map makes up part of the document reference number.

Repeating segments or records

A set ID or application name map may contain a segment or record ID that is repeated in the document. This table describes which occurrence of a repeating segment or record ID the system uses for the document reference number.

Note
This table does not apply to XML name and NCPDP transaction code maps.

IF…	THEN the data manager and translator use the…
You specified the occurrence that you want used	Specified occurrence
You did not specify the occurrence that you want used	Last occurrence to determine that part of the document reference number
The repeating segments are part of an interchange header	First occurrence of the segment

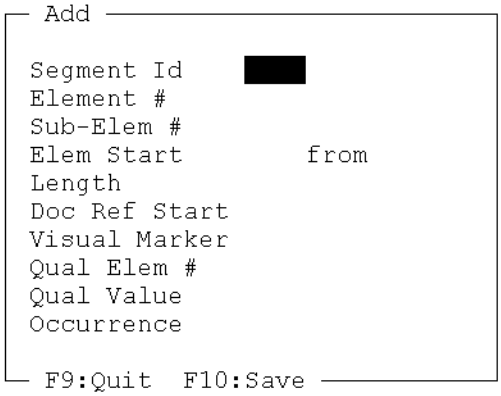
EDI Add Screen

Introduction

To create an EDI map for a document specifier table, you use the EDI Add screen. You must complete one screen for each line in the map.

Illustration

This is the EDI Add screen.



EDI Add screen fields and functions

This table lists the fields on the EDI Add screen and their functions.

Field	Description
Segment ID	Defines the segment in the document that contains the unique value. The Segment ID has a maximum length of 4 characters. **Examples** BEG, DTM, EDF
Element #	Identifies the element number in the segment that contains a unique value. The element number can be any number from 1 to 999. inclusive.
Sub-Elem #	The sub-element number in the element that contains a unique value. The sub-element can be any number from 1 to 99, inclusive.

(Contd) Field	Description

Elem Start from	The character number of the selected element or sub-element character that starts the unique value. You must also specify whether the position is counted **from** the left or the right of the element value. The Elem Start from value can be any number from 1 to 99, inclusive. Enter **Left** or **right** for counting direction.
	
	Comments
	Specify “left” to extract a fixed number of characters in a known position beginning from the left of the element value.
	
	Specify “right” to extract a fixed number of characters in a known position beginning from the right of the element value.
	
	Example
	The element you want to extract values from is variable length and you want to extract the last three characters from the right.
	
	`Elem Start = 1`
	`from = right`
	`Length = 3`
	If the element value is YRG73945, the system extracts “945” as the unique value. If the element value is YRG341, the system extracts “341” as the unique value.

Length	The length (in number of characters) of the unique value. This value can be any number from 1 to 40, inclusive **Comments** If the length you specify is longer than the length of the element, the system uses the full length unless it encounters a sub-element separator, an element separator, or a segment terminator (end of line). The total length of all lines in the map cannot exceed 40 characters.
Doc Ref Start	The starting place (character number) in the document reference number that this unique value occupies. The Doc Ref Start element can be any number from 1 to 40, inclusive.

(Contd) Field	Description
Visual Marker	The symbol you want to use to mark this part of the document reference number in the map picture. The Visual Marker is only 1character in length. Examples: %, &, *, A, B **Comment** The system does not allow you to use the same marker more than once. If you have used the marker for another line, Sterling Gentran:Server displays a message to let you know you must choose another marker. **Reference** See How to Display a Map Picture.
Qual Element #	The 3-digit element number used with the qualifying value (Qual Value field) that helps determine when the segment and element characters are to be used in the document reference number. The qualifying element must be in the same segment as the element specified in the Element # field and can be any number from 1 to 999, inclusive. **Comment** When the element with this number has the value specified in the Qualifying Value field, the specified characters are extracted. Otherwise, the characters are not used. **Reference** See Using the Qualifier and Occurrence Fields for more information.

(Contd) Field	Description
Qual Value	The value that the element number in the Qual Element field must have before Sterling Gentran:Server extracts the specified segment and element or sub-element characters for the document reference number. This value can be up to 10 characters in length. **Example** 102097 means “if 102097 is the qualifying value, then extract values from specified segment and element”
Occurrence	The occurrence of the segment from which the specified characters are to be extracted. This field can be up to 5 characters in length. ◗ 1 means use the first occurrence of the segment. ◗ 2 means use the second occurrence of the segment. ◗ 0 means use the last occurrence of the segment. **Comments** If this field is empty, Sterling Gentran:Server uses the last occurrence of the segment. For interchange segments, Sterling Gentran:Server uses only the first and last occurrences. Therefore, you must have either 0 (last) or 1 (first) in the Occurrence field to extract the interchange segment contents.

Defaults

The Doc Ref Start value defaults to 1 in the first segment line you add. The starting position of subsequent segment lines defaults to a number determined by the length in the previous segment or record lines.

Example
If the length specified in the first line is 6, the Doc Ref Start value in the second line defaults to 7. You can override the default values.

EDI Add screen function keys

This table describes the function keys of the EDI Add screen and their functions.

Key	Function
F2	Displays a list of choices for the selected field.

(Contd) Key	Function
F9	Exits the screen.
F10	Saves the line to the **Set ID** map and clears the screen so that you can add another segment line to the map.

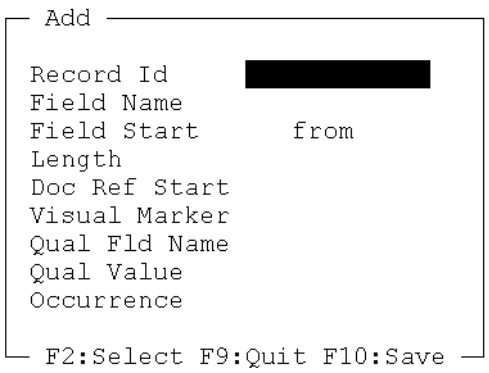
APP Add Screen

Introduction

To create an APP map for a document specifier table, you use the APP Add screen. You must complete one screen for each line to the map.

Illustration

This is the APP Add screen.



APP Add screen fields and functions

This table lists the fields on the APP Add screen and their functions.

Field	Description
Record ID	Defines the record identifier in the document that contains a unique value. This value can be up to 128 characters in length, though only 12 are displayed **Example** 01 means the 01 record
Field Name	Defines the field name. This value can be up to 128 characters, though only 12 are displayed. **Example** weight

(Contd) Field	Description

Field Start
 	
from
	Defines the character number in the selected field that starts the unique value. You must also specify whether the position is counted **from** the left or the right of the field. This value can be any from 1 to 99, inclusive. Use **Left** or **right** for counting direction.
	
	Comments
	Specify “left” to extract a fixed number of characters in a known position beginning from the left of the field.
	
	Specify “right” to extract a fixed number of characters in a known position beginning from the right of the field.
	
	Example 1
	The field you want to extract values from is variable length and you want to extract the last three characters from the right.
	
	`Field Start = 1`
	`from = right`
	`Length = 3`
	If the field value is YRG73945, the system extracts “945” as the unique value. If the field value is YRG341, the system extracts “341” as the unique value.
	
	Example 2
	The field you want to extract from is a 5-character fixed-length field and you want to extract the second and third characters from the left.
	
	`Field Start = 2`
	`from = left`
	`Length = 2`
	If the field value is 39386, the system extracts “93” as the unique value. If the field value is M7839, the system extracts “78” as the unique value.

Length	Defines the number of characters in the unique value. This value can be any number from 1 to 40, inclusive. **Comment** The total length of all record lines in the map cannot exceed 40.
Doc Ref Start	Designates the starting place (character number) in the document reference number that this unique value occupies. This value can be any number from 1 to 40 for a starting character number.

(Contd) Field	Description
Visual Marker	Defines the symbol you want to use to mark this part of the document reference number in the map picture. This is a single-character value. Examples: %, &, * **Comment** The system does not allow you to use the same marker more than once. If you already used the marker for another line, Sterling Gentran:Server displays a message. You must choose another marker. **Reference** See How to Display a Map Picture.
Qual Fld Name	Defines the field name used with the qualifying value that helps determine when the record ID and field characters will be used in the document reference number. This value could be up to 128 characters, though only 12 characters are displayed. **Comment** When the field with this name has the value specified in the **Qual Value** field, the specified characters are extracted. Otherwise, the characters are skipped. **Reference** See Using the Qualifier and Occurrence Fields for more information.
Qual Value	Defines the value that the field named in the Qualifying Field Name field must have before the specified characters in the field and record are extracted. This value can be up to 10 characters in length. **Example** 40
Occurrence	If the document has multiple records with the same name, this field specifies the occurrence of the record to be used. This value can be 5-digits in length. **Example** A 3 means that Sterling Gentran:Server uses the third occurrence of the record. **Comment** If this field is empty, Sterling Gentran:Server uses the last occurrence.

Defaults

The Doc Ref Start value defaults to 1 in the first record line you add. The starting position of subsequent record lines defaults to a number determined by the length in the previous record lines.

Example
If the length specified in the first line is 6, the Doc Ref Start value in the second line defaults to 7. You can override the default values with a greater value.

APP Add screen function keys

This table describes the function keys of the APP Add screen and their functions.

Key	Function
F2	Displays a list of choices for the field.
F9	Exits the screen.
F10	Saves the line to the application name map and clears the screen so that you can add another record line to the map.

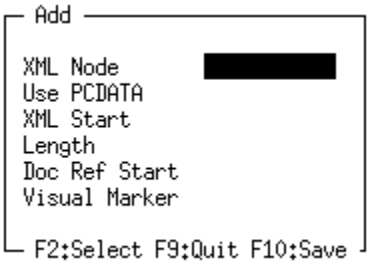
XML Add Screen

Introduction

To create a XML map for a document specifier table, you use the XML Add screen. You must complete one screen for each line to the map.

Illustration

This is the XML Add screen.



XML Add screen fields and functions

This table lists the fields on the XML Add screen and their functions.

Field	Description
XML Node	Defines the path to the target tag in the XML document. This value can be up to 128 characters in length, though only 12 are displayed. **Example** 01 means the 01 record
Use PCDATA	Indicates whether or not the PCDATA data in the target tag is used to build the document reference number. This value can be either *Yes* or *No*. **Note** If you select No for this field, the system extracts the tag name for the document reference number instead of the PCDATA.
XML Start	Defines the character number in the selected tag that starts the unique value. This value can be any number from 1 to 99, inclusive.

(Contd) Field	Description
Length	Defines the number of characters in the unique value. This value can be any number from 1 to 40, inclusive. **Comment** The total length of all record lines in the map cannot exceed 40.
Doc Ref Start	Designates the starting place (character number) in the document reference number that this unique value occupies. This value can be any number from 1 to 40 for the starting character number.
Visual Marker	Defines the symbol you want to use to mark this part of the document reference number in the map picture. This is a single-character value. Examples: %, &, * **Comment** The system does not allow you to use the same marker more than once. If you already used the marker for another line, Sterling Gentran:Server displays a message. You must choose another marker. **Reference** See How to Display a Map Picture.

Defaults

The Doc Ref Start value defaults to 1 in the first target node line you add. The starting position of subsequent record lines defaults to a number determined by the length in the previous record lines.

Example
If the length specified in the first line is 6, the Doc Ref Start value in the second line defaults to 7. You can override the default values with a greater value.

XML Add screen function keys

This table describes the function keys of the XML Add screen and their functions.

Key	Function
F2	Displays a list of choices for the field.
F9	Exits the screen.
F10	Saves the line to the XML name map and clears the screen so that you can add another record line to the map.

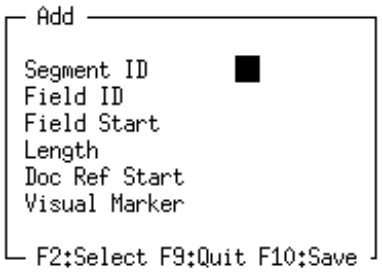
NCPDP Add Screen

Introduction

To create an NCPDP transaction code map for a document specifier table, you use the NCPDP Add screen. You must complete one screen for each line in the map.

Illustration

This is the NCPDP Add screen.



NCPDP Add screen fields and functions

This table lists the fields on the NCPDP Add screen and their functions.

Field	Description
Segment ID	Defines the segment in the document that contains the unique value identifying the data. This value is two characters in length. **Examples** 00, G1, 01
Field ID	Defines the field in the segment that contains a unique value. This value can be any number from 1 to 99 inclusive.
Field Start	The character number of the character in the selected field that starts the unique value. This value can be any number from 1 to 99, inclusive.

(Contd) Field	Description
Length	The length (in number of characters) of the unique value. This value can be any number from 1 to 40, inclusive. **Comments** If the length you specify is longer than the length of the field, the system uses the full length unless it encounters a group separator, field separator, or a segment terminator (end of line). **Note** The total length of all lines in the map cannot exceed 40.
Doc Ref Start	The starting place (character number) in the document reference number that this unique value occupies. This value can be any number from 1 to 40, inclusive.
Visual Marker	The symbol you want to use to mark this part of the document reference number in the map picture. This value is a single character in length. **Comment** The system does not allow you to use the same marker more than once. If you have used the marker for another line, Sterling Gentran:Server displays a message to let you know you must choose another marker. **Reference** See How to Display a Map Picture. **Examples** %, &, *, A, B

Defaults

The Doc Ref Start value defaults to 1 in the first segment line you add. The starting position of subsequent lines defaults to a number determined by the length in the previous record lines.

Example
If the length specified in the first line is 6, the Doc Ref Start value in the second line defaults to 7. You can override the default values with a greater value.

NCPDP Add screen function keys

This table describes the function keys of the NCPDP Add screen and their functions.

Key	Function
F2	Displays a list of choices for the field.
F9	Exits the screen.
F10	Saves the line to the transaction code map and clears the screen so that you can add another line to the map.

How to Add a Map to a Document Specifier Table

Introduction

Adding a set ID, application name, XML name, or NCPDP transaction code map to a document specifier table is the second task in defining the values that you want to comprise the document reference number.

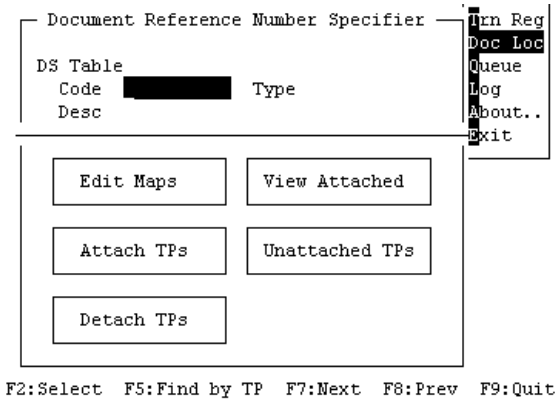
Before you begin

You must create your application descriptions, implementation guides, and file definitions before adding their names to a document specifier table.

Reference
See the *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide* for instructions.

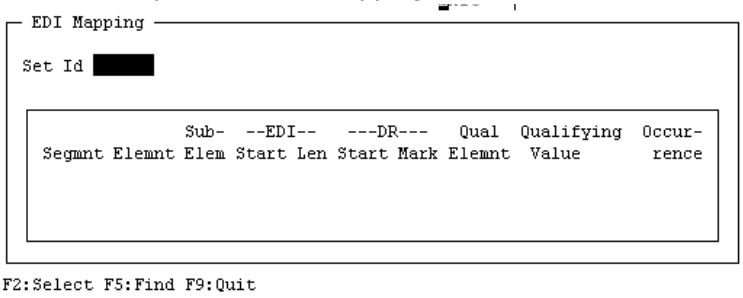
Adding a map to a document specifier table

Use this procedure to add a set ID, application name, XML name, or NCPDP transaction code map to a document specifier table.

Step	Action
1	Select **Doc Loc** from the host Util menu. **System Response** Sterling Gentran:Server displays the Document Reference Number Specifier screen. 
2	Display the table you want to map. **Reference** See How to Display a Document Specifier Table.

<table>
<tr><th>(Contd) Step</th><th>Action</th></tr>
<tr><td>3</td><td>Select Edit Maps and then press ENTER.

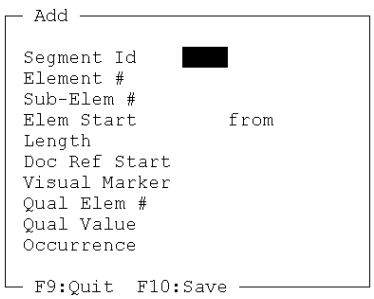
System Response
The system displays a Mapping screen for the table type you specified (EDI, APP, XML, or NCP). The screen's fields are blank.

Example
This is an example of the EDI Mapping screen.
 
</td></tr>
<tr><td>4</td><td>Type the set ID, application name, DDF name, or NCPDP transaction code in the first field and then press ENTER.

TIP
To select from a list of all existing set IDs, application names, or DDFs, press F2.

System Response
<table>
<tr><th>IF the set ID, application name, or DDF…</th><th>THEN Sterling Gentran:Server…</th><th>AND you should…</th></tr>
<tr><td>Is mapped to the document specifier table</td><td>Displays the map</td><td>Press F10 to clear the screen; then enter a different name.</td></tr>
<tr><td>Is not mapped to the document specifier table</td><td>Displays a prompt that asks if you want to add the map</td><td>Type y to continue.

System Response
The system displays the Add screen for the first line.</td></tr>
<tr><td>Does not describe the layout of an application file</td><td>Displays an error message</td><td>Select a different DDF.</td></tr>
</table></td></tr>
</table>

(Contd) Step	Action
5	Complete the fields. 
6	Press F10 to save the line to the map. **System Response** Sterling Gentran:Server clears the fields in the Add screen so that you can add another segment or record line to the map.
7	Repeat Step 5 to add each subsequent line in your map. **REMINDER** **The values that you define must result in a unique number for each document.**
8	When your map is finished, press F10 to save the map and then press F9 to exit the Add screen. **System Response** If you mapped multiple segments and elements, the system checks to make sure the total length is less than or equal to 40 (the maximum length of the document reference number). Sterling Gentran:Server displays the mapping screen, which now lists all the lines you added to the map. **TIP** **You can display a picture of the map by pressing F6.** **Reference** See How to Display a Map Picture for details.

Using the Qualifier and Occurrence Fields

Introduction

The EDI Add and APP Add screens have optional fields. These fields enable you to define the circumstances under which you want the system to use the map line.

These are the optional fields:

- **Qual Element #** and **Qual Value** for EDI map lines
- **Qual Fld Name** and **Qual Value** for APP map lines
- **Occurrence** for either EDI or APP map lines.

Purpose of the qualifier fields

In an EDI map, you use the qualifying element number (Qual Element #) field with the qualifying value (Qual Value) field to define the circumstances under which Sterling Gentran:Server is to extract the value defined by the map line.

In an APP map, you use the qualifying field name (Qual Fld Name) with the qualifying value (Qual Value) field for the same purpose.

When you enter information into the pair of fields, you instruct Sterling Gentran:Server to extract the characters only if the value of the qualifying element number or qualifying field name matches the value in the qualifying value field.

Example

```
Segment ID = N1
Element #= 2
SubElem # = (None)
Elem Start = 1
Length = 40
Doc Ref Start = 1
Visual Marker = ~
Qual Elem # = 3
Qual Value = MA
Occurrence = (None)
```

In this example, Sterling Gentran:Server extracts the contents of element 2 in the N1 segment only if the value of element 3 is MA.

Purpose of the Occurrence field

Use the Occurrence field to specify which occurrence of the segment or record you want Sterling Gentran:Server to use for the document reference number.

Example

A 4 in the Occurrence field means use the fourth occurrence.

Option precedence

If you choose to use optional qualifiers in a map line, you can base the value extraction on the circumstances, the occurrence, or both.

This table describes the process that Sterling Gentran:Server follows when you use the optional fields in a map line.

IF you use…	THEN Sterling Gentran:Server…
The Occurrence field only	Extracts the value from the specified occurrence of the segment and element or record and field.
The two qualifier fields only (Qual Element # and Qual Value for EDI map lines or Qual Fld Name and Qual Value for APP map lines)	Extracts the value only if the Qual Element # or Qual Fld Name contains the value you specified in the Qual Value field.

The Occurrence field and the two qualifier fields (Qual Element # and Qual Value for EDI map lines or Qual Fld Name and Qual Value for APP map lines)	Looks for the specified occurrence that has the qualifying value in the Qual Value field and extracts the value.
	Example
	`If the:`
	`Occurrence is 3`
	`Qualifying Element # is 9`
	`Qualifying Value is 38`
	then Sterling Gentran:Server looks for the third occur-rence in which the value of the ninth element is 38.

CAUTION

Sterling Gentran:Server cannot distinguish between interchange segments and other segments. For interchange segments, Sterling Gentran:Server uses only the first and last occurrences. Therefore, you must have either 0 (for last occurrence) or 1 (for first occurrence) in the Occurrence field to extract the interchange segment contents.

How to Display a Map Picture

Introduction

Once you have created an EDI, APP, XML, or NCPDP document specifier map, you can display a map picture to help you determine whether or not the map will extract the correct values.

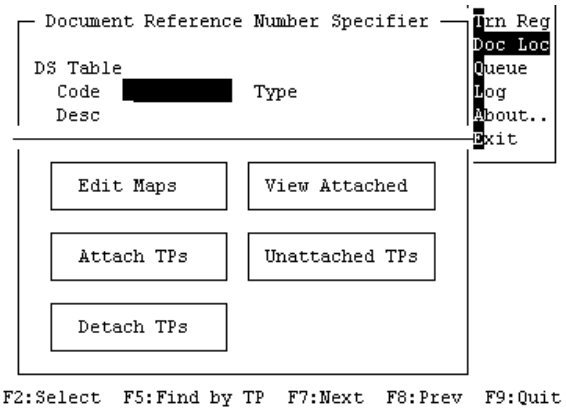
Map picture illustration

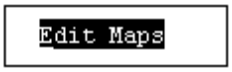
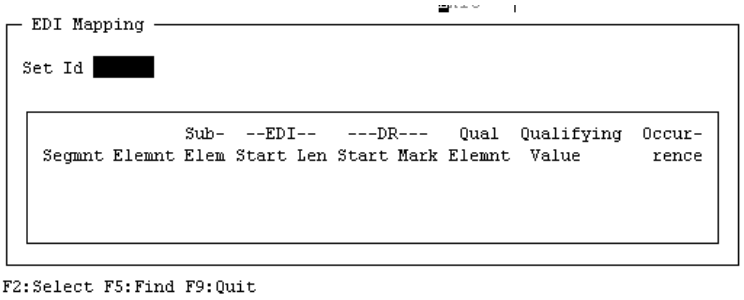
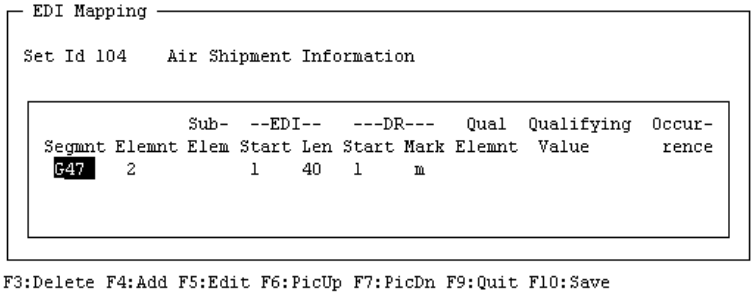
This illustration shows a map picture.

```
BEG                 EDF       DTM
|                   |         |
|                   |         |
%                   ******    ))))))))))
----------------------------------------
12345678901234567890123456789012345678901
|         1         | 2       |  3          4
|                   |         |
1                   18        27
```

Displaying a map picture

Use this procedure to display a map picture.

Step	Action
1	Select **Doc Loc** from the host Util menu. **System Response** Sterling Gentran:Server displays the Document Reference Number Specifier screen. 
2	Select the DS Table Code field.

(Contd) Step	Action
3	Press F2 to display a list of document specifier tables and then select the table that has the map you want to view.
4	Select **Edit Maps** and then press ENTER.  **System Response** Sterling Gentran:Server displays the Mapping screen. 
5	Select the **Set ID, Application Name**, **DDF Name,** or **Transaction Code** field.
6	Press F5 to display a list of maps for the table, and then select the map you want to view. **System Response** Sterling Gentran:Server displays the segment or record lines in the map. 

(Contd) Step	Action
7	Press F6 to display a picture of the map. **System Response** The markers you used in the map identify the place in the document reference number that the segment or record values occupy. **CAUTION** **You can edit the map while the picture is displayed.**
8	When you are finished, press F7 to remove the picture from your screen.

Attaching a Trading Partnership Code to a Table

How to Attach a Trading Partnership Code to a Table

Introduction

After you create a document specifier table and its maps, you must attach a Trading Partnership code to the table. The link enables the data manager or translator to apply the set of rules for extracting the document reference number to the documents of a particular Trading Partnership.

Rules for Trading Partnership attachments

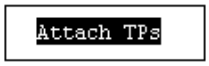
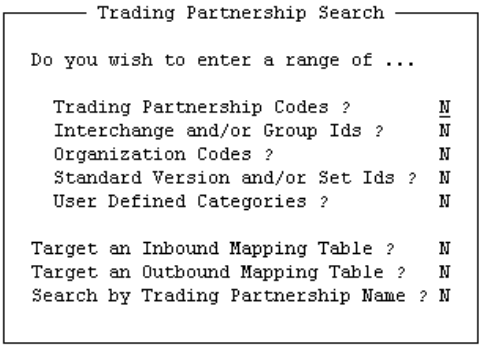
The following rules apply to Trading Partnership attachments:

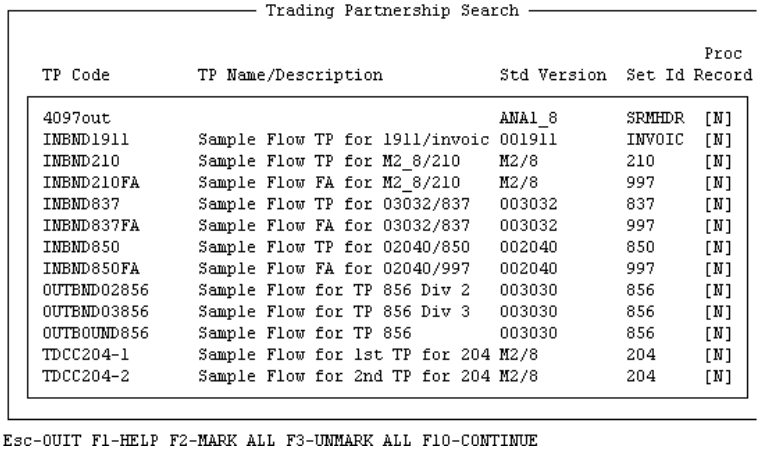
- You can attach several Trading Partnership codes to a single table.
- You can attach a specific Trading Partnership code to only one table.

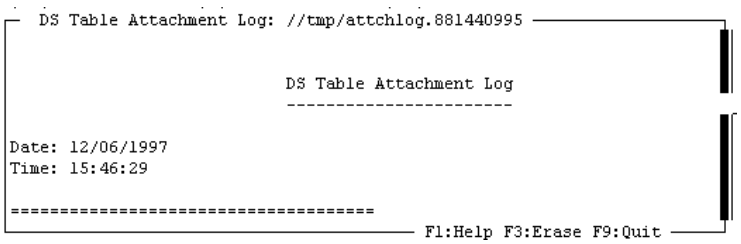
Procedure

Use this procedure to attach a Trading Partnership code to a table.

Step	Action
1	Display the table to which you want to attach Trading Partnership codes. **Reference** See How to Display a Document Specifier Table in this chapter.

(Contd) Step	Action
2	Select **Attach Trading Partners**; then press ENTER.  **System Response** The system displays the Trading Partnership Search screen.  **Reference** See the Understanding the Basics chapter in this guide for instructions on how to search for a Trading Partnership.

(Contd) Step	Action
3	Type search criteria on the screen for the Trading Partnerships you want to see, then press F10 to continue. **CAUTION** **To list all the Trading Partnerships, press F10 without entering search criteria.** **System Response** The system displays a list of Trading Partnerships that matched the criteria you entered. 
4	Select the Trading Partnerships you want to attach to the table by entering **y** in the Trading Partnership's Proc Record (Process Record) field. **Note** The following shortcut keys are available: ◗ To select every Trading Partnership code, press F2 (Mark All). ◗ To remove all the **y**'s from the Proc Record field, press F3 (Unmark All).

(Contd) Step	Action
5	Press F10 to continue. **System Response** Sterling Gentran:Server creates a Trading Partnership/Table record for each Trading Partnership attached to the table. When the attachment process has ended, the system displays the DS Table Attachment Log.  **Reference** See the The Screen Viewer topic in the Monitoring Processes chapter of this guide for information about navigating the log.
6	Check the **Status** field of the DS Table Attachment Log to verify that the Trading Partnership codes attached successfully to the table. **Note** The DS Table Attachment Log is a temporary file. When you leave this screen, Sterling Gentran:Server deletes the file.
7	Press ESC to exit the table.

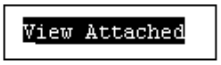
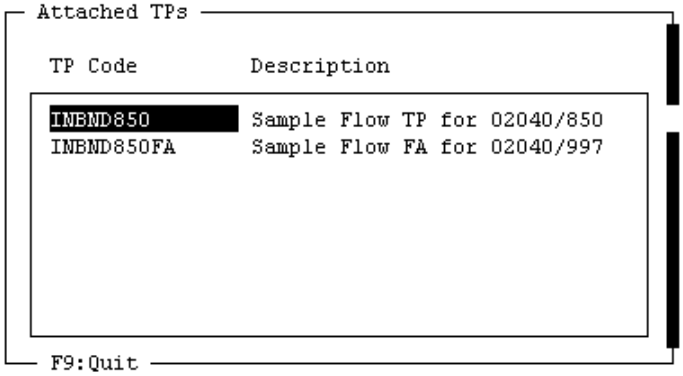
How to Verify Trading Partnership Code Attachments

Introduction

To verify that Trading Partnership codes are attached or not attached to a document specifier table, you can view attachment lists.

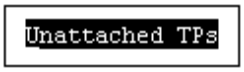
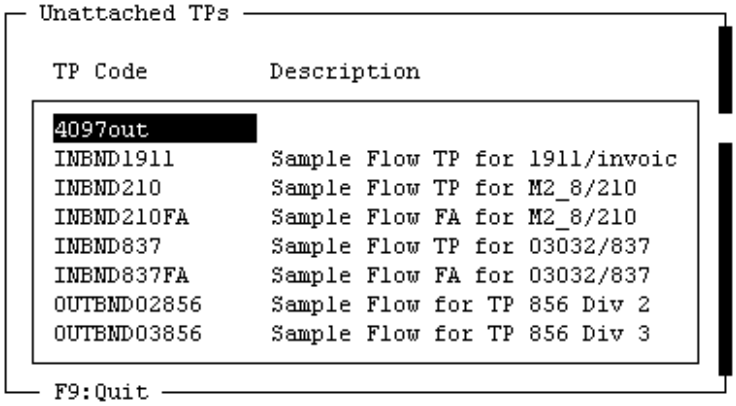
Viewing a list of Trading Partnerships attached to a table

Use this procedure to view a list of Trading Partnership records attached to a table.

Step	Action
1	Display the table for which you want to view a list of the attached Trading Partnership codes. **Reference** See the How to Display a Document Specifier Table topic in this chapter for instructions.
2	Select **View Attached** and then press ENTER.  **System Response** The system displays the Attached TPs screen, which lists the Trading Partnership codes linked to the table. 
3	Press F9 to exit the **Attached TPs** screen.

Viewing a list of unattached Trading Partnership codes

Use this procedure to view a list of Trading Partnership codes that are not attached to a table.

Step	Action
1	Select **Doc Loc** from the host Util menu to display the Document Reference Number Specifier screen.
2	Click **Unattached TPs** and then press ENTER.  **System Response** The system displays the Unattached TPs screen. It lists all the Trading Partnership codes that are not linked to any table. 
3	Press F9 to exit the list and return to the Document Reference Number Specifier screen.

Maintaining Document Reference Number Tables

Overview

In this section

This section contains procedures for:

- Changing a Trading Partnership code and table attachment
- Removing a Trading Partnership code attachment from a table
- Deleting a document specifier table.

CAUTION

Any change you make to a document specifier table affects all the Trading Partnership code records attached to that table.

How to Change a Trading Partnership Code and Table Attachment

Introduction

Sterling Gentran:Server allows you to attach a Trading Partnership code to only one table. When you attach a Trading Partnership code that is already attached to a table to a different table, the new attachment overrides and replaces the previous link.

Procedure

To change a Trading Partnership code and table attachment, attach the Trading Partnership code to the new table.

Reference

See How to Attach a Trading Partnership Code to a Table.

How to Remove a Trading Partnership Code from a Table

Introduction

If a Trading Partnership code is attached to the wrong table, you can remove the attachment.

Procedure

Use this procedure to remove a Trading Partnership code attachment from a table.

Step	Action
1	Display the table from which you want to attach Trading Partnership codes. **Reference** See the How to Display a Document Specifier Table topic in this chapter.
2	Select **Detach Trading Partners** and press ENTER. Detach TPs **System Response** The system displays a list of all the Trading Partnership codes attached to the table. Detach TPs TP Code: TP Desc: Table Code: Detach? 1 x12 to fixed app DSTLB1 [N] 10 Edifact to x12: DSTLB1 [N] 10174 DSTLB1 [Y] 10759 10759 OUTBOUND v DSTLB1 [N] 10821 Kellogg's Inboun DSTLB1 [Y] 10930 -n flag v3040 94 DSTLB1 [N] 10b edifact to x12 : DSTLB1 [N] 11 fixed app to fix DSTLB1 [Y] 11010 Outbound ST:02 t DSTLB1 [N] Esc-Quit F1-Help F2-Mark All F3-Unmark all F10-Continue

(Contd) Step	Action
3	Select each Trading Partnership you want to detach from the table by entering **y** in the Detach field. **Note** The following shortcut keys are available: ◗ To select every Trading Partnership code, press F2 (Mark All). ◗ To remove all the **y**'s from the Detach field, press F3 (Unmark All).
4	Press F10 to continue. **System Response** Sterling Gentran:Server deletes the Trading Partnership/ Table record from the table. When the detachment process ends, the system displays a log to show which Trading Partnership codes were detached.
5	Check the Status field of the log to verify that the Trading Partnership codes were successfully removed from the table.
6	Press ESC to exit the table.

How to Delete a Document Specifier Table

Introduction

If you do not need a document specifier table, you can delete it from your system.

WARNING

When you delete a table, the system deletes only the records that attach the Trading Partnership codes to the table. The actual Trading Partnership records are not affected.

Procedure

Use this procedure to delete a document specifier table.

Step	Action
1	Display the table that you want to delete. **Reference** To display a table, see How to Display a Document Specifier Table.
2	Press TAB until the Desc field is selected.
3	Press F3 to delete the document specifier table. **System Response** The system counts the number of Trading Partnership records that use the table and displays this information in a Confirmation prompt. Confirmation Delete DS Table: Default EDI Default Mapping Are You Sure (y/n) ? _
4	Type **y** at the prompt to confirm the deletion.

CAUTION

When you delete a document specifier table to which a Trading Partnership record was attached, the data manager treats subsequent documents as though the Trading Partnership record is not attached to a table.

Maintaining Document Specifier Maps

Overview

In this section

This section contains these topics:

- EDI Mapping Screen
- APP Mapping Screen
- XML Mapping Screen
- Transaction Code Mapping Screen
- How to Display a Document Specifier Map
- How to Change a Document Specifier Map
- How to Delete a Map from a Table

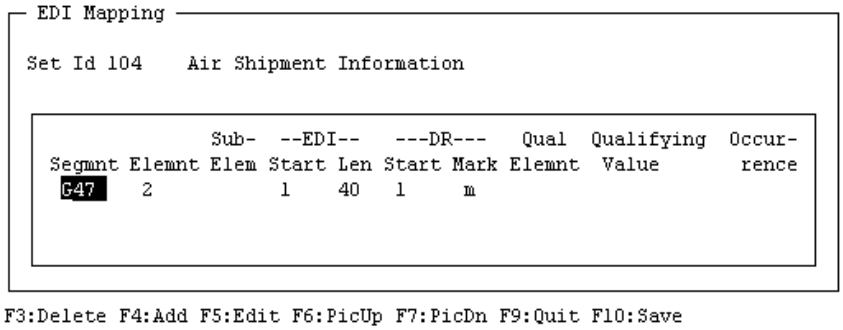
EDI Mapping Screen

Introduction

To modify an EDI map for a document specifier table, you use the EDI Mapping screen.

Illustration

This is the EDI Mapping screen.



EDI Mapping screen fields and functions

This table lists the fields on the EDI Mapping screen and their functions.

Field	Function Max. Length Values
Set ID	Defines the document number that represents the a document or transaction. This value can be up to six positions in length. **Example** 850 for purchase order
Segmnt	Defines the segment in the document that contains the unique value. This value can be up to four positions in length. **Examples** BEG, DTM, EDF
Elemnt	The element number in the segment that contains a unique value. This value can be any number from 1 to 999, inclusive.
Sub-Elem	The sub-element number that contains a unique value. This value can be any number from 1 to 99, inclusive.

(Contd) Field	Function Max. Length Values
EDI-Start	The starting position (character number) of the character in the element or sub-element that starts the unique value. This value can be any number from 1 to 99, inclusive.
Len	The length (in number of characters) of the unique value. This value can be any number from 1 to 40, inclusive. **Comments** The system uses the full length unless it encounters a sub-element separator, an element separator, or a segment terminator (end of line). **Note** The sum total length of the lines in the map cannot exceed 40.
DR Start	The starting place (character number) in the document reference number that this unique value occupies. This value can be any number from 1 to 40, inclusive.
Mark	The symbol you want to use to mark this part of the document reference number in the map picture. You must use a different mark on each line. This is a single-character value. **Examples** %, &, *
Qual Elemnt	The element number used with the qualifying value. This value helps determine when Sterling Gentran:Server uses the segment and element characters in the document reference number. This value can be any number from 1 to 999, inclusive. **Comment** When the element with this number has the value specified in the Qualifying Value field, Sterling Gentran:Server extracts the specified characters. Otherwise, the system skips the characters.

(Contd) Field	Function Max. Length Values
Qualifying Value	The value that the element number in the Qual Elemnt field must have before Sterling Gentran:Server extracts the specified segment and element or sub-element characters for the document reference number. This value can be any string up to ten-characters in length.
Occurrence	The occurrence of the segment from which Sterling Gentran:Server extracts the specified characters. This value can be any number up to five characters in length. **Comment** If this field is empty, Sterling Gentran:Server uses the last occurrence of the segment. **Examples** ◗ 1 means use the first occurrence of the segment. ◗ 2 means use the second occurrence. ◗ 0 means use the last occurrence.

EDI Mapping screen function keys

This table describes the function keys on the EDI Mapping screen and their functions.

Key	Function
F3	Deletes the selected line from the map.
F4	Displays the Add screen so that you can add a line to the map.
F5	Displays the Edit screen for the selected line so that you can change the values in the line.
F6	Displays a pictorial representation of the map.
F7	Removes the map picture from the display.
F9	Exits the screen.
F10	Saves your changes to the map.

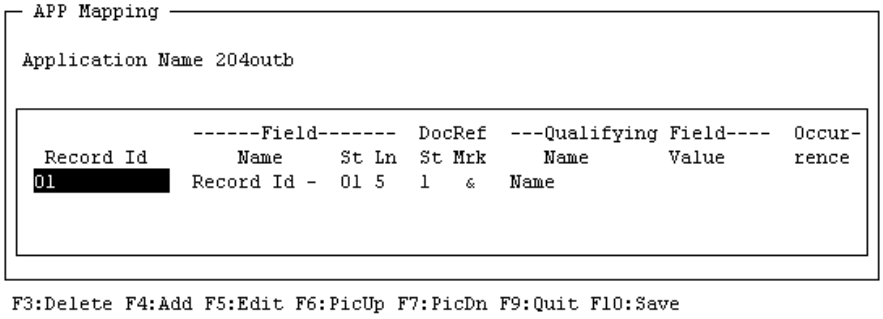
APP Mapping Screen

Introduction

The APP Mapping screen is used to modify an APP map for a document specifier table.

Illustration

This is the APP Mapping screen.



APP Mapping screen fields and functions

This table lists the fields on the APP Mapping screen and their functions.

Field	Description
Application Name	Defines the name of the application file for the document or transaction. This value can be up to six characters in length.
Record ID	Defines the record identifier in the document that contains a unique value. This value can be up to 128 characters in length, though only 12 are displayed.
Field Name	Defines the field name. This value can be up to 128 characters in length, though only 12 are displayed.
(Field) St	Defines the starting place (character number) of the character in the field that starts the unique value. This value can be any number from 1 to 99, inclusive.
Ln	Defines the number of characters in the unique value. This value can be any number from 1 to 40, inclusive. **Comment** The sum total length of the record lines in the map cannot exceed 40.

(Contd) Field	Description
DocRef St	Identifies the starting place (character number) in the document reference number that this unique value occupies. This value can be any number from 1 to 40, inclusive.
Mrk	Defines the symbol you want to use to mark this part of the document reference number in the map picture. This is a single-character value. **Examples** %, &, *
Qualifying Field Name	Defines the field name used with the qualifying value (Value field). These values help determine when Sterling Gentran:Server uses the record ID and field characters in the document reference number. This value can be up to 128 characters in length, though only 12 are displayed. **Comment** When the field with this name has the value specified in the Value field, Sterling Gentran:Server extracts the specified characters. Otherwise, Sterling Gentran:Server skips the characters.
Value	Defines the value that the field named in the Qualifying Field Name field must have before Sterling Gentran:Server extracts the specified characters in the field. This value can be up to ten characters in length.
Occurrence	If the document has multiple records with the same name, this field specifies the occurrence of the record to be used. This value can be up to five characters in length. **Comment** If this field is empty, Sterling Gentran:Server uses the last occurrence. **Examples** ◗ 1 means use the first occurrence of the segment. ◗ 2 means use the second occurrence. ◗ 0 means use the last occurrence.

APP Mapping screen function keys

This table describes the function keys on the APP Mapping screen and their functions.

Key	Function
F3	Deletes the selected line from the map.
F4	Displays the Add screen so that you can add a line to the map.
F5	Displays the Edit screen for the selected line so that you can change the values in the line.
F6	Displays a pictorial representation of the map.
F7	Removes the map picture from the display.
F9	Exits the screen.
F10	Saves your changes to the map.

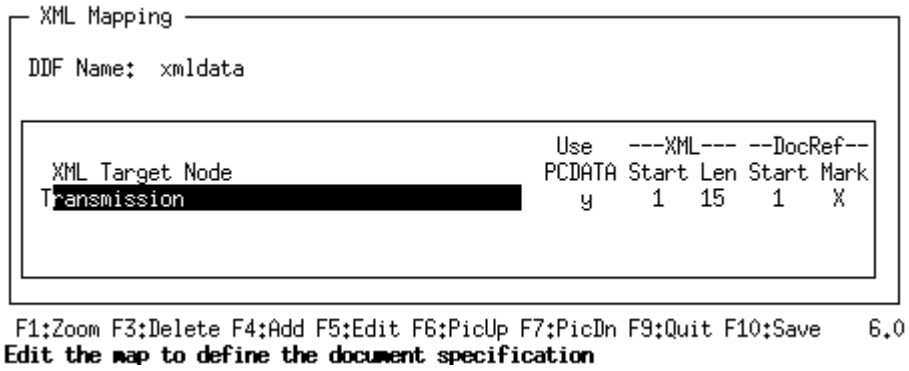
XML Mapping Screen

Introduction

The XML Mapping screen is used to modify an XML map for a document specifier table.

Illustration

This is the XML Mapping screen.



XML Mapping screen fields and functions

This table lists the fields on the XML Mapping screen and their functions.

Field	Description
DDF Name	Defines the name of the DDF file for the document or transaction. This value can be up to six characters in length.
XML Target Node	Defines the path to the target tag in the XML document. This value can be up to 128 characters in length, though only 40 are displayed.
Use PCDATA	Indicates whether or not the PCDATA data in the target tag is used to build the document reference number. This value is either *y* for *Yes* or *n* for *No.* **Note** If you select No for this field, the system extracts the tag name for the document reference number instead of the PCDATA.
XML Start	Defines the character number in the selected tag that starts the unique value. This value can be any number from 1 to 99.

(Contd) Field	Description
Len	Defines the number of characters in the unique value. This value can be any number from 1 to 40. **Comment** The total length of all record lines in the map cannot exceed 40.
Doc Ref Start	Designates the starting place (character number) in the document reference number that this unique value occupies. this value can be any number from 1 to 40, inclusive, for the starting character number.
Visual Marker	Defines the symbol you want to use to mark this part of the document reference number in the map picture. This is a single-character value. **Comment** The system does not allow you to use the same marker more than once. If you already used the marker for another line, Sterling Gentran:Server displays a message. You must choose another marker. **Reference** See How to Display a Map Picture. **Examples** % & *

XML Mapping screen function keys

This table describes the function keys on the XML Mapping screen and their functions.

Key	Function
F1	Displays up to 78 characters of the path to the selected target node in the status bar at the bottom of the screen. Used to extend the view of a truncated line.
F3	Deletes the selected line from the map.
F4	Displays the Add screen so that you can add a line to the map.
F5	Displays the Edit screen for the selected line so that you can change the values in the line.

(Contd) Key	Function
F6	Displays a pictorial representation of the map.
F7	Removes the map picture from the display.
F9	Exits the screen.
F10	Saves your changes to the map.

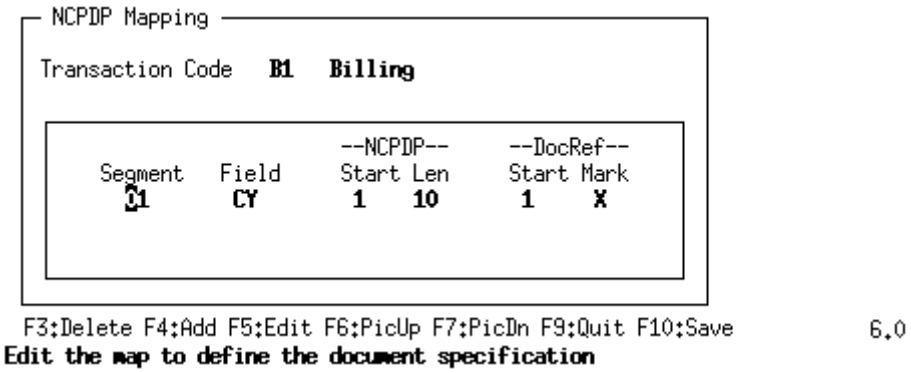
NCPDP Mapping Screen

Introduction

To modify an NCPDP transaction code map for a document specifier table, you use the NCPDP Mapping screen.

Illustration

This is the NCPDP Mapping screen.



NCPDP Mapping screen fields and functions

This table lists the fields on the NCPDP Mapping screen and their functions.

Field	Description
Transaction Code	Defines the NCPDP Transaction Code that represents the document or transaction. This value can be up to two-characters in length. You can use the F2 key to list available values. **Example** B1 for Billing
Segment	Defines the segment in the document that contains the unique value. This value can be up to two-characters in length. You can use the F2 key to list available values.
Field	The field ID of the field in the segment that contains a unique value.
NCPDP Start	The starting position (character number) of the character in the field that starts the unique value. This value can be any number from 1 to 99, inclusive.

(Contd) Field	Description
NCPDP Len	The length (in number of characters) of the unique value. This value can be any number from 1 to 40, inclusive. **Comments** ◗ The system uses the full length unless it encounters a separator or terminator character. ◗ The sum total length of the lines in the map cannot exceed 40.
DocRef Start	The starting place (character number) in the document reference number that this unique value occupies. This value can be any number from 1 to 40.
Mark	The symbol you want to use to mark this part of the document reference number in the map picture. You must use a different mark on each line. This is a single-character value. **Examples** %, &, *

NCPDP Mapping screen function keys

This table describes the function keys on the NCPDP Mapping screen and their functions.

Key	Function
F3	Deletes the selected segment and field from the map.
F4	Displays the Add screen so that you can add a line to the map.
F5	Displays the Edit screen for the selected line so that you can change the values in the line.
F6	Displays a pictorial representation of the map.
F7	Removes the pictorial representation from the display.
F9	Exits the screen.
F10	Saves your changes to the map.

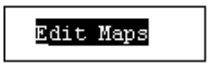
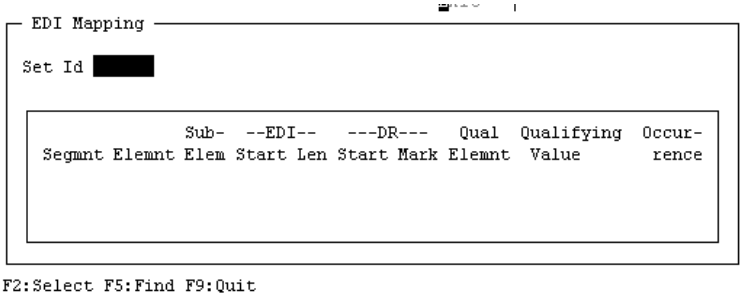
How to Display a Document Specifier Map

Introduction

You must display a document specifier map in the Mapping screen before you can delete it or modify it.

Procedure

Use this procedure to view a set ID, application name, XML name, or NCPDP transaction code map.

Step	Action
1	Display the document specifier table that has the map you want to display. **Reference** See the How to Display a Document Specifier Table topic in this chapter for instructions.
2	Select Edit Maps and press ENTER.  **System Response** Sterling Gentran:Server displays the Mapping screen for the table type. 

<table>
<tr><th>(Contd) Step</th><th>Action</th></tr>
<tr><td>3</td><td>Select the Set ID, Application Name, DDF Name, or Transaction Code field and then press F5 to display a list of names mapped to the current table.

CAUTION
To view a different map, press F10. The system clears the map name and fields and then prompts you for a new set map name.

<table>
<tr><th>IF you want to…</th><th>THEN…</th></tr>
<tr><td>To scroll the list</td><td>Use the cursor keys.</td></tr>
<tr><td>Display a map</td><td>Select the set ID, application name, DDF name, or NCPDP transaction code; then press ENTER.</td></tr>
</table>
System Response
The system redisplays the Mapping screen, filling in the set ID, application name, DDF name, or transaction code identifiers and the segments or records mapped to the name.

WARNING
You cannot edit the information on this screen. To change the values in the map, see the topic How to Change a Document Specifier Map in this chapter.</td></tr>
</table>

How to Change a Document Specifier Map

Introduction

This topic explains how to:

- Add segment or record lines to a document specifier map
- Change the values in a map line
- Delete lines from a map.

WARNING

If you change any part of a table, your change affects all the Trading Partnership code records attached to that table.

Adding a line to a map

Use this procedure to add a line to a map.

Step	Action
1	Display the map that you want to edit. **Reference** See How to Display a Document Specifier Map.
2	Press F4 to add a line to the map. **System Response** The system displays the appropriate Add screen. Add Segment Id Element # Sub-Elem # Elem Start from Length Doc Ref Start Visual Marker Qual Elem # Qual Value Occurrence F9:Quit F10:Save

(Contd) Step	Action
3	Complete the fields on the Add screen; then press F10 to save the line. **System Response** The system adds the segment or record line and then clears the screen so that you can add another segment or record line. **CAUTION** **On the APP Add screen only, the Record ID, Field Name, and Qual Fld Name fields display 12 characters at a time, but hold up to 128 characters. As you enter characters, the field scrolls to the right.**
4	After you have saved the last line you want to add, press F9 to exit.

Changing the values in a map line

Use this procedure to change a line in a map.

Step	Action
1	Display the map that you want to edit. **Reference** See How to Display a Document Specifier Map.
2	Select the line you want to edit and then press F5 to edit the line. **System Response** The system displays the Edit screen.

(Contd) Step	Action
3	Change the fields on the Edit screen; then press F10 to save your changes. **Note** You cannot change the values in the first three fields (Segment or Record ID, Element or Field #, and Sub-Elem#). **System Response** The system adds the segment or record line and then clears the screen so that you can add another segment or record line. **CAUTION** **On the APP Add screen, the Record ID, Field Name, and Qual Fld Name fields display 12 characters at a time, but hold up to 128 characters. As you enter characters, the field scrolls to the right.**
4	After you have saved the last line you want to add, press F9 to exit.

Deleting a line from a map

Use this procedure to delete a line from a document specifier map.

Step	Action
1	Display the map that you want to edit. **Reference** See How to Display a Document Specifier Map.
2	Select the line you want to delete and then press F3 to delete the line. **System Response** The system displays a confirmation prompt.
3	Type **y** at the prompt to confirm the deletion. **System Response** Sterling Gentran:Server deletes the line.
4	Press F10 to save your changes to the map.

How to Delete a Map from a Table

Introduction

To delete a document specifier map from a table, you must delete all the lines in the map. Once you have deleted all the map's lines, Sterling Gentran:Server removes the set ID, application name, XML name, or NCPDP transaction code map from the list associated with the document specifier table.

WARNING

If you change any part of a table, your change affects all the Trading Partnership/Table records attached to that table.

Deleting a map from a document specifier table

Use this procedure to delete a document specifier map from a table.

Step	Action
1	Display the map that you want to delete. **Reference** See How to Display a Document Specifier Map.
2	Select a line and then press F3 to delete the line. **System Response** The system displays a confirmation prompt. Confirmation Delete Segment: G47 Element: 2 Sub-Element: Are You Sure (y/n) ? _
3	Type **y** at the prompt to confirm the deletion. **System Response** Sterling Gentran:Server deletes the line.
4	Repeat Steps 2 and 3 until you have deleted all the lines in the map.
5	Press F10 to save your changes. **System Response** Sterling Gentran:Server removes the document specifier map from the document specifier table.

CHAPTER
8

Setting Up Life Cycle

Contents

Overview
- Introduction 2
- Database Software 4
- Life Cycle Components 6

Life Cycle Tables
- Life Cycle Table lc221 8
- Life Cycle Table: Informix 10
- Life Cycle Table: Oracle 14
- Life Cycle Table: Sybase 18

Life Cycle Configuration
- Overview 22
- Life Cycle Configuration Process 23
- How to Set Up Life Cycle for Oracle 26
- How to Set Up Life Cycle for Informix 32
- How to Set Up Life Cycle for Sybase 35
- How to Edit Data Manager Initialization Files 38

Testing Life Cycle Setup
- How to Test Outbound Processing 47
- How to Test Inbound Processing 49

Overview

Introduction

The Life Cycle utility

Data manager activity generates Life Cycle event records. These records chronicle the actions of data managers during processing. You use these event records to track data manager activity.

The programs in the Sterling Gentran:Server Life Cycle utility enable you to load Life Cycle event records to an auditing file, which is normally a relational database table. You can then use the records for auditing purposes.

In this chapter

This chapter explains how to configure your system to load Life Cycle event records to a relational database. It is divided into these sections:

- Overview, which describes the resources you need to set up Life Cycle and the components that comprise Life Cycle
- Life Cycle Configuration, which contains procedures to configure your system to use a Life Cycle database table
- Testing Life Cycle Setup, which explains how to test inbound and outbound Life Cycle Processing.

Required resources

You need the following resources in order to set up and run the Life Cycle utility:

- Relational database software for one of the supported databases
- A programmer who knows the database package

Key terms

This table lists the key terms used in this chapter.

Term	Description
auditing facility	The Sterling Gentran:Server facility that loads data manager event files to an auditing file, such as a relational database table, so that you can use the records for auditing purposes.
database	A collection of stored data often shared by different applications.

(Contd) Term	Description
EDI_AUDIT	The environment variable that sets the path to the database you use for your Life Cycle table.
event record	A record produced when a data manager processes a file. The record contains the date, time, name, and location of the data as it is passed through the data manager.
functional acknowledgment (FA)	The standard transaction set used to acknowledge receipt of a transmission.
lcld	The Sterling Gentran:Server program that loads new event records to the Life Cycle table. All data managers except line managers and archive managers create event records.
Life Cycle event file	The file that contains a data manager's Life Cycle event records. The name of the event file is the data manager's name with a .v suffix.
Life Cycle programs	The programs **lcld** and **xlld**, which load and update the Life Cycle table with data manager event files.
Life Cycle table	The database table that holds your audit file records. Your database administrator creates the table during the Life Cycle setup process and gives it public access.
mksrvdb	The program or script that creates the Life Cycle database table.
tracker	The Sterling Gentran:Server command line program that enables you to run a statistical report on the translation traffic.
translation audit files	The event files, *edistat.i* and *edistat.o*, that the translator produces. These files are also referred to as temporary audit files or status records.
xlld	The Sterling Gentran:Server program that updates the Life Cycle table records that **lcld** loaded with information from the translation audit files.

Database Software

Introduction

To take advantage of the Sterling Gentran:Server ability to pass its event records to a relational database, you must have in place database software purchased from a supported database vendor. You do not need database software to run Sterling Gentran:Server, but you must have it to view your Life Cycle event records from Sterling Gentran:Server.

Reference
See the installation and setup instructions provided with the database software for instructions on installing and configuring the relational database.

Supported databases

The Sterling Gentran:Server Life Cycle auditing facility interfaces with:

- Informix, versions 5.x, 6.0, 7.0 and higher
- Oracle, versions 7.1 and higher
- Sybase, version 11.03.

Required software

This table lists, by vendor, the software required to run Life Cycle.

Vendor	Required Software	Description
Informix	ESQL/C Openline	Embedded SQL precompiler
	I-SQL Openline	SQL statement interpreter
	Informix Standard Engine (SE)	Standard database engine
Oracle	PRO*C	Embedded SQL precompiler for C source code
	Oracle RDBMS	Database server
	PL/SQL	SQL statement interpreter
Sybase	Sybase SQL Server	Database server
	Open Client/C	Necessary libraries to communicate with Sybase
	Embedded SQL/C	Embedded SQL precompiler for C source code

Location of the database software

The database software may reside on:

- The same machine as Sterling Gentran:Server
  or
- Another machine on the same network as the UNIX host that contains Sterling Gentran:Server.

WARNING

We do not recommend running Life Cycle on a computer that is not on the same network as the UNIX host that contains Sterling Gentran:Server.

Life Cycle Components

Introduction

This section describes the major components of the Life Cycle utility.

Event records

Data managers produce **event records** of transaction activity. A Life Cycle event record describes where the data came from, where it went, and the date and time it happened. The event record does not contain the transaction data.

Life Cycle event file

Data managers store their event records in a **Life Cycle event file.** The name of the file is the data manager's name with either a *.v* suffix or a *.v.<uniqueID>* suffix. The suffix is determined by the value for the MULTIPLE_LIFE_CYCLE_FILES parameter in the data manager's initialization file.

Examples

```
dnld.v
dnld.v.839283
```

Life Cycle event file location

The Life Cycle event file is stored in the directory named in the LIFE_CYCLE_DIR parameter of the data manager's initialization file. The default directory is lcld.

Event file is loaded to database

When you configure your system to use a relational database to audit your Sterling Gentran:Server activity, Sterling Gentran:Server loads the Life Cycle event file to the Life Cycle database table. Each record in the file creates one Life Cycle database record.

Database table

The **Life Cycle table** is the database table to which the Life Cycle programs load your Life Cycle event file records. Your Life Cycle table must contain the same fields as the Life Cycle event records.

Note

You must create the Life Cycle table with your Informix, Oracle, or Sybase database software. The instructions in this chapter guide you through the process.

Reference

See the *File Record Layouts* chapter of the*IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide* for a description of the layout of the Life Cycle table.

Load program

The Life Cycle load program, **lcld**, loads event records to the Life Cycle database table.

Translation audit files

The Sterling Gentran:Server translation process generates two types of audit files: *edistat.i* (inbound translation) and *edistat.o (*outbound translation). Translation also produces *edistat.o* for inbound translations if translation creates functional acknowledgments. The Life Cycle process uses information in *edistat.i* and *edistat.o* to update event records in the Life Cycle table.

Reference
See the *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide* for more information about the audit files produced during translation.

Update program

The Life Cycle update program, **xlld**, uses information in the translation audit files to update records that **lcld** loaded to the Life Cycle table.

Life Cycle Tables

Life Cycle Table lc221

Introduction

The **Life Cycle table** is the primary destination database table for the data manager's Life Cycle event files and the translator's audit files. Sterling Gentran:Server supports Life Cycle tables for these databases:

- Oracle
- Informix
- Sybase.

Life Cycle table name

The Life Cycle table is named lc221.

How the Life Cycle table is created

Your organization creates the Life Cycle table as part of the Life Cycle setup process.

Reference
See the Setting Up Life Cycle chapter in this guide for information about creating the Life Cycle table.

Key fields

To avoid duplicate table entries, the Life Cycle programs **lcld** and **xlld** check the values in certain **key fields** of the Life Cycle table to identify entries. Each table entry has a unique value in at least one of the key fields to distinguish it from all other entries.

Set level
For set level entries, these are the four basic key fields:

- TP
- DOC
- IOX
- SEQ.

Group level

For group level entries, the programs check one additional field:

- GSCTL.

Functional acknowledgments

For functional acknowledgment entries, the programs check these fields in addition to the four basic key fields:

- MYISID
- TPISID
- MYGSID
- TPGSID
- GSCTL
- STSETID.

Sequence number

The sequence number (SEQ) field of the Life Cycle table contains a system-generated value. The **lcld** program generates the sequence number when it finds that the Life Cycle table already contains an entry with values matching the four basic key fields.

When **lcld** finds a match, it increments the sequence number by 1. This ensures that the new entry has a unique value in the SEQ field to distinguish it from other entries.

Life Cycle table modification

You can add columns and indexes to a Life Cycle table.

Reference

See your database documentation for instructions on modifying a database table.

Life Cycle Table: Informix

Introduction

This topic describes the Informix Life Cycle Table.

The lc221 table

This table describes the columns in the lc2211 Life Cycle table.

Name	Type	Description
TP	VARCHAR(31)	Trading Partnership identification code
DOC	VARCHAR(40)	Document reference number
IOX	VARCHAR(1)	Direction of data manager ◗ i = inbound ◗ o = outbound ◗ x = other
SEQ	VARCHAR(2)	Sequence number
ISUNIQ	VARCHAR(9)	Unique identifier assigned in the data manager
MYISID	VARCHAR(35)	Interchange sender code
MYGSID	VARCHAR(35)	Application/group sender code
TPISID	VARCHAR(35)	Interchange receiver code
TPGSID	VARCHAR(35)	Application receiver code
GSVERS	VARCHAR(15)	Version number
STSETID	VARCHAR(6)	Set identifier
PGM	VARCHAR(6)	Data manager name
RSLT	VARCHAR(1)	Data manager result code (0=Success)
DT	DATE	Data manager process date (CCYYMMDD)
TM	VARCHAR(9)	Data manager process time (HHMMSS)
SDIR	VARCHAR(60)	Source directory name (work directory)
SFIL	VARCHAR(60)	Source file name (drop-off name)

(Contd) Name	Type	Description
DDIR	VARCHAR(60)	Destination directory name
DFIL	VARCHAR(60)	Destination file name
ADIR	VARCHAR(60)	Archive directory
ISCTL	VARCHAR(15)	Interchange control number
GSCTL	VARCHAR(15)	Group control number
STCTL	VARCHAR(15)	Set control number
TDT	DATE	Translation date (CCYYMMDD)
TTM	VARCHAR(9)	Translation time (HHMMSS)
FADT	DATE	Functional acknowledgment date (CCYYMMDD)
FATM	VARCHAR(9)	Functional acknowledgment time (HHMMSS)
FAREQ	VARCHAR(1)	Functional acknowledgment request flag Values: ◗ A = Accept ◗ P = Partially accepted ◗ R = Reject ◗ E = Accepted with errors ◗ N = Inbound: no acknowledgment generated outbound: no acknowledgment expected ◗ Y = Inbound: acknowledgment to be generated, outbound: acknowledgment expected
SEGCNT	VARCHAR(9)	Segment count
CHARCNT	VARCHAR(9)	Character count
MAILGROUP	VARCHAR(9)	Mailbox operations

The LCDestInfo table

This table describes the columns in the LCDestInfo secondary Life Cycle table. This table joins the parent table based on the **tp**, **doc**, and **iox** fields.

Name	Type	Description
ActivitySysKey	VARCHAR(16)	Reserved for future use
tp	VARCHAR(36)	Trading Partnership code expanded
doc	VARCHAR(81)	Document Reference
IOX	VARCHAR(2)	Direction of data manager ◗ i = inbound ◗ o = outbound ◗ x = other
mailbagid	VARCHAR(10)	Run number (Mail bag ID)
DestType	VARCHAR(4)	Reserved for future use (Destination Type)
DestHost	VARCHAR(128)	Destination host name
DestID	VARCHAR(16)	Destination File Unique Key
DestUser	VARCHAR(128)	Reserved for future use (Destination User)
DestDir	VARCHAR(60)	Destination Directory Name
DestFile	VARCHAR(128)	Destination File Name
DestCharCnt	VARCHAR(16)	Destination Character Count
DestRecCnt	VARCHAR(16)	Destination Record Count
DestResult	VARCHAR(3)	Translation Result
DestContentType	VARCHAR(125)	File Type (Binary, EDI, etc.)
DestContentSubType	VARCHAR(125)	Reserved for future use
DestDesc	VARCHAR(80)	Description of subject

(Contd) Name	Type	Description
DestEncrypt	VARCHAR(2)	Reserved for future use (Encryption Flag)
DestCmp	VARCHAR(2)	Reserved for future use (Compression Flag)

Life Cycle Table: Oracle

Introduction

This topic describes the Oracle Life Cycle Table.

The lc221 table

This table describes the columns in the lc221 Life Cycle table.

Name	Type	Description
TP	VARCHAR2(31)	Trading Partnership code
DOC	VARCHAR2(40)	Document reference number
IOX	VARCHAR2(1)	Direction of data manager ◗ i = inbound ◗ o = outbound ◗ x = other
SEQ	VARCHAR2(2)	Sequence number
ISUNIQ	VARCHAR2(9)	Unique identifier assigned in the data manager
MYISID	VARCHAR2(35)	Interchange sender code
MYGSID	VARCHAR2(35)	Application/group sender code
TPISID	VARCHAR2(35)	Interchange receiver code
TPGSID	VARCHAR2(35)	Application receiver code
GSVERS	VARCHAR2(15)	Version number
STSETID	VARCHAR2(6)	Set identifier
PGM	VARCHAR2(6)	Data manager name
RSLT	VARCHAR2(3)	Data manager result code (0=Success)
DT	DATE	Data manager process date (CCYYMMDD)
TM	VARCHAR2(9)	Data manager process time (HHMMSS)
SDIR	VARCHAR2(60)	Source directory name (work directory)
SFIL	VARCHAR2(60)	Source file name (drop-off name)

(Contd) Name	Type	Description
DDIR	VARCHAR2(60)	Destination directory name
DFIL	VARCHAR2(60)	Destination file name
ADIR	VARCHAR2(60)	Archive directory
ISCTL	VARCHAR2(15)	Interchange control number
GSCTL	VARCHAR2(15)	Group control number
STCTL	VARCHAR2(15)	Set control number
TDT	DATE	Translation date (CCYYMMDD)
TTM	VARCHAR2(9)	Translation time (HHMMSS)
FADT	DATE	Functional acknowledgment date (CCYYMMDD)
FATM	VARCHAR2(9)	Functional acknowledgment time (HHMMSS)
FAREQ	VARCHAR2(1)	Functional acknowledgment request flag ◗ A = Accept ◗ P = Partially accepted ◗ R = Reject ◗ E = Accepted with errors ◗ N = inbound: No acknowledgment generated, outbound: No acknowledgment expected ◗ Y = inbound: Acknowledgment to be generated, outbound: Acknowledgment expected
SEGCNT	VARCHAR2(9)	Segment count
CHARCNT	VARCHAR2(9)	Character count
MAILGROUP	VARCHAR2(9)	Mailbox operations

The LCDestInfo table

This table describes the columns in the LCDestInfo secondary Life Cycle table. This table joins the parent table based on the **tp**, **doc**, and **iox** fields.

Name	Type	Description
ActivitySysKey	VARCHAR2(16)	Reserved for future use
tp	VARCHAR2(36)	Trading Partnership code expanded
doc	VARCHAR2(81)	Document Reference
IOX	VARCHAR2(2)	Direction of data manager ◗ i = inbound ◗ o = outbound ◗ x = other
mailbagid	VARCHAR2(10)	Run number (Mail bag ID)
DestType	VARCHAR2(4)	Reserved for future use (Destination Type)
DestHost	VARCHAR2(128)	Destination host name
DestID	VARCHAR2(16)	Destination File Unique Key
DestUser	VARCHAR2(128)	Reserved for future use (Destination User)
DestDir	VARCHAR2(60)	Destination Directory Name
DestFile	VARCHAR2(128)	Destination File Name
DestCharCnt	VARCHAR2(16)	Destination Character Count
DestRecCnt	VARCHAR2(16)	Destination Record Count
DestResult	VARCHAR2(3)	Translation Result
DestContentType	VARCHAR2(125)	File Type (Binary, EDI, etc.)
DestContentSubType	VARCHAR2(125)	Reserved for future use
DestDesc	VARCHAR2(80)	Description of subject

(Contd) Name	Type	Description
DestEncrypt	VARCHAR2(2)	Reserved for future use (Encryption Flag)
DestCmp	VARCHAR2(2)	Reserved for future use (Compression Flag)

Life Cycle Table: Sybase

Introduction

This topic describes the Sybase Life Cycle Table.

The lc221 table

This table describes the columns in the lc221 Life Cycle table.

Name	Type	Description
TP	CHAR(40)	Trading Partnership identification code
DOC	CHAR(40)	Document reference number
IOX	CHAR(1)	Direction of data manager ◗ i = inbound ◗ o = outbound ◗ x = other
SEQ	CHAR(2)	Sequence number
ISUNIQ	CHAR(9)	Unique identifier assigned in the data manager
MYISID	CHAR(35)	Interchange sender code
MYGSID	CHAR(35)	Application/group sender code
TPISID	CHAR(35)	Interchange receiver code
TPGSID	CHAR(35)	Application receiver code
GSVERS	CHAR(15)	Version number
STSETID	CHAR(6)	Set identifier
PGM	CHAR(6)	Data manager name
RSLT	CHAR(3)	Data manager result code (0=Success)
DT	DATE	Data manager process date (CCYYMMDD)
TM	CHAR(9)	Data manager process time (HHMMSS)
SDIR	CHAR(60)	Source directory name (work directory)
SFIL	CHAR(60)	Source file name (drop-off name)

(Contd) Name	Type	Description
DDIR	CHAR(60)	Destination directory name
DFIL	CHAR(60)	Destination file name
ADIR	CHAR(60)	Archive directory
ISCTL	CHAR(15)	Interchange control number
GSCTL	CHAR(15)	Group control number
STCTL	CHAR(15)	Set control number
TDT	DATE	Translation date (CCYYMMDD)
TTM	CHAR(9)	Translation time (HHMMSS)
FADT	DATE	Functional Acknowledgment Date (CCYYMMDD)
FATM	CHAR(9)	Functional Acknowledgment Time (HHMMSS)
FAREQ	CHAR(1)	Functional acknowledgment request flag ◗ A = accept ◗ P = Partially accepted ◗ R = Reject ◗ E = Accepted with errors ◗ N = inbound: No acknowledgment generated outbound: No acknowledgment expected ◗ Y = inbound: Acknowledgment to be generated, outbound: Acknowledgment expected
SEGCNT	CHAR(9)	Segment count
CHARCNT	CHAR(9)	Character count
MAILGROUP	CHAR(9)	Mailbox operations

The LCDestInfo table

This table describes the columns in the LCDestInfo secondary Life Cycle table. This table joins the parent table based on the **tp**, **doc**, and **iox** fields.

Name	Type	Description
ActivitySysKey	CHAR(16)	Reserved for future use
tp	CHAR(36)	Trading Partnership code expanded
doc	CHAR(81)	Document Reference
IOX	CHAR(2)	Direction of data manager ◗ i = inbound ◗ o = outbound ◗ x = other
mailbagid	CHAR(10)	Run number (Mail bag ID)
DestType	CHAR(4)	Reserved for future use (Destination Type)
DestHost	CHAR(128)	Destination host name
DestID	CHAR(16)	Destination File Unique Key
DestUser	CHAR(128)	Reserved for future use (Destination User)
DestDir	CHAR(60)	Destination Directory Name
DestFile	CHAR(128)	Destination File Name
DestCharCnt	CHAR(16)	Destination Character Count
DestRecCnt	CHAR(16)	Destination Record Count
DestResult	CHAR(3)	Translation Result
DestContentType	CHAR(125)	File Type (Binary, EDI, etc.)
DestContentSubType	CHAR(125)	Reserved for future use
DestDesc	CHAR(80)	Description of subject

(Contd) Name	Type	Description
DestEncrypt	CHAR(2)	Reserved for future use (Encryption Flag)
DestCmp	CHAR(2)	Reserved for future use (Compression Flag)

Life Cycle Configuration

Overview

In this section

This section contains instructions for setting up Life Cycle for each of the supported database products.

It begins with an overview of the Life Cycle configuration process.

Life Cycle Configuration Process

Introduction

The specific steps you perform to configure your system to use the Life Cycle utility depend upon the database product you are using for your Life Cycle records. However, the overall processes are similar for all supported databases.

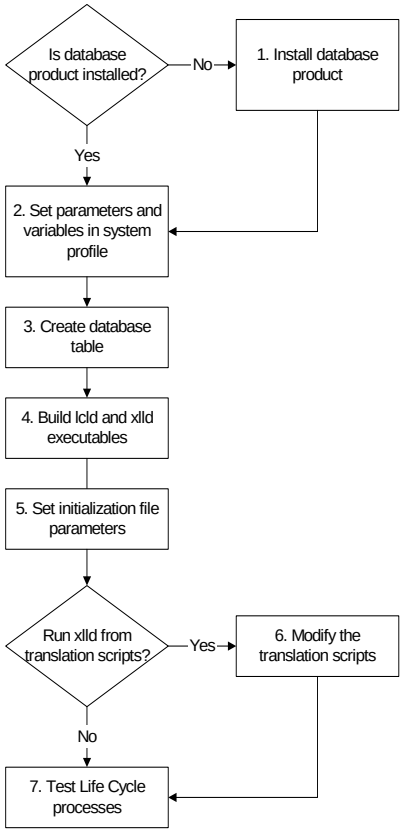
Reference

See the appropriate section in this chapter for specific procedures for the database product you use:

- How to Set Up Life Cycle for Oracle
- How to Set Up Life Cycle for Informix
- How to Set Up Life Cycle for Sybase.

Process flow diagram

This flow diagram shows the stages in the configuration process for the Life Cycle utility.



Task summary

This table summarizes the tasks you must complete to configure your system to load Life Cycle records to a database table. The table also provides references to information needed to perform each stage.

Task	Description
1	Is the database product installed? ◗ If YES, continue to Task 2. ◗ If NO, install the database product on the same machine as Sterling Gentran:Server or on another machine that is on the same network as the machine that has Sterling Gentran:Server. **Reference** See the installation instructions for your database product.
2	Configure the Sterling Gentran:Server system profile for database paths and environment variables. **Reference** See the appropriate topic in this chapter for instructions: ◗ How to Set Up Life Cycle for Oracle ◗ How to Set Up Life Cycle for Informix ◗ How to Set Up Life Cycle for Sybase
3	Create the Life Cycle database table. Grant write and update access privileges for the database table to any user who has permission to activate Sterling Gentran:Server data managers. Create a unique table index. **Reference** See the appropriate topic in this chapter for instructions: ◗ How to Set Up Life Cycle for Oracle ◗ How to Set Up Life Cycle for Informix ◗ How to Set Up Life Cycle for Sybase
4	Build the Life Cycle load and update programs, **lcld** and **xlld**. **Reference** See the appropriate topic in this chapter for instructions. ◗ How to Set Up Life Cycle for Oracle ◗ How to Set Up Life Cycle for Informix ◗ How to Set Up Life Cycle for Sybase

(Contd) Task	Description
5	Edit the initialization file of each data manager (agent) in your flows to set the Life Cycle parameters. **Reference** See the How to Edit Data Manager Initialization Files topic in this chapter for instructions.
6	Do you want to run **xlld** from a translation script? ◗ If YES, modify the translation script to run **xlld**. Make the **xlld** command run AFTER the **prep_xlldfile** command and AFTER translation has occurred. We recommend that you run **xlld** after **ediarc**. Also, make sure that the path to the translation audit file is correct in the script. ◗ If NO, continue with Task 7. **Reference** See the Working with Scripts chapter in this guide for information about modifying scripts. **Note** When choosing when to run **xlld**, keep in mind that **xlld** cannot access and update a Life Cycle record if the **lcld** process has not finished. You will get an SQL error if **xlld** cannot locate the complete record. This problem most often affects sites that process large amounts of data.
7	Test the Life Cycle processes. **Reference** See the Testing Life Cycle Setup section in this chapter for instructions.

How to Set Up Life Cycle for Oracle

Introduction

This topic describes how to configure your system to use an Oracle Life Cycle table.

Before you begin

Use this table as a checklist to make sure that you are ready to begin setting up an Oracle Life Cycle table.

Done	Task
	Install Oracle. **WARNING** **Be sure that your Oracle software includes the developer's kit that contains Oracle's C compiler, Pro*C/C++. Depending on which revision of Oracle you purchased, the developer's kit is called either Programmer or Programmer 2000.**
	Install **Pro*C/C++** into your Oracle *bin* directory. **Reference** See your Oracle installation documentation for instructions.
	If you use Sterling Gentran:Server on the RS6000 with AIX 4.x, make sure you have a minimum of revision 6.2 of **xlld.pc**. **How to check your revision number** To check your revision number, change to the *$EDI_ROOT/src/oracle* directory and enter this command: `what xlld.pc` If you do not have a minimum of revision 6.2, contact IBM customer support.

(Contd) Done	Task
	If you use Sterling Gentran:Server on the HP-UX and are using Oracle version 7.3.3 or 7.3.4, then the HP UNIX ANSI C compiler must be version A.10.32.11 or higher. **Note** Sterling Gentran:Server uses both the UNIX ANSI C compiler and the **Pro*C** compiler to build **xlld**. **How to check your version number** 1. To find the UNIX ANSI C compiler, enter this command: **which cc** 2. Change directories to the directory displayed by the previous command. 3. Enter the following command: **what cc** **System response** The system displays the version number.
	Check the path to the database linker program. **How to check the linker** Enter the following command: **which ld** If the resulting path is */usr/uscb* or */usr/uscb/bin*, then you need to edit the path so that the following path appears first: */usr/ccs/bin*

Procedure

Use this procedure to configure your system to use an Oracle Life Cycle table.

Step	Action
1	Configure the Sterling Gentran:Server system profile to contain the following: ◗ The path to Oracle *bin* directory ◗ The Oracle environment variables. ◗ Set ORACLE_HOME. This is mandatory. ◗ Set ORACLE_SID if the Oracle database is on a different UNIX machine than Sterling Gentran:Server. ◗ You may need to set ORAENV_ASK, TWO_TASKS, or ORACLE_LPPROG, depending on your Oracle environment. ◗ We recommend that you put the Oracle libraries in the shared library path: SHLIB_PATH (HP), LIBPATH (AIX) or LD_LIBRARY_PATH (Solaris). **Reference** Refer to your Oracle documentation, or ask your database administrator for the correct values for the environment variables.
2	Add the environment variables from Step 1 to the profile of the owner of the Sterling Gentran:Server Foreground Manager. **Comment** This step ensures that the database user has access to both the Sterling Gentran:Server environment and the database environment.
3	Make sure that the Oracle binaries are in $PATH.

(Contd) Step	Action
4	Do you want to use the default configuration? ◗ If YES, — Grant connection to one instance of the database table for both the Sterling Gentran:Server login ID and the Sterling Gentran:Server administrator ID. **Example** Grant connection privileges to OPS$<edi_server>, identified by password, where <edi_server> is the Server account name. This creates a link from the UNIX account to the database account. **Reference** See your Oracle documentation for information about passwords. — Grant resource privileges to the Sterling Gentran:Server login ID (for example, grant resource to OPS$<edi_server>, where <edi_server> is the Server account name). — Continue with Step 3. ◗ If NO, GO TO to Step 6.
5	Log in as the administrative user you created to own security. **Reference** See the *How to Install Security Administration Software* topic in the *Installing Sterling Gentran:Server On the UNIX Host* chapter in the *IBM® Sterling Gentran:Server® for UNIX Getting Started Guide*.

6	Execute the following command to create the Life Cycle database table:
	`sqlplus / @./src/oracle/mksrvdb221`
	System Response
	The system displays these messages:
	`Table created`
	`Index created`
	`Grant succeeded`
	Tip
	If you want only one record per document, create a unique key using the fields **tp**, **doc**, **iox**, and **seq** as the key.

7	If you did not define the database account, or if the last step resulted in errors, execute the following command: `sqlplus userid/password @./src/oracle/mksrvdb221`

(Contd) Step	Action
8	Copy the executable *$EDI_ROOT/bin/load* to *$EDI_ROOT/bin/lcld*.
9	Ensure that the shell scripts **lcld.sh** and **lcld.ctl** are copied to the EDI_ROOT/bin directory and that **lcld.sh** has execute permissions. **WARNING** **Please use the attached lcld.sh and lcld.ctl examples. They are designed to eliminate a potential concurrency problem that could cause files to be overwritten.**
10	Copy the Oracle **make** file from the source file in the Oracle environment into *$EDI_ROOT/src/oracle*. **Note** For Oracle versions 7.x, the **make** file is named *proc.mk*. For Oracle versions 8.0 or higher, the **make** file is named *demo_proc.mk*. **CAUTION** **The exact source location of this file depends upon the Oracle version and release. Check your Oracle documentation or ask your database administrator.**
11	Change to the *$EDI_ROOT/src/oracle* directory.
12	Do you have ORACLE version 7.3 or higher? ◗ If YES: — Open the *proc.mk* or *demo_proc.mk* file. — Search for the string “SAMPLES=sample1” and replace it with the string “SAMPLES=xlld”. (Do not type the quotes.) ◗ If NO, continue with Step 13.

<table>
<tr><th>(Contd) Step</th><th colspan="2">Action</th></tr>
<tr><td>13</td><td colspan="2">Execute the command to create the translation load and FA update program, xlld.

Note
xlld has optional parameters U and P:
◗ -U <Oracle userID>
◗ -P <Oracle password></td></tr>
<tr><td></td><td>IF you have this version of ORACLE…</td><td>THEN use this command, entered on one line…</td></tr>
<tr><td></td><td>7.1 or 7.2</td><td>make -f proc.mk USERID=user/password objs=xlld.o exe=xlld</td></tr>
<tr><td></td><td>7.3 through 7.3.2</td><td>make -f proc.mk xlld</td></tr>
<tr><td></td><td>7.3.3 or 7.3.4</td><td>make -f proc.mk xlld LLIBXA=</td></tr>
<tr><td></td><td>8.0 or higher</td><td>make -f demo_proc.mk SAMPLES</td></tr>
<tr><td>14</td><td colspan="2">Move the xlld executable to the $EDI_ROOT/bin directory.</td></tr>
<tr><td>15</td><td colspan="2">Set the Life Cycle parameters in the data manager’s initialization file.

Reference
See the How to Edit Data Manager Initialization Files topic in this chapter for instructions.</td></tr>
<tr><td>16</td><td colspan="2">Modify the translation script if you want to run xlld from the translation script.

Reference
See the Working with Scripts chapter in this guide for information about modifying scripts.</td></tr>
</table>

How to Set Up Life Cycle for Informix

Introduction

This topic describes how to configure your system to use an Informix Life Cycle table.

Before you begin

You must have the embedded sql compiler **esqlc** properly installed before you use these instructions. Make sure this executable is in your Informix *bin* directory.

Procedure

Use this procedure to configure your system to use an Informix Life Cycle table.

Step	Action
1	Install Informix. **Reference** See your Informix installation documentation for additional information.
2	Configure the Sterling Gentran:Server system profile to contain the following: ◗ Path to the Informix *bin* directory. ◗ Informix environment variables TERMCAP and INFORMIXDIR. ◗ Set SHLIB_PATH (HP), LIBPATH (AIX) or LD_LIBRARY_PATH (Solaris) to the path to shared database libraries. ◗ Other environment variables that your database administrator identifies.
3	Add the environment variables from Step 2 to the profile of the owner of the Sterling Gentran:Server Foreground Manager.
4	Make sure that the Informix binaries are in $PATH.

<table>
<tr><th>(Contd) Step</th><th>Action</th></tr>
<tr><td>5</td><td>Change to the $EDI_ROOT/src/infx subdirectory and execute the make command, makeinfx.

CAUTION
The makeinfx command located in the infx directory is configured to build xlld, lcld and mksrvdb.

Example
make -f makeinfx

Tip
If you want only one record per document in the Life Cycle table, create a unique key using the fields tp, doc, iox, and seq as the key.</td></tr>
<tr><td>6</td><td>After you have successfully built xlld, lcld, and mksrvdb, move the resulting executable files to $EDI_ROOT/bin.</td></tr>
<tr><td>7</td><td>Log in as the administrative user you created to own security.

Reference
See the How to Install Security Administration Software topic in the Installing Sterling Gentran:Server On the UNIX Host chapter in the IBM® Sterling Gentran:Server® for UNIX Getting Started Guide.</td></tr>
<tr><td>8</td><td>Use this table to determine the appropriate command.
<table>
<tr><th>IF your Informix version is…</th><th>THEN run…</th></tr>
<tr><td>6.0 and above, openview</td><td>mksrvdb6</td></tr>
<tr><td>5.x, 6.0, 7.0 standard engine</td><td>mksrvdb4</td></tr>
</table>
System Response
The system displays the following messages:

Create Database: edisrv2, sqlcode=0

Create Table: edisrv2: lc221, sqlcode=0

Finished</td></tr>
</table>

(Contd) Step	Action
9	Set the Life Cycle parameters in the data manager's initialization file. **Reference** See the How to Edit Data Manager Initialization Files topic in this chapter for instructions.
10	Modify the translation script if you want to run **xlld** from the translation script. **Reference** See the Working with Scripts chapter in this guide for information about modifying scripts.

How to Set Up Life Cycle for Sybase

Introduction

This topic describes how to configure your system to use a Sybase Life Cycle table.

WARNING

The procedure in this topic is for Sybase versions 11.0 through 11.03. If you have another version, the procedure and location of files may be different. See your database administrator for assistance.

Before you begin

You must have the compiler **cpre** properly installed before you use these instructions. Make sure this executable is in your Sybase *bin* directory.

Procedure

Use this procedure to configure your system to use a Sybase Life Cycle table.

Step	Action
1	Install Sybase. **Reference** See your Sybase installation documentation.

2	Configure the Sterling Gentran:Server system profile to contain the following:
	◗ Path to Sybase *bin*
	◗ Sybase environment variables DSQUERY, SYBASE, SYBROOT, and SYBPLATFORM
	◗ Set SHLIB_PATH (HP), LIBPATH (AIX) or LD_LIBRARY_PATH (Solaris) to the path to shared database libraries.
	
	Note
	The valid values for SYBPLATFORM are:
	
	`sun_svr4 (Sun)`
	`dce_sun_svr4 (Sun using DCE)`
	`rs6000 (AIX)`
	`dce_rs6000 (AIX Thread Safe)`
	`dce_hp800 (HP Thread Safe)`

3	Add the environment variables from Step 2 to the profile of the owner of the Sterling Gentran:Server Foreground Manager.
4	Make sure that the Sybase binaries are in $PATH.

(Contd) Step	Action
5	Change to the *$EDI_ROOT/src/sybase* subdirectory.
6	Copy these files from *$SYBASE/sample/esqlc/* to *$EDI_ROOT/src/sybase*: ◗ *makefile* ◗ *sybsqlex.h* ◗ *sybopts.sh*
7	Globally change “sample 1” to “lcld” and “sample 2” to “xlld” in *makefile*. **Note** Do not type the quotes.
8	Build the **lcld** and **xlld** executable files by running the **make** command. **Example** `make makefile lcld, make makefile xlld`
9	Change “lcld” to “mksrvdb221” and run the **make** command again to build the database table. **Tip** If you want only one record per document in the Life Cycle table, create a unique key using the fields **tp**, **doc**, **iox**, and **seq** as the key.
10	After you have successfully built **xlld**, **lcld**, and **mksrvdb221**, move the executable files to *$EDI_ROOT/bin*.
11	Log in as the administrative user you created to own security. **Reference** See the *How to Install Security Administration Software* topic in the *Installing Sterling Gentran:Server On the UNIX Host* chapter in the *IBM® Sterling Gentran:Server® for UNIX Getting Started Guide*.
12	Execute the following command: `mksrvdb221 -U<userid> -P<password>`

(Contd) Step	Action
13	Set the Life Cycle parameters in the data manager's initialization file. **Reference** See the How to Edit Data Manager Initialization Files topic in this chapter for instructions.
14	Modify the translation script if you want to run **xlld** from the translation script. **Reference** See the Working with Scripts chapter in this guide for information about modifying scripts.

How to Edit Data Manager Initialization Files

Introduction

A data manager's **initialization file** is the text file that contains the data manager's personality and processing parameters. Every data manager in a flow (source agent, processing agent, delivery agent) has an initialization file.

Initialization file names

The name of a data manager's initialization file is the data manager's name followed by the suffix *.init.*

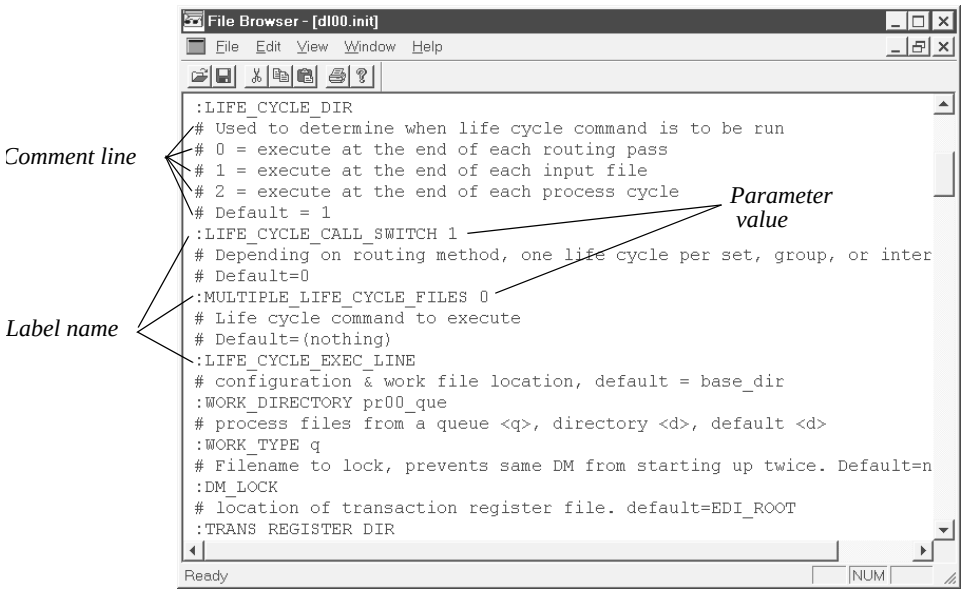
Example
If the source agent's name is **ap01**, its initialization file is named ***ap01.init***.

Location

Initialization files are stored in the *$EDI_ROOT/conf.d* directory.

Initialization file components

This illustration shows part of an initialization file and its components.



This table describes the components.

Part	Description
# comment line	A line of text that explains the parameter or contains a parameter option. A parameter's comments precede the parameter's label name.
:label name	The name of the parameter. **Important** Never change the label name.
parameter value	The value of the parameter. **Note** For most parameters, you enter the parameter value on the same line as the parameter's label name. The :LIFE_CYCLE_EXEC_LINE parameter is an exception. You must enter the value for this parameter on the line that follows the label name.

Using Life Cycle in a flow

To use Life Cycle in a flow, you must edit the initialization file of each data manager in the flow.

The parameters in the Routing section of an initialization file control the Life Cycle usage for the data manager.

Before you begin

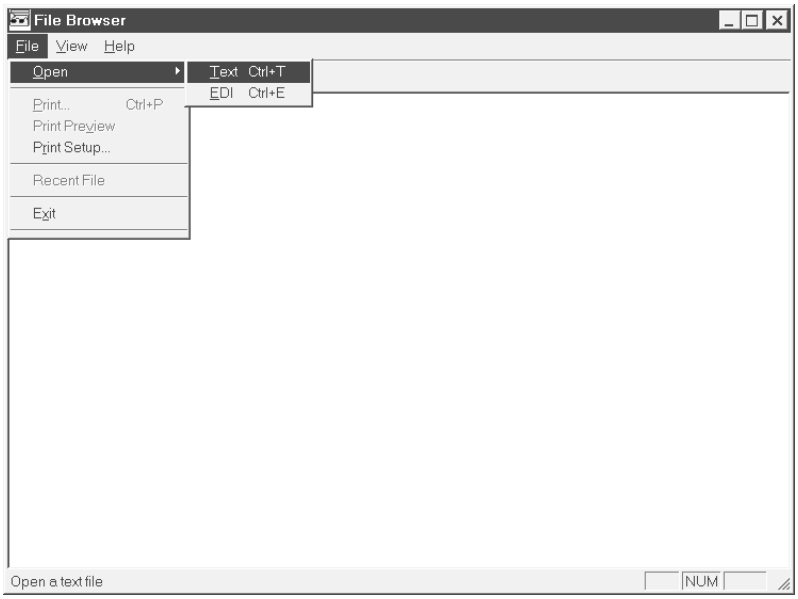
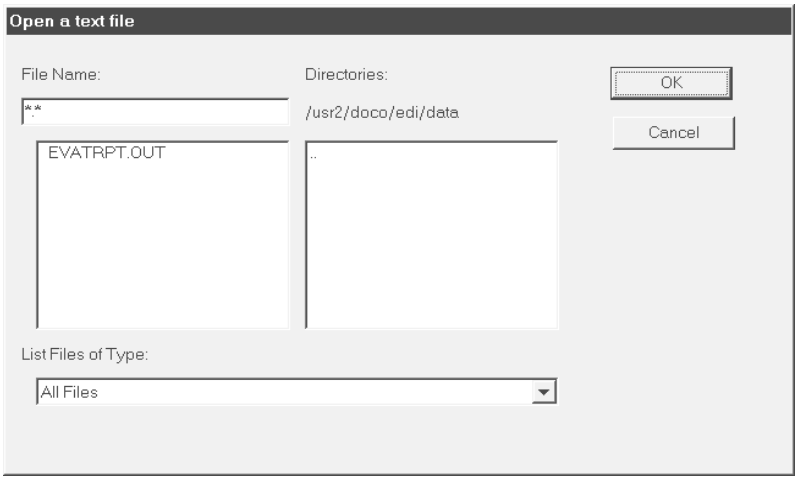
Check the flow summary tree to find the names of the data managers in the flow to which you are adding Life Cycle. You must know the names to locate and edit their initialization files.

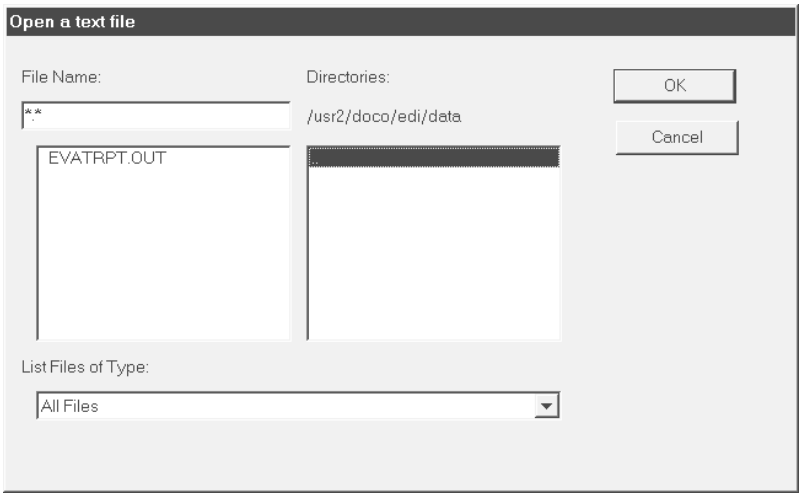
Reference
See the Using Flow Summaries section in the Creating a Flow with the PCM Wizard chapter of this guide for instructions.

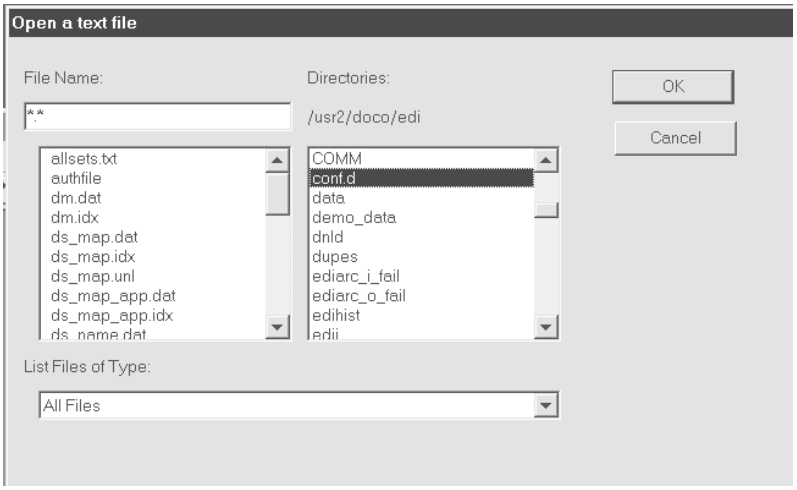
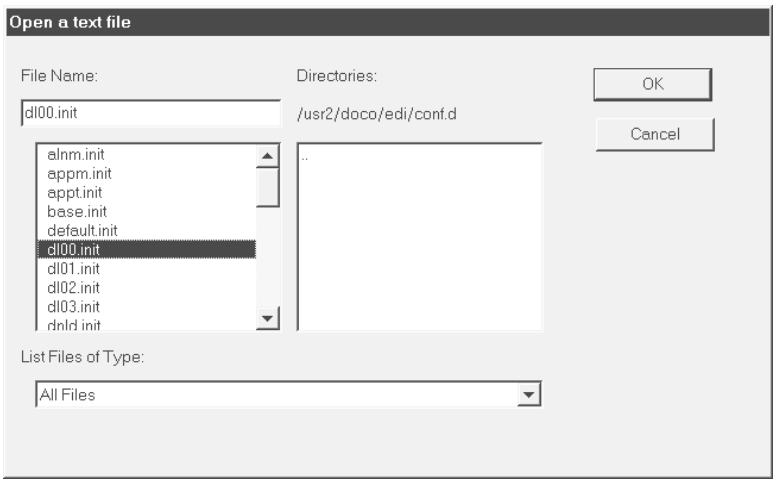
Editing an initialization file

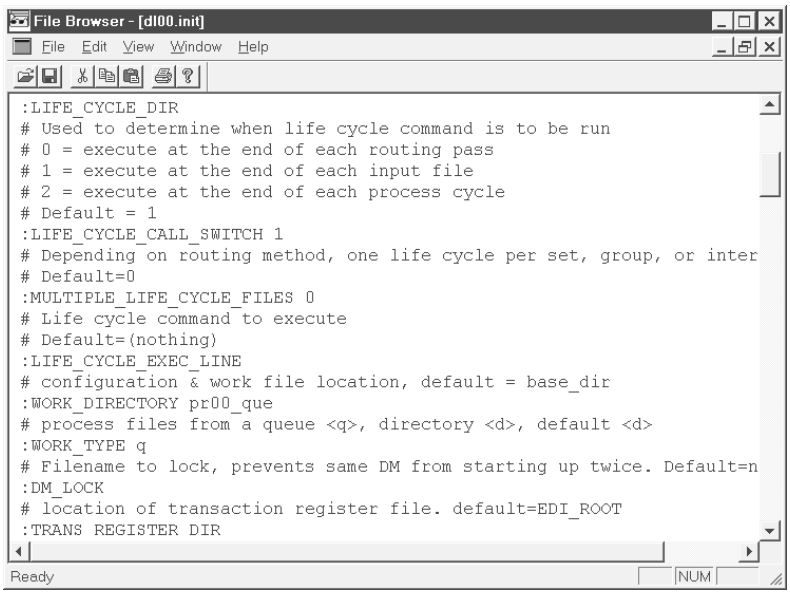
Use this procedure to open and edit a data manager's initialization file.

Step	Action
1	Click **File Browser** on the Tools menu to open the File Browser window.

(Contd) Step	Action
2	Point to **Open** on the File menu and click **Text**.  **System Response** The File Browser displays the Open a text file dialog box. 

(Contd) Step	Action
3	Double-click on the parent directory indicator (..) in the right list box to display the directories under EDI_ROOT.  **System Response** The File Browser displays a list of directories in the left list box.

(Contd) Step	Action
4	Double-click on the ***conf.d*** directory.  **System Response** The File Browser displays a list of initialization files in the *conf.d* directory. **Note** Data manager initialization files have the suffix *"init"* (*<dmname>.init*).
5	Click the initialization file you want to edit. 
6	Click **OK**. **System Response** The File Browser displays the file in a text editor window.

(Contd) Step	Action
7	Edit the Life Cycle parameters.  **Reference** For a list of parameters, the value options, and the editing guidelines, see the Initialization file parameters table that follows this procedure.
8	Click **Save** on the File menu to save your changes.
9	Repeat Steps 1 through 8 for each data manager in the flow.
10	Click **Exit** on the File menu to quit the File Browser.

Initialization file parameters

This table lists the initialization file parameters that you must set for the data manager.

Parameter	Function
LIFE_CYCLE_DIR	Defines the full path name of the directory in which the Life Cycle event files are placed for processing. The default directory is **lcld**. You must enter the value on the same line as the :LIFE_CYCLE_DIR label. **Example** :LIFE_CYCLE_DIR $EDI_ROOT/bin/xltr2
MULTIPLE_LIFE_CYCLE_FILES	Determines whether each set, group, or interchange has its own Life Cycle file. ◗ 1 = Yes ◗ 0 = No You must enter the value on the same line as the :MULTIPLE_LIFE_CYCLE_FILES label name. **Comment** If you set this parameter to 1 (the default), **lcld** creates a unique Life Cycle file for every event, so each event file contains one event record. This makes it easy to collect and move event files on a scheduled basis. **Example** :MULTIPLE_LIFE_CYCLE_FILES 0

(Contd) Parameter	Function
LIFE_CYCLE_CALL_SWITCH	Determines the point at which the Life Cycle command is run. ◗ 0 = Execute at the end of each routing pass ◗ 1 = Execute at the end of each input file ◗ 2 = Execute at the end of each process cycle You must enter the value on the same line as the :LIFE_CYCLE_CALL_SWITCH label name. **Example** :LIFE_CYCLE_CALL_SWITCH 1
LIFE_CYCLE_EXEC_LINE	Defines the Life Cycle command to be executed. You must enter the value on the line that follows the :LIFE_CYCLE_EXEC_LINE label name. **Example** :LIFE_CYCLE_EXEC_LINE lcld -f<dmnm> -U<user> -P<password> **WARNING** **If you omit this parameter, Sterling Gentran:Server does not call the Life Cycle load program.**

(Contd) Parameter	Function
USE_RECON_IDS	Determines which values are entered in Life Cycle event records: ◗ The extracted values ◗ The alternate reconciliation IDs you entered in the Inbound Acknowledgment tab of the Trading Partnership dialog box. **When to use** Use this option only for the source agent in an outbound application flow. Use this option to handle reconciliation when the IDs on the inbound functional acknowledgments that your trading partner sends you differ from the IDs on the outbound document you send. **Values** ◗ 1 = Yes ◗ 0 = No The default is 0. **Reference** See the *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide* for more information about reconciliation IDs.

Testing Life Cycle Setup

How to Test Outbound Processing

Introduction

This topic explains how to test the Life Cycle facility for outbound processing

Before you begin

Before you test the Life Cycle facility for outbound processing, you need to:

- Create the Life Cycle table
- Configure the outbound application process flow
- Configure the translator.

Procedure

Use this procedure to test outbound application processing.

Step	Action
1	Pass an application file through the source agent (application data manager). **WARNING** **Make sure the business application file contains a unique document reference number which can be referred to during translation. Make sure that you have properly defined the document reference number with the Document Specifier utility.**
2	Check the source agent's log file in the Life Cycle directory to make sure that **lcld.sh** has added the transaction set records to the Life Cycle database.
3	Run **lftran** from the command line to translate the outbound file.
4	Copy the *edistat.o* file to *xlto.v*.
5	Run the Life Cycle update program **xlld** from the command line. **Example** xlld -fxlto

(Contd) Step	Action
6	Check the *xlld.l* log file to make sure that **xlld** has updated the Life Cycle records with the translation date/time, control numbers, and Set ID.
7	Check the Life Cycle database table to make sure that **xlld** has updated the records.

How to Test Inbound Processing

Introduction

This topic explains how to test the Life Cycle facility for inbound processing

Before you begin

Before you test the Life Cycle facility for inbound processing, you need to:

- Create the Life Cycle table
- Configure the inbound process flow
- Configure the translator.

Note
If you are using an Oracle database, you must also make sure that lcld.sh is configured for a minimum of read/execute (-rwxr-x----) permissions.

Procedure

Use this procedure to test inbound processing.

Step	Action
1	Pass an EDI file through the source agent (inbound data manager) in the inbound process flow. **WARNING** **Make sure the EDI file used during inbound processing matches the EDI file layout used to create the translation map table.**
2	Check the source agent's log file in the Life Cycle directory to make sure that **lcld.sh** has added the transaction set records to the Life Cycle database.
3	Run lftran from the command line to translate the inbound file.
4	Copy the *edistat.i* file to *xlti.v*.

(Contd) Step	Action
5	Run the **xlld** command from the command line. **Example** xlld -fxlti This updates the Life Cycle records from the source agent with the translation date/time, control numbers, Set ID, and acceptance status. **CAUTION** **If you pass an inbound 997 file that matches the outbound Life Cycle data through the translator, the xlld program updates the Life Cycle records in the same way as the Sterling Gentran:Server archive program (ediarc) updates its archive file.**
6	Check the Life Cycle database table to make sure that **xlld** has updated the records.

CHAPTER
9

Tracking Data with Life Cycle Files

Contents

Overview
- Introduction 2
- Understanding Life Cycle Records 4
- The Life Cycle Facility 6
- The Life Cycle Table 7

The Life Cycle Load Programs
- Overview 8
- The lcld and xlld Life Cycle Process 10
- The lcld Program 12
- How to Run lcld from the Command Line 14
- The xlld Program 15
- How xlld Updates the Life Cycle Table 16
- How xlld Handles Functional Acknowledgments 18
- How xlld Handles Update Failures 21
- The xlld Log File 22
- How to Run xlld from the Command Line 23

Life Cycle Reports
- Overview 24
- The Translation Activity Report 26
- The FA Exception Report 28
- The FA Due Report 30
- The TP List Report 32
- The Translation Traffic Report 34

Running Life Cycle Reports
- Overview 37
- How to Run Informix Life Cycle Reports 39
- How to Run Oracle Life Cycle Reports 40
- How to Run the Translation Traffic Report 41

Overview

Introduction

In this chapter

This chapter describes the Life Cycle facility. The chapter explains:

- The Life Cycle event records that the data managers produce and the Life Cycle audit files that the translator produces
- The Life Cycle table
- How the Life Cycle load programs load the Life Cycle records to a database Life Cycle table
- How to configure your system to access the Life Cycle table from Sterling Gentran:Server
- How to run the Life Cycle reports.

Reference
For instruction on setting up your system to use the Life Cycle facility, see the Setting Up Life Cycle chapter in this guide.

Key terms

This table lists the key terms used in this chapter.

Term	Description
database	A collection of stored data often shared by different applications.
functional acknowledgment (FA)	The standard transaction set used to acknowledge receipt of a transmission.
key fields	The set of fields that the Life Cycle load programs use to identify entries in a Life Cycle table.
lcld	The Sterling Gentran:Server executable program that loads new Life Cycle event files to the Life Cycle database table.
lcld.sh	The Sterling Gentran:Server shell script that loads data manager's Life Cycle event records to the Life Cycle table.

(Contd) Term	Description
Life Cycle event file	The file that contains a data manager's Life Cycle event records. The name of the Life Cycle event file is the data manager's name with a ".v" suffix. A unique ID or a date and time stamp may follow the suffix in some cases.
Life Cycle event record	A record produced when a data manager processes a file. The record contains the date, time, name, and location of the data as it is passed through the data manager.
Life Cycle facility	The Sterling Gentran:Server facility that loads data manager Life Cycle event files to an auditing file, such as a relational database table, so that you can use the records for auditing purposes.
Life Cycle load programs	The programs **lcld** and **xlld.** These programs load and update the Life Cycle table with data manager Life Cycle event files.
Life Cycle table	The database table that holds your audit file records. Your EDI or database administrator creates this the table during the Life Cycle setup process and gives it public access.
mksrvdb	The program or script that creates the database Life Cycle table.
tracker	The Sterling Gentran:Server command line program that enables you to run a statistical report on the translation traffic.
translation audit files	The translation audit files, *edistat.i, edistat.o*, and *dbaudit.i* that the translator produces. These files are also referred to as temporary audit files or status records.
xlld	The Sterling Gentran:Server program that updates the Life Cycle table with information from translation audit files, including functional acknowledgments. The **xlld** program updates Life Cycle event records that already exist.

Understanding Life Cycle Records

Introduction

Sterling Gentran:Server processes generate two types of Life Cycle records:

- Life Cycle event records
- Translation audit records.

If you use the Sterling Gentran:Server Life Cycle facility, the Life Cycle programs load Life Cycle event records and translation audit records to an auditing file, which is normally a relational database table. You can then use the records for auditing purposes.

Life Cycle event records

When a data manager processes a file, Sterling Gentran:Server creates a Life Cycle event record. A Life Cycle event record stores the date, time, name and location of the data as it is passed through the data manager. All data managers except archive data managers (and line managers if you have IBM® Sterling Gentran:Server® for UNIX with ADD) produce Life Cycle event records.

Contents of Life Cycle event records

Life Cycle event records describe:

- Where the data came from (directory and file name)
- Whether the data passed the data manager's syntax checks
- Where the data went (directory and file name)
- The date and time the data arrived
- If and where the data was archived.

The Life Cycle event record does not contain the data itself; it just records where the data came from, where it went, and what time it happened.

Document reference number

To identify the document, a Life Cycle event record includes the document reference number. This is the number used to track the life of a document as it passes through Sterling Gentran:Server.

Where event records are stored

The Life Cycle event records are stored in one or more data manager Life Cycle event files. The LIFE_CYCLE_DIR parameter of a data manager’s initialization file sets the directory for its Life Cycle event file or files. The default directory name is **lcld**.

Translation audit files

The Sterling Gentran:Server translation process can generate three types of translation audit files:

- *edistat.i* (inbound translation)
- *edistat.o* (outbound translation). Translation also produces *edistat.o* for inbound translations if functional acknowledgments were created during translation.
- *dbaudit.i* (inbound database translation).

Reference
See the *IBM® Sterling Gentran:Server® for UNIX Application Integration User Guide* for more information about the audit files produced during translation.

Inbound database translation event records

You can configure the translator, **Iftran**, to generate audit records for inbound database translations. The translator generates one audit record per document per database table.

The Life Cycle Facility

Introduction

The Sterling Gentran:Server Life Cycle facility enables you to load Life Cycle event files and translation audit file information to an auditing file. The auditing file is normally a relational database table, referred to as the Life Cycle table.

Use of database software

To take advantage of Sterling Gentran:Server's ability to pass the Life Cycle event file to a supported relational database, you must have the database software and compiler installed on your system. You must purchase this software from the database vendor.

CAUTION

You do not need database software to run Sterling Gentran:Server, but you must have it to use the Life Cycle feature.

Reference

See the Setting Up Life Cycle chapter in this guide for information about configuring Life Cycle to work with a supported relational database.

If you do not use a database

Using a relational database with Sterling Gentran:Server is optional. You can load event data to a non-database file. However, you must develop your own utilities to load event data and generate reports.

The Life Cycle Table

Introduction

The Life Cycle table is the destination table for the Life Cycle event files and update information from the translation audit files. The Life Cycle load programs enter the Life Cycle data into the Life Cycle table.

Reference
See the Life Cycle Tables section in the Setting Up Life Cycle chapter in this guide for the structure of the Life Cycle table.

Purpose of the table

The Life Cycle table enables you to:

- Monitor the life of data as it passes through Sterling Gentran:Server
- Generate reports
- Use other features of the database to track your data.

How the table was created

When your EDI administrator set up your system to use a relational database with Sterling Gentran:Server, the administrator used a Sterling Gentran:Server script or program named **mksrvdb** to create a database table called the Life Cycle table.

Caution

The version of mksrvdb used to create the table depends on the database you use for Life Cycle data. For example, the Oracle Life Cycle table is created by the *mksrvdb221.sql* SQL script, and the Informix Life Cycle table is created by the mksrvdb ESQL-C program.

Reference
See the Setting Up Life Cycle chapter in this guide for information about configuring your system to use a relational database with the Sterling Gentran:Server Life Cycle facility.

The Life Cycle Load Programs

Overview

Introduction

The Life Cycle load programs load Life Cycle event files to the Life Cycle table. There are two Life Cycle load programs:

- lcld
- xlld.

Functions

This table describes the functions of the **lcld** and **xlld** programs.

This program…	Does this…	With information from the…
lcld	Loads NEW Life Cycle events records	Inbound and application data managers.
xlld	Updates EXISTING Life Cycle records that already exist in the database	Translation audit files (*edistat.i* and *edistat.o*) that the translator produces.

Load failure

If the database is full or if the database disk space is insufficient, the **xlld** and **lcld** cannot update or load Life Cycle event records to the database table. In this case,

- lcld and xlld save the data in a temporary file. The name of the temporary file is <*data_manager_name*>.*v.*<*datetime*>, where <*data_manager_name*> is the name of the data manager and <*datetime*> is the date and time that the temporary file was created.
- lcld program writes a message to the data manager log files to record the unsuccessful load attempt.

The lcld and xlld Life Cycle Process

Introduction

The **lcld** and **xlld** programs update the Life Cycle database table in different ways.

Process

This table describes how **lcld** and **xlld** are called in the Life Cycle process.

Stage	Description
1	The data manager processes all files in its work directory and produces a Life Cycle event file named <*data_manager_name*>.*v* or <*data_manager_name*>.v.<*uniqueID*>. **Note** A <*data_manager_name*>.*v* file contains all records for all the sets. A <*data_manager_name*>.*v*.<*uniqueID*> file contains an event record for one interchange, group, or set, according to the routing method.
2	The data manager passes its Life Cycle event file to the Life Cycle load programs, **lcld**.
3	The **lcld** program attempts to load NEW Life Cycle event files to the database table. **Note** If **lcld** fails, it saves the data in a temporary file named <*data_manager_name*>.v.<*datetime*>
4	The program **xlld** attempts to update records that already exist in the database with information from the translation audit files (*edistat.i, edistat.o*). **Note** The **xlld** program handles inbound functional acknowledgments the same any other record. **Reference** See How xlld Handles Functional Acknowledgments for information.

<table>
<tr><th>(Contd) Stage</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">5</td><td>IF functional acknowledgment information is…</td><td>THEN xlld updates…</td></tr>
<tr><td>Present</td><td>These columns:
The Functional Acknowledgment Date/ Time (FADT)
The Functional Acknowledgment Request Flag (FAREQ).
Comment
The detail depends on the relationship established with the trading partner.</td></tr>
<tr><td>Not present</td><td>These columns:
The three control numbers (Interchange, Application, and Set)
The Translation Date and Translation Time
The Functional Acknowledgment Request Flag.</td></tr>
<tr><td>6</td><td colspan="2">If the database is full, disk space is insufficient, or the load fails for any other reason, the lcld and xlld processes:
Fail
Save the data in a temporary file. The temporary file is <data_manager_name>.v.<datetime>, where data_manager_name is the name of the data manager and datetime is the date and time that the temporary file was created.
Write a message to the data manager’s log file to record the event.
The lcld program also writes a message to the data manager log files to record the load failure.
CAUTION
For information about unsuccessful lcld and xlld attempts, look in the data manager’s log. Note that if you run xlld from the command line, xlld will have its own log.</td></tr>
</table>

The lcld Program

Introduction

The **lcld** program loads new data manager Life Cycle event records to the Life Cycle table.

How lcld is called

How Sterling Gentran:Server calls **lcld** depends on whether or not your database is on the same machine as Sterling Gentran:Server.

This table describes how Sterling Gentran:Server calls **lcld**.

IF the database is…	AND you…	THEN…
On the same machine as Sterling Gentran:Server	—	The **lcld** program loads the Life Cycle event file as part of data manager operations if you modified the data manager's initialization file correctly.
On a different host	Have SQL network services to link the machines	The **lcld** program loads the Life Cycle event file via the SQL network as part of data manager operations.

WARNING

If your database is not on the same machine as Sterling Gentran:Server, your database administrator must install SQL network services to link the machines so that lcld and xlld can find the Life Cycle database table.

The lcld run log

The **lcld** program creates a log called <*data_manager_name*>*.l.* This file resides in the directory specified in the LIFE_CYCLE_DIR parameter of the data manager's initialization file. (By default, this is the lcld directory.) The log is a text file, so you can open it in any UNIX text editor to review it.

The log contains a record of the **lcld** processing and any errors that occurred.

Example
This is a sample line from an **lcld** run log:

```
lcld:Lifecycle:OK=0:dup=0:other=3
```

lcld versions

Each database uses a different version of **lcld**. This table describes the **lcld** version, companion files, and location.

Database	lcld program version	Location
Informix	**lcld.ec**, which is an embedded SQL C program **lc221.per**, which describes the order and length of the fields in the lc221 table	$EDI_ROOT/src/infx
Oracle	**lcld.sh**, which is a shell script **lcld.ctl**, which describes the order in which fields appear in the lc221 table	$EDI_ROOT/src/oracle
Sybase	**lcld.cp**, which is an embedded SQL C program	$EDI_ROOT/src/sybase

How to Run lcld from the Command Line

Introduction

Normally, Sterling Gentran:Server runs **lcld** as part of the data manager processing. You can also run the **lcld** program from the command line.

Format

This is the command line format:

```
lcld -f<file name>
```

where <file name> is the path name of the data manager event file without the .v extension.

Example

```
lcld -flcld/edii
```

Reference

See the *Command Reference* chapter in the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide* for more information.

The xlld Program

Introduction

The **xlld** program uses the translation audit files to update existing records in the Life Cycle table.

Translation script renames files

The model translation scripts that came with your Sterling Gentran:Server system:

- Copy the *edistat.i* and *edistat.o* files
- Rename the files to <*data_manager_name*>.*v* or <*data_manager_name*>.*v*.<*uniqid*> (You set this in the translation script.)
- Move the copies to the xlld directory.

Ways to call xlld

We suggest that you enter the **xlld** command into the translation scripts provided with your Sterling Gentran:Server software to run **xlld** from the translation scripts. Use the format for the command line invocation in the translation script.

WARNING

The xlld command must come after translation has occurred. Make sure you enter the correct path to the translation audit file.

If you experience timing issues when running **xlld** from a translation script, contact IBM customer support for other ways to call **xlld,** such as from a file data manager.

How xlld Updates the Life Cycle Table

Introduction

The columns in the Life Cycle table that **xlld** updates depend on whether or not the translation process generated functional acknowledgments.

Life Cycle table columns updated

This table describes which columns **xlld** updates.

IF FA information is…	THEN xlld updates…
Present	These columns: ◗ The Functional Acknowledgment Date/Time (FADT) ◗ The Functional Acknowledgment Request Flag (FAREQ) ◗ Other columns, depending on the relationship established with the trading partner. See the How xlld Handles Functional Acknowledgments topic.
Not present	These columns: ◗ The three control numbers (Interchange Code, Application Code, and Set Identifier) ◗ Translation date (tdt) ◗ Translation time (ttm) ◗ The Functional Acknowledgment Request Flag (for outbound only).

CAUTION

Sterling Gentran:Server removes all leading zeroes from the group control number field prior to loading it in the database. This enhances the Life Cycle update program's ability to find the correct record.

Update types

The **xlld** program performs three types of updates. This table lists the update types and the Life Cycle table columns affected by each type.

Life Cycle Table Column	Update Type		
	Set and Group	Set	Group
TP	x		
GSCTL	x	x	x
STCTL	x	x	
MYISID		x	x
TPISID		x	x
STSETID		x	
FADT*	x	x	x
FAREQ*	x	x	x

*xlld updates FADT and FAREQ only if a functional acknowledgment exists.

How xlld Handles Functional Acknowledgments

Introduction

Translation produces an *edistat.o* audit file for inbound translations if functional acknowledgments were created during translation.

The **xlld** program handles inbound functional acknowledgments as it does other translation audit records.

Functional acknowledgment update logic

When functional acknowledgment information is presentin the translation audit record (*edistat.o),* Sterling Gentran:Server bases the **xlld** update type on the presence of the Trading Partnership code (TP) and set identifier (STSETID).

This table describes the update logic.

IF the Trading Partnership code is…	AND the set identifier is…	THEN xlld attempts to update…
Present	--	All levels
Not present	Present	The Set level
Not present	Not present	The Group level

WARNING

For group-level functional acknowledgments, xlld must either update all of the referenced transaction records in the Life Cycle table or none. This is because the Life Cycle table contains an entry for each transaction set processed, and a group-level functional acknowledgment may reference many transactions. For a group level update, you *cannot* set up Life Cycle to update only selected records within the group.

Functional acknowledgment columns

If functional acknowledgment information is present, **xlld** updates these columns:

- Functional acknowledgment date (fa_date)
- Functional acknowledgment time (fa_time)
- Functional acknowledgment status (fa_req).

CAUTION

Sterling Gentran:Server combines the FA date and time into one field (FADT) before xlld updates the record.

How xlld locates the correct record

Sterling Gentran:Server uses two methods to identify the Life Cycle record to be updated with the functional acknowledgment status.

Method 1

The functional acknowledgment update mode begins when the update program encounters the character sequence “$$$997”, “$$$999”, or “$$$CTL” in the transaction set ID of the audit record. The update mode continues until **xlld** encounters a transaction set ID of “999”, “997”, or “CONTRL”.

Method 2

The **xlld** program uses the criteria described in this table to find the appropriate Life Cycle record:

IF the Trading Partner Code field of the audit record is…	**AND…**	**THEN xlld performs a…**	**USING the …**
Not empty (not filled with nulls /low values)	The set control number field is not empty	Set-level update	Trading Partnership Code And group control number And Transaction set control number
Empty (filled with nulls /low values)	The set ID is not empty	Set-level update	Interchange sender ID And interchange receiver ID And group sender ID And group receiver ID And Transaction set ID And Group control number And transaction set control number
Empty (filled with nulls /low values)	The set ID field is empty	Group-level update	Interchange sender ID And interchange receiver ID And group sender ID And group receiver ID And group control number

How xlld Handles Update Failures

Introduction

The **xlld** program takes certain actions if it can not update the Life Cycle table.

xlld actions

If **xlld** fails to update the Life Cycle table, the program:

- Renames the translation audit file to the data manager's name with a ".v" suffix followed by the day and time that the file was created. The format is <*file_name*>.ddhhmmss.
- Posts only the errors to the log file if you ran xlld with the -e option.
- Posts the entire event file to the log file if you ran xlld without the -e option.

Log name

The log name comes from the file name specified in the -f argument of the **xlld** command. If the -f argument is not in the command, the log name comes from the file name specified in the -o argument.

Where xlld activity is recorded

This table describes the location of the activity record for **xlld**.

IF you run xlld from…	The xlld activity is in the…
A translation script	Translation data manager's log file. The name of the log file is <*data_manager_name*>.*l*.
The command line	Run log called *xlld.l.* This file resides in the **xlld** run directory. The log is a text file, so you can open and review it in any UNIX text editor.

Reference
See the Monitoring Processes chapter in this guide for information about viewing log files.

The xlld Log File

Introduction

The **xlld** program creates a log file.

Log contents

Each log entry contains the process name, process ID number, and the date and time of the process. For each record that could not be updated, **xlld** records an SQL error, the Trading Partnership code, and the document reference number to the log file. The log also contains the total number of records that were read, successfully updated, and failed.

Log file name

The name of the file depends on how **xlld** is called.

If the translation data manager calls **xlld,** then the log is named *<data_manager_name>.l* where *<data_manager_name>* represents the name of the translation data manager. If **xlld** is called from anywhere other than the translation data manager, then the log is named *xlld.l*. In either case, the log file resides in the *$EDI_ROOT/xlld* directory.

Viewing the log file

The log is a text file, so you can open it in any UNIX text editor to review it.

Reference

See The lcld Program and the The xlld Program topics in this chapter for information about how **lcld** and **xlld** load and update the Life Cycle table.

How to Run xlld from the Command Line

Introduction

You can run the **xlld** program from the command line.

Before you begin

Before you run **xlld**, make sure that the translation audit file is in the directory in which **xlld** is invoked.

Format

This is the command line format (entered on one line):

xlld -f<*file_name*> -u[userID] -p[password] -e

Where…	Is the…
-f<file_name>	Path of the translation audit file without the .v extension. If you specify this option, **xlld** updates the Life Cycle table lc221 first.
-u[userID]	User ID for the Life Cycle database account
-p[password]	User password for the Life Cycle database account
-e	Argument that causes **xlld** to post only the records in error to the log file. The default is to post the entire event file.

Example

```
xlld -f../xlld/xi01
```

Reference

See the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide* for more information about the **xlld** command.

Life Cycle Reports

Overview

Introduction

Sterling Gentran:Server provides Life Cycle reports to help you organize and analyze your Life Cycle data.

These reports are available for the Informix and Oracle databases.

If you use Sybase

If you use the Sybase database product, contact IBM customer support for information about additional software products you need to run Life Cycle reports.

Life Cycle reports

These are the Life Cycle reports:

- Translation Activity Report (activity)
- Functional Acknowledgment Exception Report (fa_exc)
- Functional Acknowledgment Due Report (fadue)
- Trading Partner List Report (tplist)
- Translation Traffic Report (tracker).

Report/function table

This table describes the Life Cycle reports and their functions.

Report	Function
activity	Lists the translation activity by Set ID and Trading Partnership code. Used to determine how many invoices a trading partner sent.
fa_exc	Lists the functional acknowledgment request flag and the functional acknowledgment date for documents that pass through the translator. Used to determine whether a functional acknowledgment was sent to a particular trading partner.
fadue	Lists the functional acknowledgments that are due. Used to determine if trading partners received data and sent functional acknowledgments.

(Contd) Report	Function
tplist	Lists the interchange and group IDs of you and your trading partners along with the GS version and set IDs. Used to: ◗ Identify Trading Partnership records that you no longer need ◗ Understand the contents of a data file when you can determine only part of the Trading Partnership identifying information.
tracker	Summarizes translation traffic for a specified date by Trading Partnership code or by data manager name. Used to view the: ◗ Number of good and bad sets for inbound and outbound translation ◗ Total number of segments and total number of characters processed for the Trading Partnership code or data manager name ◗ File names, if any, that are in error.

In this section

This section describes the Life Cycle reports that tell you about your Life Cycle activity.

Reference
To learn how to generate these reports, see the Running Life Cycle Reports section in this chapter.

The Translation Activity Report

Description

The Activity Report summarizes the translator activity by Set ID and Trading Partnership code.

Sample Translation Activity Report

This is a sample Translation Activity Report.

```
TUE NOV 25                                                                                                                  PAGE    1
                                                                  STERLING GENTRAN:SERVER

                                                                  ACTIVITY REPORT

SET   TP CODE  MYISID    MYGSID    VERSION  TPISID    TPGSID    DOCUMENT   DATE       IOX SOURCE        DESTINATION      FA DATE
---  ------  -------- -------- ----- -------- -------- ------- --------- --- ---------- ------------- ----------
850  INBND850 007431125 007431125 002040 055253496 055253496 00431200 08/01/1997  I   INBD//BOX850  XLTR//TED.000000 08/01/1997
850           007431125 007431125 002040 055253496 055253496 00431200 08/01/1997  I   INBD//BOX850  XLTR//TED.000000 08/01/1997
850           007431125 007431125 002040 055253496 055253496 01431200 08/01/1998  I   INBD//BOX850  XLTR//TED.000000 08/01/1998
850           007431125 007431125 002040 055253496 055253496 01431200 08/01/1998  I   INBD//BOX850  XLTR//TED.000000 08/01/1998
850           007431125 007431125 002040 055253496 055253496 01631200 08/01/1999  I   INBD//BOX850  XLTR//TED.000000 08/01/1999
850           007431125 007431125 002040 055253496 055253496 01631200 08/01/1999  I   INBD//BOX850  XLTR//TED.000000 08/01/1999
850           007431125 007431125 002040 055253496 055253496 01831200 08/01/2000  I   INBD//BOX850  XLTR//TED.000000 08/01/2000
850           007431125 007431125 002040 055253496 055253496 01831200 08/01/2000  I   INBD//BOX850  XLTR//TED.000000 08/01/2000
850           007431125 007431125 002040 055253496 055253496 01931200 08/01/2001  I   INBD//BOX850  XLTR//TED.000000 08/01/2001
850           007431125 007431125 002040 055253496 055253496 01931200 08/01/2001  I   INBD//BOX850  XLTR//TED.000000 08/01/2001
850           007431125 007431125 002040 055253496 055253496 02031200 08/01/2002  I   INBD//BOX850  XLTR//TED.000000 08/01/2002
850           007431125 007431125 002040 055253496 055253496 02031200 08/01/2002  I   INBD//BOX850  XLTR//TED.000000 08/01/2002
850           007431125 007431125 002040 055253496 055253496 02131200 08/01/2003  I   INBD//BOX850  XLTR//TED.000000 08/01/2003
850           007431125 007431125 002040 055253496 055253496 02131200 08/01/2003  I   INBD//BOX850  XLTR//TED.000000 08/01/2003

997      2      2-IN      2-IN      7-OUT      7-OUT      19970820  08/20/1997  O    C:\MENTOR20\TEM /                08/20/1997
                                                                  16510001



15 ROWS SELECTED.
```

Translation Activity Report field descriptions

This table describes the fields of the Translation Activity Report.

Field	Description
Set	The transaction set number that identifies the type of business document.
TP Code	The Trading Partnership code.
MyISID	Your organization's interchange ID.
MyGSID	Your organization's group ID.
Version	The document version number.
TPISID	Your trading partner's interchange ID.
TPGSID	Your trading partner's group ID.
Document Date	The date that the document was processed.
IOX	The direction of the data. ◗ I = Inbound EDI ◗ O = Outbound EDI ◗ X = Application
Source	The source directory for the document.
Destination	The directory to which the document was directed.
FA Date	The date that the functional acknowledgment is expected or was received.

The FA Exception Report

Description

The FA Exception Report lists the functional acknowledgment request flag and the functional acknowledgment date for documents that pass through the translator.

Sample FA Exception Report

This is a sample FA Exception Report.

THU AUG 21 PAGE 1

STERLING GENTRAN:SERVER

FA EXCEPTION REPORT

TP CODE	MYISID	MYGSID	TPISID	TPGSID	VERSION	SET	DOCUMENT	DATE	IS CTRL#	GS CTRL#	ST CTRL#	FAREQ	FADT
INBND850	007431125	007431125	055253496	055253496	002040	850	00431200	08/01/1997	000000187	352	00000001		01/01/2006
	007431125	007431125	055253496	055253496	002040	850	01431200	08/01/1998	000000187	352	00000002		01/01/2005
	007431125	007431125	055253496	055253496	002040	850	01631200	08/01/1999	000000187	352	00000003		01/01/2004
	007431125	007431125	055253496	055253496	002040	850	01831200	08/01/2000	000000187	352	00000004		01/01/2003
	007431125	007431125	055253496	055253496	002040	850	01931200	08/01/2001	000000187	352	00000005		01/01/2002
	007431125	007431125	055253496	055253496	002040	850	02031200	08/01/2002	000000187	352	00000006		01/01/2001
	007431125	007431125	055253496	055253496	002040	850	02131200	08/01/2003	000000187	352	00000007		01/01/2000

7 ROWS SELECTED.

FA Exception Report field descriptions

This table describes the fields of the FA Exception Report.

Field	Description
TP Code	The Trading Partnership code.
MyISID	Your organization's interchange ID.
MyGSID	Your organization's group ID.
TPISID	Your trading partner's interchange ID.
TPGSID	Your trading partner's group ID.
Version	The document version number.
Set	The transaction set number that identifies the type of business document.
Document	The document reference number.

(Contd) Field	Description
Date	The date that the document was processed.
IS Ctrl#	The interchange control number.
GS Ctrl#	The group control number.
ST Ctrl#	The set control number.
FaReq	The functional acknowledgment request flag. ◗ Y = FA has been requested ◗ N = FA has not been requested
FAdt	The date that the functional acknowledgment was received.

The FA Due Report

Description

The Functional Acknowledgment Due Report lists the functional acknowledgments that are due.

Sample FA Due Report

This is a sample FA Due Report.

```
FUNCTIONAL ACKNOWLEDGEMENT DUE REPORT


TRADING PARTNER SET    XLDATE     XLTIME  FAREQ IS CONTROL GS CONTROL ST CONTROL DOCUMENT REFERENCE NUMBER
--------------- ------ ---------- ------- ----- ---------- ---------- ---------- -------------
2               997    08/20/1997  1651          000000047  50         0001       1997082016510001

INBND850        850    08/01/1997  105530        000000187  352        00000001   00431200
                850    08/01/1997  105530        000000187  352        00000001   00431200
                850    08/01/1998  105629        000000187  352        00000002   01431200
                850    08/01/1998  105629        000000187  352        00000002   01431200
                850    08/01/1999  105728        000000187  352        00000003   01631200
                850    08/01/1999  105728        000000187  352        00000003   01631200
                850    08/01/2000  105827        000000187  352        00000004   01831200
                850    08/01/2000  105827        000000187  352        00000004   01831200
                850    08/01/2001  105926        000000187  352        00000005   01931200
                850    08/01/2001  105926        000000187  352        00000005   01931200
                850    08/01/2002  106025        000000187  352        00000006   02031200
                850    08/01/2002  106025        000000187  352        00000006   02031200
                850    08/01/2003  106124        000000187  352        00000007   02131200
                850    08/01/2003  106124        000000187  352        00000007   02131200


15 ROWS SELECTED.
```

FA Due Report field descriptions

This table describes the fields of the FA Due Report.

Field	Description
Trading Partner	The Trading Partnership code.
Set	The transaction set number that identifies the type of business document.
XlDate	The date that the document was translated
XlTime	The time that the document was translated.
FaReq	The functional acknowledgment request flag.

(Contd) Field	Description
IS Control	The interchange control number.
GS Control	The group control number.
ST Control	The set control number.
Document Reference Number	The document reference number.

The TP List Report

Description

The Trading Partner List Report lists all the data contained in a Trading Partnership record.

Sample TP List Report

This is a sample TP List Report.

```
THU AUG 21                                                                                                                                PAGE    1
                                                GENTRAN-SERVER

                                                  TP LIST

TP ID            MY ISA ID           MY GS ID            TP IS ID            TP GS ID            GS VERSION SET
------------------- ------------------- ------------------- ------------------- ------------------- ---------- ------
2                   2-IN                2-IN                7-OUT               7-OUT                      997

INBND850         007431125           007431125           055253496           055253496           002040     850
                 007431125           007431125           055253496           055253496           002040     850
                 007431125           007431125           055253496           055253496           002040     850
                 007431125           007431125           055253496           055253496           002040     850
                 007431125           007431125           055253496           055253496           002040     850
                 007431125           007431125           055253496           055253496           002040     850
                 007431125           007431125           055253496           055253496           002040     850

8 ROWS SELECTED.
```

TP List Report field descriptions

This table describes the fields of the TP List Report.

Field	Description
TP ID	The Trading Partnership code.
My ISA ID	Your organization’s interchange ID.
My GS ID	Your organization’s group ID.
TP IS ID	Your trading partner’s interchange ID.
TP GS ID	Your trading partner’s group ID.

(Contd) Field	Description
GS Version	The document version number.
Set	The transaction set number that identifies the type of business document.

The Translation Traffic Report

Description

The Translation Traffic Report summarizes translation traffic for a specified date.

The tracker command line program

Tracker is a Sterling Gentran:Server command line program that generates the Translation Traffic Report. The command has a number of arguments you can use to request data that you want included in the report.

What the tracker program does

The tracker program:

- Retrieves the information you requested from the Oracle or Informix Life Cycle table
- Accumulates the number of good and bad sets for inbound and outbound translation
- Accumulates the total number of segments and total number of characters processed for the Trading Partnership code or data manager name
- Produces the report, listing:
  - The accumulated numbers by Trading Partnership code or data manager, as requested
  - The file names, if any, that are in error.

Sample report

This is a sample report that the tracker program produced.

Traffic Report for 14-dec-96 (By TP Code)

TP Code	Good_ in	Good_O ut	Bad_ In	Bad_O ut
INBND210	22	0	0	0
INBND837	4	0	0	0
INBND850	22	0	0	0
OUTBND02856	0	6	0	0
OUTBND03856	0	6	0	0
TDCC204-1	0	66	0	0
TDCC204-2	0	66	0	0

Total Traffic:	198				
Total Inbound:	48	Good:	48	Bad:	0
Total Outbound:	150	Good:	150	Bad:	0
Total Other:	0				

Total Segments:	1422
Inbound:	1302
Outbound:	120

Total Characters:	201170
Inbound:	158594
Outbound:	42576
Other:	

Listing of files in error (if any)

Traffic Report field descriptions

This table describes the fields of the Traffic Report.

Field	Description
TP Code	The Trading Partnership code.
Good_in	The number of inbound transaction sets that the translator processed successfully.
Good_Out	The number of outbound transaction sets that the translator processed successfully.
Bad_In	The number of inbound transaction sets that the translator could not process.
Bad_Out	The number of outbound transaction sets that the translator could not process.
Total Traffic	The total number of inbound, outbound, and other transaction sets that the translator processed.
Total Inbound	The total number of inbound transaction sets that the translator processed.
Total Outbound	The total number of outbound transaction sets that the translator processed.

(Contd) Field	Description
Total Other	The total number of application transaction sets that the translator processed.
Total Segments	The total number of segments that the translator processed.
Inbound	The number of inbound segments that the translator processed.
Outbound	The number of outbound segments that the translator processed.
Other	The number of application segments.
Total Characters	The total number of characters that the translator processed.
Inbound	The number of inbound EDI characters that the translator processed.
Outbound	The number of outbound EDI characters that the translator processed.
Other	The number of application characters that the translator processed.
Listing of files in error (if any)	The files that contained errors.

Running Life Cycle Reports

Overview

Introduction

This section explains how to run the Informix and Oracle Life Cycle reports described in the previous section.

Reports generation sources

You run these reports from your Life Cycle database:

- Translation Activity
- FA Exception
- FA Due
- TP List.

You must run the Translation Traffic Report from the UNIX command line with the **tracker** program command. The command has several arguments.

Using this section

This table describes how to use this section of the chapter.

IF you want to run this report…	AND your Life Cycle database is…	THEN see…
Translation Activity FA Exception FA Due TP List	Informix	How to Run Informix Life Cycle Reports.
	Oracle	How to Run Oracle Life Cycle Reports
Translation Traffic	--	How to Run the Translation Traffic Report

Before you begin

Before you attempt to run a Life Cycle report, make sure that your organization has loaded the reports to your database.

Reference

See the *IBM® Sterling Gentran:Server® for UNIX Maintenance and Troubleshooting Guide* for information.

How to Run Informix Life Cycle Reports

Introduction

This topic explains how to run the following reports from your Informix database software:

- Translation Activity (activity.ace)
- FA Exception (fa_exc.ace)
- FA Due (fadue.ace)
- TP List (tplist.ace).

Before you begin

To run the Informix Life Cycle reports, the application **isql** must be in your path.

Procedure

Use this procedure to run Informix Life Cycle reports.

Step	Action
1	At the UNIX command line, open the directory that contains the files that will generate the Life Cycle report files. **CAUTION** **The name of the directory that holds the *.ace* files is *$EDI_ROOT/src/infx*.**
2	Type **isql** at the prompt and press ENTER. **System Response** System displays the Informix interactive environment screen.
3	Select **Report** from the menu and then press ENTER. **System Response** The system displays the Report menu.
4	Select **Run** from the Report menu and then press ENTER. **System Response** The system displays a list of reports.
5	Select the report that you want to run and then press ENTER. **System Response** The system displays messages about the report generation.

How to Run Oracle Life Cycle Reports

Introduction

This topic explains how to run the following reports from your Oracle database software:

- Translation Activity (activity.sql)
- FA Exception (fa_exc.sql)
- FA Due (fadue.sql)
- TP List (tplist.sql).

Before you begin

To run the Oracle Life Cycle reports, the application **sqlplus** must be in your path. You must also know the path to the Life Cycle report files.

Procedure

Use this procedure to run Oracle Life Cycle reports.

Step	Action
1	Go to the UNIX command line prompt.
2	Type **sqlplus** and then press ENTER. **System Response** The system displays the Oracle sign-in screen.
3	Enter your user name and password. **System Response** The system displays the SQL prompt.
4	Enter the path to the report file that you want to run and the name of the report. You must include the report file name extension, *.sql*. **Example** To run the Translation Activity Report, enter the following: @<*full_path*>/activity.sql Where <*full_path*> is the full path to the Life Cycle report files. **System Response** The system creates the report file in the current directory and displays the report on screen.

How to Run the Translation Traffic Report

Introduction

The Translation Traffic Report is run from the UNIX command line with the **tracker** command.

Invocation

This is the invocation format for the **tracker** program:

tracker -d<*date*> -u<*username*> -p<*password*> -t[*Tpcode*] -a[*Dmname*] -b -l[*printer_name*] -v

Arguments table

This table describes the command arguments.

Argument	Required or optional	Function
-d	Optional	Defines the date of translation. The format is CCYYMMDD.
-u	Optional	Defines the user's login name for the database, if any.
-p	Optional	Defines the user's password for the database, if any.
-t	Required	Defines the Trading Partnership code.
-a	Required	Defines the data manager's name.
-b	Optional	Generates the report by Trading Partnership code and data manager name.
-l	Optional	Defines the name of the printer that is to print the report.
-v	Optional	Displays the report on screen.

CHAPTER
10

Monitoring Processes

Contents

Overview
- Introduction 2
- The Screen Viewer 4

Monitoring Data Manager Processes
- Overview 6

Data Manager Control Screen
- Data Manager Log Files 10
- How to Check a Data Manager's Status 12
- How to View a Data Manager's Log File 13

Starting and Stopping Data Managers
- Overview 15
- How to Use the Data Manager Control Screen 16
- How to Stop Data Managers 19
- How to Start Data Managers 22

Maintaining Data Manager Log Files
- Overview 25
- How to Purge Data Manager Log Entries 26
- How to Delete a Data Manager's Log File 28

Monitoring Scripts
- Overview 29
- Script Logs and Journals 30
- How to Check the Status of a Script 32
- How to View a Script Log or Journal 34
- How to Display a Script's Processing Time Statistics 35

Maintaining Script Logs and Journals
- Overview 37
- How to Purge Entries from Journals 38
- How to Delete a Log or Journal 40

Overview

Introduction

In this chapter

This chapter explains how to:

- Monitor data manager processes
- Maintain data manager log files
- Monitor script processes
- Maintain script logs and journals.

Key terms

This table lists the key terms used in this chapter.

Term	Description
agent	A data manager.
cleanlog	The command line tool that enables you to purge log file entries.
data manager	A program that periodically scans a directory or queue for data files and then processes the files it finds. Processing can include: - Routing data - Invoking scripts - Archiving data - Handling data errors.
log file	A file that contains a record of process activity and messages produced by that activity. **Example** A data manager log contains a record of the data manager's status at a given date and time and any message produced.
script journal	The file that contains a record of a script's previous activity. When a script finishes, it appends the data in its log file to its journal.

(Contd) Term	Description
script log	The file in which a script records its process activity while it is active. Each time the script starts, it creates a new log file.
status	The running state (active or inactive) of a script or data manager.
startserver	The UNIX shell script used to start data managers.
stopserver	The UNIX shell script that stops active data managers.

The Screen Viewer

Introduction

You view both data manager and script log files with the Screen Viewer, which is an ASCII text file viewer.

Example: Data manager log

This illustration shows an example how the Screen Viewer displays a data manager's log file.

```
Agent Log: ./file.l
file:11736:09091998:144545:  0:Began, pid=11736, file Revision:@(#) init_h
file:11736:09091998:144545:  0:                EDI_ROOT:  /qabox/qa511:
file:11736:09091998:144545:  0:                ENV_ROOT:  /qabox/qa511:
file:11736:09091998:144545:  0:             EDI_MAILBOX:  /qabox/qa511/mb
file:11736:09091998:144545:  0:             PERSONALITY:  file:
file:11736:09091998:144545:  0:              SCAN_DELAY:  0:
file:11736:09091998:144545:  0:            LOCK_ATTEMPT:  3:
file:11736:09091998:144545:  0:                  NOTIFY:  0:
file:11736:09091998:144545:  0:  ADD_MBAG_TO_DOC_REF_NUM:  0:
                                        F1:Help F3:Erase F9:Quit
```

Screen Viewer function keys

This table describes the function keys of the Screen Viewer.

Key	Function
F1	Displays help information.
F3	Erases the log file.
F9	Exits the screen.

Log navigation keys

Use the keys in this table to navigate the log.

Keys	Action
B or b	Position viewer window to bottom of file.
E or e	Position viewer window to next error.
F or f	Follow active file.
H or h or Left Arrow	Scroll viewer window left one character.

Keys	Action
J or j or Down Arrow	Scroll viewer window down one line.
K or k or Up Arrow	Scroll viewer window up one line.
L or l or Right Arrow	Scroll viewer window right one character.
R or r	Toggle between small and large viewer windows.
T or t	Position viewer window to top of file.

Monitoring Data Manager Processes

Overview

In this section

This section explains how to monitor individual data manager processes. It contains these topics:

- Data Manager Control Screen
- Data Manager Log File
- How to Check a Data Manager's Status
- How to View a Data Manager's Log File
- How to Start or Stop a Data Manager.

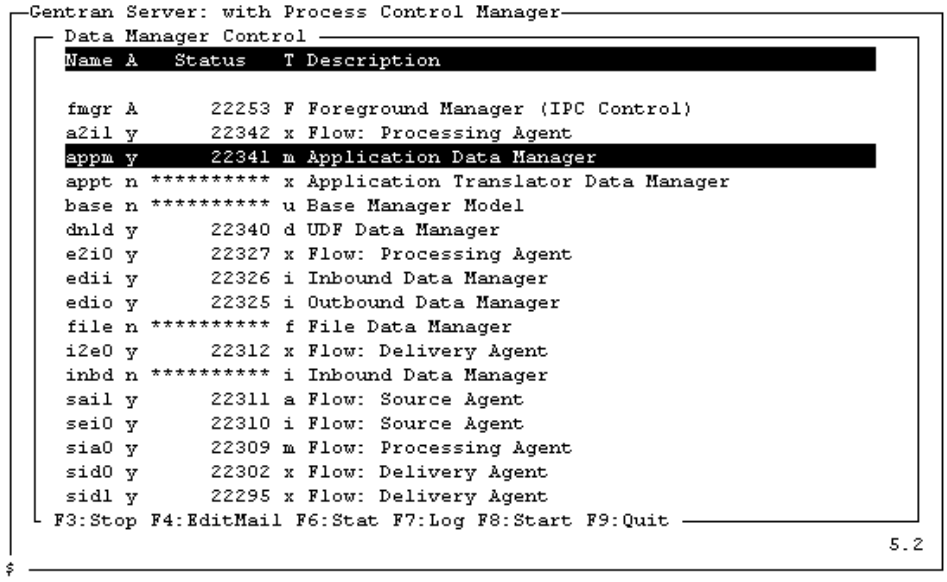
Data Manager Control Screen

Introduction

You start the data manager tasks described in this section from the Data Manager Control screen.

Data Manager Control screen

This illustration shows an example of the Data Manager Control screen.



Fields and functions

This table lists the fields of the Data Manager Control screen and their functions.

Field	Function	
Name	Displays the data manager's name. Type up to four characters. You may use any combination of alphabetic and numeric characters.	
A (Autostart)	Determines whether the data manager starts automatically when the foreground manager starts. ◗ Y = Yes ◗ N = No	
Status	Indicates the running status of the data manager. **Tip** Press F6 to update the status.	
	IF the column displays…	**THEN the data manager is…**
	A series of asterisks (******)	Not running.
	A zero (0)	Not running.
	A numeric process ID	Running.
	The word "Ending"	The data manager is stopping because someone issued a stop command.

<table>
<tr><th>(Contd) Field</th><th colspan="2">Function</th></tr>
<tr><td>T</td><td colspan="2">Defines the data manager’s personality type.</td></tr>
<tr><td></td><td>Type Code</td><td>Description</td></tr>
<tr><td></td><td>F</td><td>The foreground manager</td></tr>
<tr><td></td><td>d</td><td>Download</td></tr>
<tr><td></td><td>i</td><td>Inbound</td></tr>
<tr><td></td><td>a</td><td>Archive</td></tr>
<tr><td></td><td>f</td><td>File</td></tr>
<tr><td></td><td>x</td><td>Translator</td></tr>
<tr><td></td><td>l</td><td>Line manager</td></tr>
<tr><td></td><td>h</td><td>Host command card</td></tr>
<tr><td></td><td>m</td><td>Application</td></tr>
<tr><td>Description</td><td colspan="2">Describes the data manager. Enter up to 50 characters.
CAUTION
For an archive data manager, this field contains the archive data manager’s processing parameters.</td></tr>
</table>

Function keys of the Data Manager Control screen

This table describes the function keys of the Data Manager Control screen.

Key	Function
F3	Stops a running data manager.
F4	Opens the default editor so that you can edit the data manager’s UNIX mail script.
F6	Refreshes the screen and updates the data manager’s running status, which is displayed in the Status field.
F7	Displays the data manager’s log file.
F8	Starts a data manager that is not running.
F9	Exits the screen.

Data Manager Log Files

Introduction

When a data manager handles a data set, it produces a record of the event. These records are stored in the data manager's process or event **log file**.

Events recorded in a log file

These are the types of events that are logged:

- Data manager starts and stops
- Processing results and status messages
- Error conditions
- Life Cycle load information, if you are loading event files to a Life Cycle database table.

CAUTION

Sterling Gentran:Server also creates a log record when you make a change to the data manager's initialization file. The change is logged under a new process ID number, and the entry contains the initialization file parameters.

Sterling Gentran:Server updates the log when you add a new configuration record to the data manager.

Example log file

This is an example of a data manager log file.

```
Agent Log: ./file.l
file:11736:09091998:144545:  0:Began, pid=11736, file Revision:@(#) init_h
file:11736:09091998:144545:  0:                EDI_ROOT:  /qabox/qa511:
file:11736:09091998:144545:  0:                ENV_ROOT:  /qabox/qa511:
file:11736:09091998:144545:  0:             EDI_MAILBOX:  /qabox/qa511/mb
file:11736:09091998:144545:  0:             PERSONALITY:  file:
file:11736:09091998:144545:  0:              SCAN_DELAY:  0:
file:11736:09091998:144545:  0:            LOCK_ATTEMPT:  3:
file:11736:09091998:144545:  0:                  NOTIFY:  0:
file:11736:09091998:144545:  0:  ADD_MBAG_TO_DOC_REF_NUM:  0:
                                        F1:Help F3:Erase F9:Quit
```

Data manager log file format

This is the general format of the lines in the data manager log:

DMNM:pid:date:time:return code:message

CAUTION

Other information that may be included, such as Life Cycle load information, is not in this format.

Parts of the log file format

This table describes the parts of the general format.

Part	Description
DMNM	The data manager's name.
pid	The process ID number.
date	The date of the event in MMDDCCYY format.
time	The time of the event in HHMMSS format.
return code	The status of the data manager, expressed as the result code. ◗ 0 implies success ◗ Any other code implies an error.
message	The text message that the data manager activity produced.

Where log files are stored

Data manager logs are named *$EDI_ROOT/<User Files>/<dmnm>.l*, where *<dmnm>* is the data manager's name. The <User Files> directory is the directory specified for User Files on the Set Up Directories dialog box. The default directory for User Files is $EDI_ROOT/Temp, but your system may store User Files in a different directory.

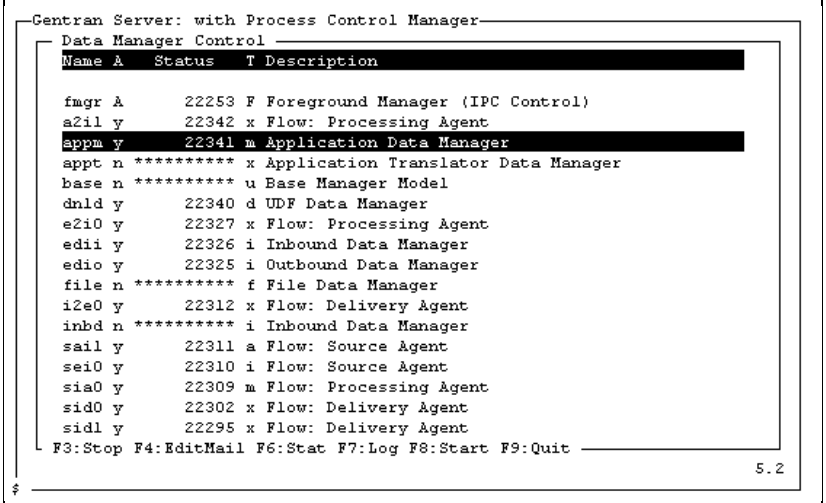
How to Check a Data Manager's Status

Introduction

The **Data Manager Control** screen lists the status of each data manager as of the time you accessed the screen. You can press F6 to update the status fields any time you want to see the current status.

Control screen

Use this procedure to check a data manager's status.

Step	Action
1	Select **DataMgr** from the Sterling Gentran:Server host Main Menu. **System Response** Sterling Gentran:Server displays the Data Manager Control screen. This screen lists all the data managers added to Sterling Gentran:Server. 
2	Check the status column of the data manager to determine whether or not it is active.
3	Press F6 to update the Status column. **System Response** Sterling Gentran:Server replaces the displayed status with the current status.
4	When you are finished, press F9 to exit the screen.

How to View a Data Manager's Log File

Introduction

You can access a data manager's process log from the Data Manager Control screen.

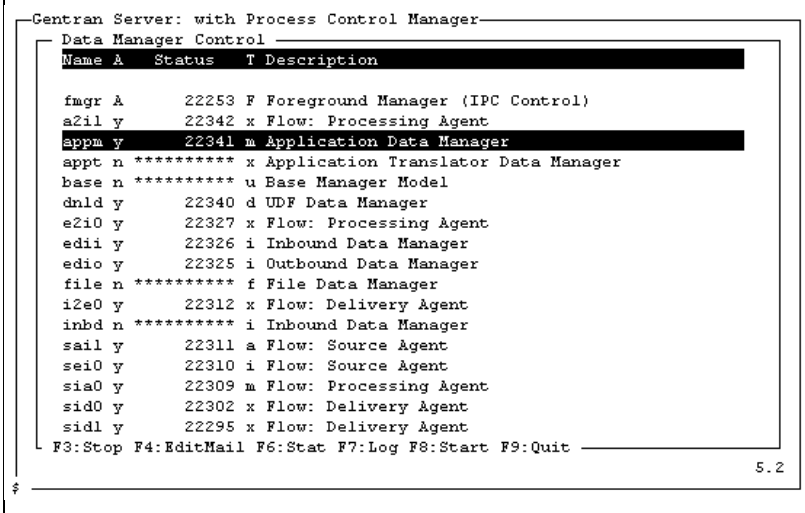
When to use

Use these procedures when you want to:

- Search for an error that occurred during the data manager's activity
- Trace processing messages for a data manager as they are produced.

Procedure

Use this procedure to access a data manager's log file from the Data Manager Control screen.

Step	Action
1	Select **DataMgr** from the Sterling Gentran:Server host Main Menu. **System Response** Sterling Gentran:Server displays the Data Manager Control screen. This screen lists all the data managers added to Sterling Gentran:Server. 
2	Select the data manager and then press F7 to view the log.

(Contd) Step	Action
3	Use the navigation keys to position the viewer window and move through the log file.
4	When you are finished, press F9 to exit the log.

Starting and Stopping Data Managers

Overview

Introduction

You can start or stop a data manager by:

- Selecting the data manager's name from the Data Manager Control screen and pressing the appropriate function key
- Running the **startserver** or the **stopserver** command
- Running the **startserver.sh** or the **stopserver.sh** UNIX shell script

In this section

This section contains these topics:

- How to Use the Data Manager Control screen (to start and stop data managers)
- How to Stop Data Managers
- How to Start Data Managers

How to Use the Data Manager Control Screen

Introduction

You can start or stop a data manager from the Data Manager Control screen.

Primary control required

You must have primary control to start or stop a data manager. This means that you must be the first user to invoke the Data Manager Control screen. If another user has primary control, Sterling Gentran:Server displays a message to let you know who has primary control.

Requested termination

When you stop a data manager from the Data Manager Control screen, the foreground manager (fmgr) sends the data manager a termination request.

CAUTION

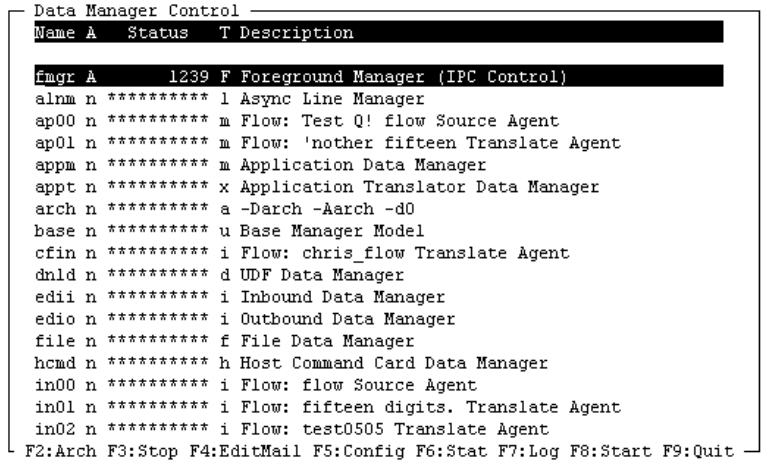
If the data manager is processing a document when it receives the termination request, it completes document processing before stopping.

Unrequested termination

If a data manager terminates without a request from the foreground manager, fmgr is configured to send a mail message. The *$EDI_ROOT/.fmgr.l log* file contains information about the termination.

Procedure

Use this procedure to start or stop a data manager.

Step	Action
1	Select **DataMgr** from the Sterling Gentran:Server host Main Menu. **System Response** Sterling Gentran:Server displays the Data Manager Control screen. 
2	Check the Status column of the data manager to determine whether or not it is active. To display current information in the Status column, press F6 to update the status. **Reference** See the How to Check a Data Manager's Status topic in this section for more information.
3	Select the data manager you want to start or stop.
4	Press the appropriate function key: ◗ F3 to stop the data manager ◗ F8 to start the data manager.

CAUTION

If you issue a stop command when a data manager is working on data, the data manager's process ID value (PID) in the Status field changes to Ending. The data manager completes processing before it stops.

Asterisks in the Status fields indicate the data manager is not active. If you start a data manager, Sterling Gentran:Server replaces the asterisks with a process ID value, which is an integer value the operating system assigns to the process to distinguish it from other processes.

You may need to press F6 to see these changes in the Status column.

How to Stop Data Managers

Introduction

Your Sterling Gentran:Server product includes a UNIX shell script named **stopserver.sh** and a command named **stopserver**. Both can be used to stop active data managers, including the foreground manager, **fmgr**.

You can run the **stopserver** command or the **stopserver.sh** script from:

- A script or batch file
- The command line.

What stopserver and stopserver.sh do

You can use **stopserver** or **stopserver.sh** to stop a single data manager, or **fmgr** and all running data managers.

Both the **stopserver** command and the **stopserver.sh** script direct **fmgr** to send the data manager a termination request. If the data manager is processing a document when it receives the termination request, it completes document processing before stopping. After all the data managers have stopped, **stopserver** or **stopserver.sh** stops the foreground manager.

The **stopserver.sh** script also checks the system for other running processes, such as the script manager (**smgr**), FTP, and communications (3780Plus and aplus). It finds the foreground manager's process ID and checks the UNIX process list for processes that have this process ID as the parent.

When to use

Use **stopserver** or **stopserver.sh** when you want to stop:

- All the data managers configured to autostart with the foreground manager
- One or all data managers as part of a script or batch process
- One or all data managers from the command line.

Running stopserver or stopserver.sh from the command line

Use this procedure to run **stopserver** or **stopserver.sh** from the command line.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">Go to the command line.</td></tr>
<tr><td>2</td><td colspan="2">Type the stopserver command or the stopserver.sh script command.</td></tr>
<tr><td></td><td>IF you want to stop…</td><td>THEN type…</td></tr>
<tr><td></td><td>A single data manager</td><td>stopserver <<i>dmname</i>>
OR
stopserver.sh -d <<i>dmname</i>>
where <<i>dmname</i>> is the name of the data manager.
Examples
stopserver edii
stopserver.sh -d edii</td></tr>
<tr><td></td><td>The foreground manager and all other data managers</td><td>stopserver
OR
stopserver.sh</td></tr>
<tr><td></td><td colspan="2">System Response</td></tr>
<tr><td></td><td>IF you…</td><td>THEN stopserver or stopserver.sh…</td></tr>
<tr><td></td><td>Specified a data manager</td><td>Has the foreground manager wait until the data manager finishes processing and then stops the data manager.</td></tr>
<tr><td></td><td>Did not specify a data manager</td><td>Displays a list of data managers that are running and prompts you to choose a timeout value.
Continue with Step 3.</td></tr>
<tr><td>3</td><td colspan="2">Type the number of the timeout value you want stopserver or stopserver.sh to use or press ENTER to use the default timeout value.
System Response
Sterling Gentran:Server sends a termination message to the foreground manager and the data managers. When the data managers stop, stopserver or stopserver.sh displays this message: The agents of foreground manager (xxxxx) are stopped.</td></tr>
</table>

Running stopserver or stopserver.sh from a script or batch file

To run **stopserver** or **stopserver.sh** from a script or batch file, type the following commands in the script or batch file.

IF you want to stop…	THEN use this command format…
A single data manager	**stopserver** <*dmname*> OR **stopserver.sh** -d <*dmname*> where <*dmname*> is the name of the data manager. **Examples** **stopserver** edii **stopserver.sh** -d edii
The foreground manager and all data managers	**stopserver** OR **stopserver.sh**

Note
For more information about the **stopserver** command or the **stopserver.sh** script, see the *Command Reference* chapter of the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide.*

How to Start Data Managers

Introduction

Your Sterling Gentran:Server product includes a command named **startserver** and a UNIX shell script named **startserver.sh**. Both can be used to start either a specified data manager or all data managers configured to autostart with the foreground manager, **fmgr**.

You can run the **startserver** command or the **startserver.sh** script from:

- A script or batch file.
- The command line.

For more information, see the *Command Reference* chapter of the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide.*

What startserver and startserver.sh do

You can use **startserver** or **startserver.sh** to start a single data manager or all the data managers configured to autostart with the foreground manager, **fmgr**.

If the foreground manager is not running when you run **startserver** or **startserver.sh**, the command or script starts **fmgr** first (as a new process) and then starts the data managers configured to autostart with the foreground manager.

If the foreground manager is running when you run **startserver** or **startserver.sh**, the command or script restarts only those data managers configured to autostart with the foreground manager.

Script recovery

The **startserver.sh** script runs the startserver command. But first, it calls the Sterling Gentran:Server ***recover.scr*** recover script. The recover script restarts (from the beginning) specified Sterling Gentran:Server scripts after a machine halt. The recover script examines the log files in the *temp* directory to determine if any scripts need to be restarted.

When to use

Use **startserver** or **startserver.sh** when you want to start:

- All the data managers configured to autostart with the foreground manager
- One or all data managers as part of a script or batch process
- One or all data managers from the command line

Running startserver or startserver.sh from the command line

Use this table to run **startserver** or **startserver.sh** from the command line.

IF you want to start…	THEN use this command line format…
A single data manager	**startserver <***dmname***>** OR **startserver.sh** -d <*dmname*> where <*dmname*> is the name of the data manager. **Examples** **startserver** edii **startserver.sh** -d edii
The foreground manager and all data managers configured to autostart with the foreground manager	**startserver** OR **startserver.sh**

Running startserver or startserver.sh from a script or batch file

To run **startserver** or **startserver.sh** from a script or batch file, type the following commands in the script or batch file.

IF you want to start…	THEN use this command format…
A single data manager	**startserver** <*dmname*> OR **startserver.sh** -d <*dmname*> where <*dmname*> is the name of the data manager. **Examples** **startserver** edii **startserver.sh** -d edii
The foreground manager and all data managers configured to autostart with the foreground manager	**startserver** OR **startserver.sh**

Note
If you run **startserver** <*dmname*> or **startserver.sh** -d <*dmname*> when **fmgr** is not running, the data managers start in this order:

- The foreground manager, **fmgr**
- All the data managers configured to autostart with foreground manager
- The data manager you specified in the <*dmname*> parameter

Maintaining Data Manager Log Files

Overview

In this section

This section explains how to maintain data manager log files. It contains these procedures:

- How to Purge Data Manager Log Entries
- How to Delete a Data Manager's Log File.

How to Purge Data Manager Log Entries

Introduction

Data manager log files can take up a large amount of disk space. For this reason, you should purge the files periodically. Sterling Gentran:Server provides a command line tool called cleanlog that enables you to purge log file entries.

Log file location and name

Data manager log files are in the $EDI_ROOT/*temp* directory. The name of the log file is the data manager's name with a ".l" extension.

Example
The name of the dnld data manager's log is *dnld.l.*

Purging options

The cleanlog tool enables you to purge log file entries. You have the option of purging entries:

- Older than 90 days
- Older than a specific date
- Within a specific date range.

You may purge entries in all log files in the current directory or in a single log file.

When to use

Use this procedure to remove old log file entries that you no longer need.

Purging log entries

Use this procedure to purge log entries.

Step	Action
1	Go to the UNIX command line and make the EDI_ROOT/*temp* directory the current directory.

<table>
<tr><th>(Contd) Step</th><th>Action</th></tr>
<tr><td>2</td><td>
From the following table, choose the type of purge you want to perform and enter the command at the command line.

Rules
Enter the name of the log in the <logfile> portion of the command.

Enter dates in the format MMDDYY or MMDDCCYY.
<table>
<tr><th>To purge…</th><th>Use this command…</th></tr>
<tr><td>All entries older than 90 days in all logs</td><td><code>ksh cleanlog.sh</code></td></tr>
<tr><td>All entries older than the specified number of days in all logs</td><td><code>ksh cleanlog.sh <days></code></td></tr>
<tr><td>All entries older than the specified number of days in a single log</td><td><code>cleanlog <logfile> -a<days></code></td></tr>
<tr><td>All entries for a specific date range in a single log</td><td><code>cleanlog <logfile> -B<delete from date> -E<delete to date></code></td></tr>
</table>
Reference
For more information about the cleanlog command, see the Command Reference chapter of the IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide.
</td></tr>
</table>

How to Delete a Data Manager's Log File

Introduction

When you no longer need a data manager's log file, you can delete it.

WARNING

You cannot recover a deleted log file. Make sure that you do not need it before you delete the file.

Permissions

You must have write permission to the log files in order to delete them.

Deleting a log file

Use this procedure to delete a data manager's log file.

Step	Action
1	Select **DataMgr** from the Sterling Gentran:Server host main menu. **System Response** Sterling Gentran:Server displays the Data Manager Control screen. This screen lists all the data managers added to Sterling Gentran:Server.
2	Select the data manager and then press F7 to view its log file.
3	Press F3 to delete the log file. **System Response** Sterling Gentran:Server deletes the log file and returns to the Data Manager Control screen.

Monitoring Scripts

Overview

Introduction

To monitor a script's activity, you can check the script's log or journal. Script logs and journals are a good place to look for problems when you are troubleshooting data flow.

You can also display processing time statistics for a script. Processing time statistics show how much CPU time was needed to execute the instructions in a script.

Screen Viewer

You view a script's log or journal with the Screen Viewer, which is an ASCII text file viewer. This is the same Screen Viewer that you use to view data manager log files.

Reference
See The Screen Viewer topic in this chapter for information about navigating the log or journal in the Screen Viewer.

In this section

This section contains these topics:

- Script Logs and Journals
- How to Check the Status of a Script
- How to View a Script Log or Journal
- How to Display a Script's Processing Time Statistics.

Script Logs and Journals

Introduction

Sterling Gentran:Server scripts record their process activity in script logs and script journals.

Script log

When a Sterling Gentran:Server script is running, its process activity is recorded in a log in the *./temp* directory. Log records display each script step and its result. Sterling Gentran:Server keeps the log file only while the script is running. Each time a script restarts, it creates a new log file.

Example
This is an example of a script log.

```
smgr:4502:10121998:151144:0:Began beeper, smgr Revision:@(#) smgr.c 5.133/20/97
sint:4502:10121998:151144:0:End ENVIRON group 0 variables loaded. :
sint:4502:10121998:151144:0:Begin lock group: | -->dummy label <--|:
sint:4502:10121998:151144:0: locked, ./script/beeper.scr   :
sint:4502:10121998:151144:0:  locked, ./script/beeper.scr:
sint:4502:10121998:151144:0:End lock group: |-->dummy label <--|:
sint:4502:10121998:151144:0:STEP CNT:0:STEP in:start:if mv_nums then beeper else
sint:4502:10121998:151144:1:   mv_nums:mv script/beeper.nums script/tmp.nums :
sint:4502:10121998:151144:0:STEP CNT:0:STEP in:end-1:if end-1 then end else end:
sint:4502:10121998:151144:2:   end-1: rm ./temp/beeper.old 2>/dev/null :
smgr:4502:10121998:151144:0:Ended beeper:
```

Script log name

The name of the log file is the script's name followed by a "_PID.l" (PID = Process ID, ".l" for "log") suffix.

Example
A script named *xfer* creates a log named *xfer_PID.l* in the *./temp* directory.

Script log file format

This is the general format of the lines in a script log:

```
process name:process ID number:date:time:return code:message
```

Parts of the script log format

This table describes the parts of the general format.

Part	Description
Process name	The name of the script process. **Example** The name of the script manager process is **smgr**.
process ID number	The process ID number of the script process.
date	The date of the event in MMDDCCYY format.
time	The time of the event in HHMMSS format.
return code	The status of the script, expressed as the return code. ◗ 0 implies success ◗ Any other code implies an error
message	The text message that the script activity produced.

Script journal

A script's journal (<*scriptname*>.*old*) contains all the script's previous process activity.

When a script has finished running, Sterling Gentran:Server:

- Appends the data in the script's log (<*scriptname*>_PID.*l*) to the script's journal (<*scriptname*>.*old*) if the journal exists.
- Creates <*scriptname*>.*old* if the journal does not exist.
- Deletes the script's log.

A script journal looks like a script log. Both have the same general format.

Journal name

The name of the journal is the script's name followed by the suffix *old.*

Example
A script named *xfer* creates a journal named *xfer.old.*

A script named *sint* creates a journal named *sint.old*.

Location

Like script logs, script journals are in the *./temp* directory.

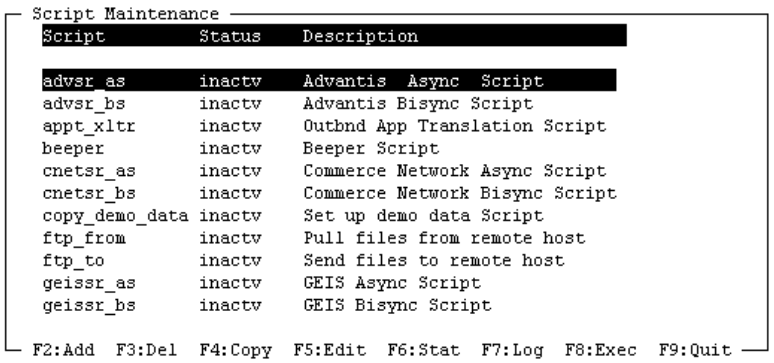
How to Check the Status of a Script

Introduction

To determine if a Sterling Gentran:Server script is running, you can check its status on the Script Maintenance screen. Sterling Gentran:Server bases the status on the presence or absence of the script's log file. If the log is present, the status is active. If the log is absent, the status is inactive.

Procedure

Use this procedure to check the status of a script.

Step	Action
1	Select **Script** from the host main menu to access the Script Maintenance screen.  **System Response** Sterling Gentran:Server: ◗ Scans the *./temp* directory to determine which scripts have log files ◗ Displays a list of the scripts with the status and description of each script on the Script Maintenance screen.

<table>
<tr><th>(Contd) Step</th><th>Action</th></tr>
<tr><td>2</td><td>Locate the name of the script and check the status in the Status column.
<table>
<tr><th>IF the status is…</th><th>THEN the script is…</th></tr>
<tr><td>Active</td><td>Running</td></tr>
<tr><td>Inactv</td><td>Not running</td></tr>
</table>
CAUTION
When you restart your machine after a machine halt, you may notice that a script's status on the Script Maintenance screen is active, even though the script is not running. This is because the script was running when the machine halted and its log file is still present. In this case, you need to delete the script's log file.

WARNING
Do not delete a script log while a script is running. All the script activity, including that produced after you delete the log, is erased. For this reason, we recommend that you delete script logs only after machine halts and restarts.

Reference
See the How to Delete a Log or Journal.</td></tr>
</table>

Updating the status column

You can recheck the status by pressing F6. When you press this key, Sterling Gentran:Server scans the directory again and updates the Status column.

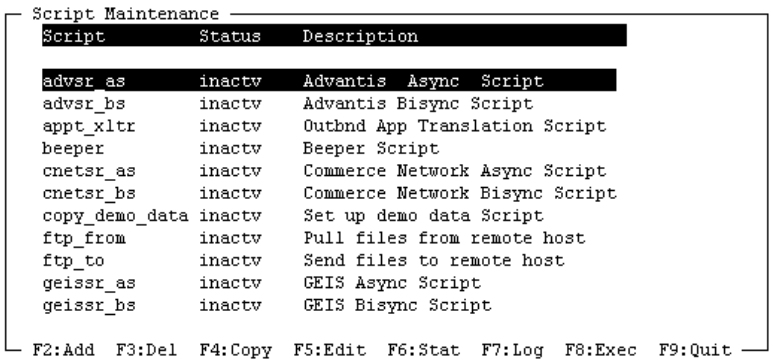
How to View a Script Log or Journal

Introduction

The Sterling Gentran:Server Screen Viewer enables you to view a script's log or journal.

Procedure

Use this procedure to view a script log or journal.

Step	Action
1	Select **Script** from the Sterling Gentran:Server host main menu. **System Response** Sterling Gentran:Server displays the Script Maintenance screen. 
2	Select the name of the script that you want to view.
3	Press F7 to display the script's log in the Screen Viewer. **System Response** If the script is running, Sterling Gentran:Server displays the script log (*script_name.l*). If the script is not running, Sterling Gentran:Server displays the script journal (*script_name.old*).
4	When you have finished, press F9 to exit.

How to Display a Script's Processing Time Statistics

Introduction

To determine or analyze the effect of a script on the overall performance of your system, you can display the script's processing time statistics with the UNIX time command. The time command runs the specified script and records the CPU time. The results tell you how much CPU time it took to process the script.

When to use

Use this procedure to:

- Study resource use in Sterling Gentran:Server
- Check the efficiency of a script.

Example

This is an example of processing time statistics for a Sterling Gentran:Server script.

The time is shown in minutes (m) and seconds (s).

```
real   0m0.31s
user   0m0.06s
sys    0m0.07s
```

Reading usage statistics

This table describes the components of the usage statistics.

Line Label	Description
real	Time elapsed during the command.
user	Time spent executing the command.
sys	Time spent in the system.

Displaying script usage statistics

Use this procedure to display processing time statistics for a Sterling Gentran:Server script.,

Step	Action
1	Go to the UNIX command line.

2	Enter the **time** command to run the script manager, **smgr**, and the script. This is the command format:
	`time smgr -s<server_script>`
	Example
	This example times the script named xltr.scr.
	`time smgr -sxltr`
	Comment
	Omit the *.scr* extension in the script's name. Do not leave a space between the -s argument and the script's name.

CAUTION

For some machines and operating systems you can substitute timex for the time command.

Maintaining Script Logs and Journals

Overview

Introduction

Script journals can take up a large amount of disk space. For this reason, you should purge the files periodically. You may also delete script logs and journals if you no longer need them.

In this section

This section contains these topics:

- How to Purge Entries from Journals
- How to Delete a Log or Journal.

How to Purge Entries from Journals

Introduction

Sterling Gentran:Server provides a command line tool called cleanlog that enables you to purge log file lines from script journals.

Note
To automate the purging of entries from journals, you can run **cleanlog** from a script and enter the script into the Permanent Schedule.

Reference
See the Working with Scripts chapter for information about creating scripts.

See the Running Scripts chapter for information about adding a script to the Permanent Schedule.

When to use

Use this procedure when you want to remove old activity from journals to free disk space.

Purging entries

The command format depends on the type of file purge you want to perform.

From the following table, choose the type of purge you want to perform and enter the command at the command line.

- Enter the name of the log in the <logname> portion of the command.
- Enter dates in the format MMDDYY or MMDDCCYY.

IF you want to purge entries…	THEN use this command…
Older than 90 days in all journals	`ksh cleanlog.sh` **Note** cleanlog.sh is a script that calls the cleanlog program.

(Contd) IF you want to purge entries…	THEN use this command…

Older than a specific number of days from all journals	`ksh cleanlog.sh <days>`
	Note
	cleanlog.sh is a script that calls the cleanlog program.
	Example
	To purge from all entries from a journal older than 60 days:
	`ksh cleanlog.sh 60`

Older than a specified number of days in a single journal	`cleanlog <logname> -a<number of days>`
	Example
	To purge entries older than 90 days from the ftp_to.scr script:
	`cleanlog ftp_to.scr -a90`

In a specific date range in a single journal	`cleanlog <logname> -B<delete _begin_ date>`
	`-E<delete_end_ date>`
	Example
	To purge from the ftp_to.scr script entries dated May 1, 1997 through May 15, 1997:
	`cleanlog ftp_to.scr -B04011997 -E04161997`

Reference
For more information, see the *Command Reference* chapter in the *IBM® Sterling Gentran:Server® for UNIX Technical Reference Guide*.

How to Delete a Log or Journal

Introduction

When you no longer need a script's log or journal, you can delete it.

When to delete a script's journal

Delete a script's journal only when you are certain that you will never need to refer to the script's activity.

Example
If you use a script for testing purposes only, you may want to delete the script's journal after you have completed your tests.

When to delete a script's log

The only time you should need to delete a script's log is after a machine halt and restart.

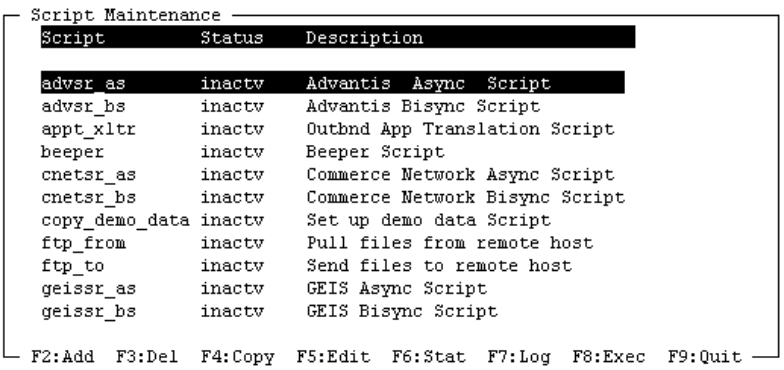
When you restart your machine, you may notice that a script's status on the Script Maintenance screen is active, even though the script is not running. This is because the script was running when the machine halted and its log file is still present. In this case, you need to delete the script's log file.

WARNING

Do not delete a script log while a script is running. All the script activity, including that produced after you delete the log, is erased. For this reason, we recommend that you delete script logs only after machine halts and restarts.

Deleting a journal or log

Use this procedure to delete a script's journal or log.

Step	Action
1	Select **Script** from the Sterling Gentran:Server host main menu. **System Response** Sterling Gentran:Server displays the Script Maintenance screen. 
2	Select the script name.
3	Press F7 to display the script log or journal. **System Response** If the script is running, Sterling Gentran:Server displays the script log (*script_name.l*). If the script is not running, Sterling Gentran:Server displays the script journal (*script_name.old*). **SUGGESTION** **Look at the name of the log to determine whether the script log or the script journal is displayed.**
4	Press F3 to delete the log or journal. **System Response** Sterling Gentran:Server deletes the log or journal and displays the Script Maintenance screen again.

CHAPTER
N

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA__95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may

vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are ficticious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2011. Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. 2011.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions

worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise, Gentran®, Gentran:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

action line	The line in a script that contains the actual commands you want carried out.
agent	A data manager.
application name map	A record that identifies the records and fields that the data manager extracts from the application transaction to build the document reference number.
archive	The process of capturing and storing a copy of a document after a data manager processes it.
audit file	The file that a data manager produces to pass archiving instructions to the archive handler.
auditing facility	The Sterling Gentran:Server facility that loads data manager event files to an auditing file, such as a relational database table, so that you can use the records for auditing purposes.
base initialization file	The default initialization file that contains all the processing parameters for all data manager personality types except the archive data manager.
category	A class or grouping of Trading Partnership records.
child process	A process that is started by and is part of another process. The other process is called the parent process.

cleanlog The command line tool that enables you to purge log file entries.

comment line A phrase or sentence in a script that explains the purpose or effect of the line of instructions that follows the comment line.

configuration record A record that describes how a data manager directs the data that it handles for a particular Trading Partnership code or file name. The record:

- Specifies the Trading Partnership code or file name that the data manager is to use to identify data
- Tells the data manager what to do with the data it has identified.

cron The UNIX system daemon that starts programs identified in the system's **crontab** at scheduled times.

crontab A UNIX system file that contains the files listing all the programs to be run by the **cron** daemon. Sterling Gentran:Server submits entries in the permanent schedule to **crontab**.

database

A collection of stored data often shared by different applications.

data manager A program that periodically scans a directory or queue for data files and then processes the files it finds. Processing can include:

- Routing data
- Invoking scripts
- Archiving data
- Handling data errors.

data manager log A record produced when a data manager handles a data set. The log contains the name of the data set, what was done with it, and Life Cycle load information.

date-time stamp The label that Sterling Gentran:Server attaches to a document to identify the date and time the document was received.

default value The value that Sterling Gentran:Server uses if you do not specify a different one.

delimiter Special characters that designate the type of information on a line in the script.

document reference number The unique number that Sterling Gentran:Server assigns to each data set or document to track the movement of the data set or document through the system.

document specifier table A collection of set ID or application name maps that specify the places in a trading partner's document that certain Sterling Gentran:Server processes use to construct the document reference number.

document specifier utility The Sterling Gentran:Server tool that enables you to specify the characters in a document that the inbound data manager, appm data manager, and the translator extract to derive the document reference number.

error handler A device in an initialization file used to specify how you want a particular error handled.

Foreground Manager The parent data manager of all other data managers. You configure data managers through the Foreground Manager (fmgr). The Foreground Manager must be running before other data managers can run.

FTP daemon The background program that controls file transfer protocol. File transfer protocol moves or copies files between computers.

functional acknowledgment (FA)	The standard transaction set used to acknowledge receipt of a transmission.
group	A set of related parameters in an initialization file.
initialization file	The configurable file that sets the data manager's personality and processing parameters.
intelligent agent	An event-driven computer program that can operate without interaction from a person at a computer terminal.
key fields	The set of fields that the Life Cycle load programs use to identify entries in a Life Cycle table. Each Life Cycle entry contains a unique value in at least one of the key fields to distinquish it from other entries.
label line	The line in a script that contains the name that you assign to the action that you want performed.
lcld	The Sterling Gentran:Server shell script that loads event records that the inbound, download, file, host command card, and archive type data managers produce to the Life Cycle table.
Life Cycle	The Sterling Gentran:Server auditing facility that enables you to load data manager event files and translation audit files to an auditing file, such as a relational database table, so that you can use the records for auditing purposes.
Life Cycle event file	The file that contains a data manager's Life Cycle event records. The name of the event file is the data manager's name with a ".v" suffix.

Life Cycle event record

A record produced when a data manager processes a file. The record contains the date, time, name, and location of the data as it is passed through the data manager.

Life Cycle load programs

The programs **lcld** and **xlld**. These programs load and update the Life Cycle table with data manager event files.

Life Cycle table

The database table that holds your audit file records. Your EDI administrator creates this the table during the Life Cycle setup process and gives it public access.

log file

A file that contains a record of process activity and messages produced by that activity.

Example
A data manager log contains a record of the data manager's status at a given date and time and any message produced.

longterm

The script that controls long-term archiving.

mail_proc file

The UNIX mail script that is used with a data manager to send messages based on the consequences of data manager operations. The mail script has the same name as the data manager.

mailbag ID

The 6-character, base-32 code that Sterling Gentran:Server generates to identify a session in which files were received and data files passed in the session.

map picture

A pictorial representation of the map for the document reference number. Each field in the map is represented with a unique symbol.

mksrvdb

The program or script that creates the database Life Cycle table.

Permanent Schedule	The Sterling Gentran:Server feature that enables you to run scripts on a specified schedule.
personality	The data manager type that determines how the data manager processes data.
PID	Process identification number. An integer value the operating system assigns to a process to distinguish it from other processes. PIDs are shown on the Data Manager Control screen and in log files.
process flow	A flow of data files from one data manager to the next. The flow may contain any number of data managers. Also, a set of parameters and commands that describes how data is moved from a source to a destination.
Process Control Manager (PCM) wizard	The Sterling Gentran:Server wizard that guides you through the process of creating a two-agent flow.
real-time processing	A system configuration that enables your system to move critical documents through the processing cycle as quickly as possible.
record file layout	The user-defined file layout of an application file.
recover script	A script that (1) examines the processing environment for any scripts that may have been active at the time your machine stopped unexpectedly (2) restarts the scripts.
script	A set of commands that controls processes or performs some action.
script directory	The directory that contains all Sterling Gentran:Server scripts.

script editor The default editor that Sterling Gentran:Server calls when you create or edit a script.

script journal The file that contains a record of a script's previous activity. When a script finishes, it appends the data in its log file to its journal.

script library A file that lists and describes all the Sterling Gentran:Server scripts that have been added to the file.

script log The file in which a script records its process activity while it is active. Each time the script starts, it creates a new log file.

Script Manager The Sterling Gentran:Server program that directs the script interpreter to execute the commands in a script.

sequence number Value that **lcId** assigns to the SEQ field when making an entry in the Life Cycle database table. The value is assigned when the table contains an entry with duplicate TP, DOC, and IOX values. The **lcId** program increments the sequence number by one to distinguish the new entry from the existing entry.

set ID map A record that identifies the segments, elements, and sub-elements that the data manager or translator extracts from the transaction set or EDI document to build the document reference number.

source agent The data manager (inbd or appm personality) that begins a Process Control Manager inbound process flow.

status The running state (active or inactive) of a script or data manager.

tracker	The Sterling Gentran:Server command line program that enables you to run a statistical report on the translation traffic from Life Cycle data.
Trading Partner record	One of the fiverecords maintained in trading partner files: Trading Partnership record, Interchange Organization record, Group Organization record, and Contact record, and TRADACOMS record.
Trading Partnership	An arrangement with a specific trading partner to exchange information in a specific document type, described by a map file.
Trading Partnership code	A user-defined code that uniquely identifies a Trading Partnership record.
Trading Partnership record	The record that contains information about one of the Trading Partnerships you have established. The record include the Trading Partnership code, the translation map to be used when translating business documents for this trading partner, and whether an acknowledgment is to be generated.
translation agent	The translation data manager (xltr personality). For Process Control Manager process flows, the translation agent is the destination in an inbound process flow.
translation audit files	The event files, *edistat.i* and *edistat.o*, that the translator produces. These files are also referred to as temporary audit files or status records.
translation script	A special Sterling Gentran:Server script associated with and invoked by a translation data manager. The script calls the translator, **lftran**, and other runtime programs.
UNIX mail script	A UNIX script that you can use to send electronic mail messages based on the results of a Sterling Gentran:Server script operation.

wizard A process that automatically presents, in order, a complete sequence of dialog boxes required to perform a task.

xlld The Sterling Gentran:Server program that updates the Life Cycle table with translation audit file and functional acknowledgment information.

10 Glossary

Index

Symbols

$EDITOR environment variable *5-30, 5-38*
./mail_proc *5-54*
./script directory *5-30*
  effect of deleting script from *5-52*
_gen_xltr.scr *3-39, 3-59*

A

action line (in a script) *5-10*
Add Queue Entry screen
  fields and functions *4-15*
  function keys *4-16*
  illustration *4-15*
administrative login ID *1-6*
APP Add screen *7-32*
  fields and functions *7-32*
  function keys *7-35*
  illustration *7-32*
APP Mapping screen
  fields and functions *7-65*
  function keys *7-67*
  illustration *7-65*
  purpose *7-65*
application description
  length of file name *1-10*
Application name map *7-24*
application name map
  adding *7-42*
  adding to a document specifier table *7-42*
  changing *7-75*
  deleting from a document specifier table *7-78*
application-to-application process flow *3-38*
application-to-standard process flow *3-38*
audit file *8-7, 9-5*

B

background process *1-8*
build (reserved word) *5-25*

C

comment delimiter
  required position in a script line *5-33*
compare (reserved word) *5-25*
configuration record
  created by PCM wizard *3-5*
copyright screen
  displaying *1-18*
cpre compiler
  requirements for Sybase database table *8-35*
cron
  enabling cron to find executable files *6-11*
crontab
  relationship to Permanent Schedule *6-11*

D

data flow administration *1-4*
  exiting *1-14, 1-18*
  starting *1-14*
data flow administrator main menu *1-12*
DATA group
  avoiding blank lines in *5-33*
  description *5-17*
  example *5-18*
  general format *5-17*
  relationship to DPROCS and PROCS groups *5-17*
  uses *5-17*
data manager
  how to select *2-22*
  length of file name *1-10*
  starting *10-16*
  starting with startserver or startserver.sh *10-22*
  stopping *10-16*
  stopping with stopserver or stopserver.sh *10-19*
  terminating without a request from fmgr *10-16*
data manager initialization file
  Life Cycle settings *8-38*
data manager log file *10-10*
  deleting *10-28*
  purging entries from *10-26*
  viewing *10-13*
data manager pattern
  length of file name *1-10*
data manager's status *10-12*
DDF name map *7-24*
  adding *7-42*
  changing *7-75*
delimiters
  role in a script *5-8*
  standard symbols in a script *5-8*

delivery agent
destination of files *2-13*
function in an inbound NCPDP process flow *3-86*
function in an inbound process flow *3-31*
function in an outbound application flow *3-51, 3-67*
in an inbound NCPDP process flow *3-89*
in an inbound process flow *3-35*
in an outbound application flow *3-38, 3-55, 3-58, 3-71*
source of files *2-12*
Delivery Agent dialog box
for inbound NCPDP process flow *3-86*
for inbound process flow *3-31*
for outbound application flow *3-51, 3-67*
directory name length *1-9*
directory tree *1-7*
displaying
copyright information *1-18*
version information *1-18*
document reference number
functions *7-5*
how it is set *7-6*
in Life Cycle event record *9-4*
overview *7-5*
picture *7-47*
purpose of mapping *7-23*
role in duplicate checking *7-5*
tasks in defining *7-11*
Document Reference Number Specifier screen
fields and functions *7-12*
function keys *7-13*
illustration *7-12*
document specifier map
changing *7-75*
document specifier table
adding *7-15*
attaching a Trading Partnership code to *7-50*
changing a Trading Partnership code attachment *7-57*
copying *7-21*
creating overview *7-14*
deleting *7-54, 7-60*
displaying *7-18*
locating by Trading Partnership code *7-19*
maintainence overview *7-56*
mapping overview *7-23*
overview *7-9*
verifying Trading Partnership code attachments *7-54*
viewing list of attached Trading Partnership codes *7-54*
ways to create *7-14*
downstream data manager
as data manager that reads from a queue *4-4*
DPROCS group
description *5-19*
example *5-19*
general format *5-19*

E

EDI Add screen
fields and functions *7-27*
function keys *7-30*
illustration *7-27, 7-39*
EDI data *3-18, 3-75*
EDI Mapping screen *7-62*
fields and functions *7-62*
function keys *7-64*
illustration *7-62*
edi_env
sample file *6-18*
edi_env file
using to control environment *5-35*
ediarc *3-45, 3-46, 3-63*
called by translation script *5-46*
electronic data interchange (EDI) *1-4*
end (reserved word) *5-24*
envelope
called by translation script *5-46*
ENVIRON group
description *5-11*
example *5-12*
general format *5-11*
environment file
creating *6-18*
sample *6-18*
setting environment variables in *6-18*
environment variables
setting for scripts run on a schedule *6-18*
Error Handling dialog box *3-93*
error handling instructions *3-95*

F

file definition
length of file name *1-10*
File name
character restrictions *1-11*
conventions *1-9*
invalid characters *1-11*
length *1-10*

validation *1-9*
file name *1-7*
File record layouts
org.dat/idx *8-21*
Flow Identification dialog box *3-14*
flow summary *3-104*
flow summary reports *3-110*
printing *3-112*
types *3-110*
foreground process *1-8*
function keys
using to start an action *1-17*
Functional Acknowledgment Due report (Life Cycle) *9-30*
Functional Acknowledgment Exception Report (Life Cycle) *9-28*
functional acknowledgments
columns updated by xlld *9-19*
fields in Life Cycle table *9-11*
how xlld handles *9-18*
how xlld locates Life Cycle record *9-19*
update logic *9-18*
xlld and *9-11*

G

group (UNIX) *1-6*
group name (in a script) *5-10*
guidelines for using *5-33*
groups (in a script) *5-9*
general format *5-9*
names and functions *5-9*

H

host operating environment *1-6*

I

if (reserved word) *5-24*
implementation guide
length of file name *1-10*
inbound process flow
creating with the PCM wizard *3-18, 3-75*
Informix Life Cycle reports
running *9-39*
Informix Life Cycle table *8-10*
initialization file
editing for Life Cycle *8-38*
parameters for Life Cycle activity *8-44*
input file
length of file name *1-10*
interchange segment *7-30*
setting Occurrence field to extract contents *7-46*
isql
required in path to run Informix Life Cycle reports *9-39*

L

label name (in a script) *5-10*
guidelines for *5-33*
lcld *9-12*
functions *9-8*
log file *9-13*
run log *9-13*
running from the command line *9-14*
versions *9-13*
lftran *3-18, 3-75*
called by translation script *5-46*
Life Cycle
audit files *9-1*
auditing facility *9-2*
configuration process *8-23*
configuring system profile for using Oracle database *8-28*
data manager initialization file settings *8-38*
event file *9-4*
event record *9-4*
how xlld handles update failures *9-21*
load programs *9-8, 9-10*
process flow diagram *8-23*
process, how lcld and xlld are called in *9-10*
records *9-4*
reports *9-24*
running overview *9-37*
setting up for Informix database table *8-32*
setting up for Sybase database table *8-35*
setting up to use Oracle database table *8-26*
using reconciliation IDs in *8-46*
wetting processing directory *8-44*
Life Cycle command
defining the command *8-45*
setting when it is run *8-45*
Life Cycle event record *3-18, 3-38, 3-39, 3-58, 3-59, 3-75*
Life Cycle file
setting contents *8-44*
Life Cycle reports
FA Due Report *9-30*
FA Exception report *9-28*
list of *9-24*
overview *9-24*
running Informix *9-39*
running Translation Traffic report *9-41*

TP List report *9-32*
Translation Activity *9-26*
Translation Traffic report *9-34*
Life Cycle table
how xlld updates *9-16*
LIFE_CYCLE_CALL_SWITCH
setting timing of Life Cycle command *8-45*
LIFE_CYCLE_DIR *9-4*
setting Life Cycle event file directory *8-44*
LIFE_CYCLE_EXEC_LINE
setting Life Cycle command *8-45*
linker
setting path to database linker *8-27*
lock file names
adding to LOCKS directory *5-45*
LOCKS
directory *5-45*
types of *5-13*
LOCKS directory
adding lock file names to *5-45*
contents *5-14*
LOCKS group
description *5-13*
execution order *5-13*
log
script *10-38*
log file
Screen View navigation keys *10-4*
screen viewer *10-4*
viewing a data manager *10-13*
login ID, administrative *1-6*

M

mail notification (of script failure) *5-6*
mail_proc file *3-95*
main menu *1-12*
map
length of file name *1-10*
map picture
displaying *7-47*
mapping table
length of file name *1-10*
menu
selecting an option from *1-14, 1-16*
mksrvdb program *9-7*
monitoring processes *10-1*
MULTIPLE_LIFE_CYCLE_FILES
setting Life Cycle file scope with *8-44*

N

NCPDP Add screen *7-39*
fields and functions *7-39*
function keys *7-41*
illustration *7-39*
NCPDP Mapping screen *7-71*
fields and functions *7-71*
NCPDP-to-application process flow *3-75*
NCPDP-to-standard process flow *3-75*
non-shareable resources
definition *5-13*

O

Occurrence field (on EDI and APP screens) *7-45*
purpose *7-45*
operating environment (of host) *1-6*
Oracle C compiler
requirement for Life Cycle *8-26*
Oracle Life Cycle table *8-14*
org.dat/idx
file record layouts *8-21*
outbound application flow
agents in a PCM *3-38*
creating with the PCM wizard *3-38*
input file names for *3-38*
output file
length of file name *1-10*

P

PATH *1-14*
path name *1-7*
PCM process flow
agents in an inbound *3-18*
agents in an inbound NCPDP *3-75*
agents in outbound *3-38*
beginning a *3-11*
creating an inbound *3-18*
creating an inbound NCPDP *3-75*
creating an outbound application *3-38*
creating supporting files for *3-13*
deleting *3-118*
deleting Trading Partnerships from *3-117*
delivery agent for inbound *3-35*
delivery agent for inbound NCPDP *3-89*
delivery agent for outbound application *3-55, 3-71*
editing *3-115*
naming and describing *3-16*
processing agent for inbound *3-29*
processing agent for inbound NCPDP *3-85*
processing agent for outbound application *3-49, 3-66*
source agent for inbound *3-23*

source agent for inbound NCPDP *3-79*
source agent for outbound application *3-43*
types *3-8*
Permanent Schedule
adding a script to *6-19*
changing a scripts schedule *6-24*
copying a script's schedule *6-21*
default environment *6-18*
deleting a script from *6-26*
function *6-11*
need for environment file *6-11*
overview *6-11*
overwriting crontab files *6-11*
Permanent Schedule Maintenance screen
fields and functions *6-12*
function keys *6-14*
illustration *6-12*
Permanent Schedule screen
entering a schedule on *6-17*
fields and functions *6-15*
illustration *6-15*
permissions (UNIX file) *1-6*
pro*c
in Oracle bin directory *8-26*
Process Control Manager wizard *3-5*
purpose *3-5*
process flow
agents in a PCM outbound application *3-38*
completing a PCM *3-92*
components of a PCM *3-7*
creating an inbound *3-18*
creating an inbound NCPDP *3-75*
creating an outbound application with the PCM wizard *3-38*
creating supporting files for a PCM *3-13*
creating with the PCM wizard *3-9*
definition of *2-4, 3-7*
deleting a PCM *3-118*
deleting Trading Partnerships from PCM *3-117*
delivery agent in an inbound flow *3-35*
delivery agent in an inbound NCPDP flow *3-89*
delivery agent in an outbound application flow *3-55, 3-71*
designing basic *2-1*
designing overview *2-19*
editing a PCM *3-114, 3-115*
error handling instructions for *3-95*
example inbound *2-14*
example of designing inbound *2-24*
example with script *2-18*
identifying purpose of *2-20*
maintaining PCM *3-114*
naming and describing a PCM *3-16*
processing agent in an inbound flow *3-29*
processing agent in an inbound NCPDP flow *3-85*
processing agent in an outbound application flow *3-49, 3-66*
source agent in an inbound flow *3-23*
source agent in an inbound NCPDP flow *3-79*
source agent in an outbound application *3-43*
two basic types of *2-4*
types of PCM flows *3-7*
worksheet *2-28*
process flow summary *3-104*
process flow summary reports *3-110*
process ID number (PID) *10-18*
process log *10-10*
processing agent
destination of files *2-12*
function in an inbound NCPDP process flow *3-85*
function in an inbound process flow *3-29*
function in an outbound application flow *3-45, 3-63*
in an inbound NCPDP process flow *3-75, 3-81*
in an inbound process flow *3-18, 3-25*
in an outbound application flow *3-38, 3-49, 3-58, 3-66*
source of files *2-12*
Processing Agent dialog box
for inbound NCPDP process flow *3-81*
for inbound process flow *3-25*
for outbound application flow *3-45, 3-63*
PROCS group
description *5-23*
putenv (reserved word) *5-25*

Q

Qualifier field (on EDI and APP screens) *7-45*
purpose *7-45*
queue
adding an entry on the Queue File screen *4-17*
adding an entry with the srvr_enq command *4-17*
benefits of using *4-5*
creating a new *4-10*
definition *4-4*
deleting *4-23*
deleting an entry using the Queue File screen *4-19*
deleting an entry with the srvr_deq command

4-19
information found in a *4-4*
length of file name *1-10*
names *4-5*
removing from the Select list *4-22*
searching for an entry in *4-14*
selecting from the queue Select screen *4-7*
viewing entries in *4-13*
Queue File screen
deleting an entry from *4-19*
fields and functions *4-8*
function keys *4-9*
illustration *4-8*
queue process *4-6*

R

reconciliation IDs
using alternates for application or download data managers *8-46*
recover script
adding to the inittab or rc initialization file *6-32*
location *6-29*
modifying *6-30*
modifying to include script name *6-30*
overview *6-29*
role in restarting scripts *6-29*
tips for using *6-29*
using to generate mail notices *6-29*
recover.scr
see recover script *6-28*
using to restart scripts *6-29*
relational data base *9-6*
release (reserved word) *5-25, 5-35*
requirements
incorporating new *2-23*
reserved words (in a script) *5-24*
build *5-25*
compare *5-25*
else *5-24*
end *5-24*
if *5-24*
putenv *5-25*
release *5-25*
then *5-24*
RESULTS group
avoiding blank lines in *5-33*
choosing steplabels for *5-28*
description *5-27*
example *5-28*
general format *5-27*
root directory *1-7*

S

schedule
changing for a script *6-24*
copying a run schedule for a script *6-21*
Screen Viewer *10-4*
function keys *10-4*
navigation keys *10-4*
Script
deleting *5-52*
script
adding a *5-38*
adding a script to the Permanent Schedule *6-19*
checking the status of *10-32*
deleting a *5-52*
description of *5-5*
displaying usage statistics for a *6-28*
guidelines for writing a *5-33*
length of file name *1-10*
processing on a schedule *6-1*
removing a script from the Permanent Schedule *6-26*
running a script from the command line *6-5*
running a script from the Script Maintenance screen *6-8*
tasks in creating a *5-32*
tips for using a *5-35*
using a script to invoke another script *6-6*
ways of starting a *5-8*
ways to run a *6-3*
script directory *5-5, 5-30*
effect of deleting script from *5-52*
script editor
set in $EDITOR environment variable *5-50*
script journal *10-31*
deleting *10-40*
purging entries from *10-38*
viewing *10-34*
script journal file *10-30*
script library
contents *5-31*
script log *10-30*
erasing *10-40*
viewing *10-34*
script log file *10-30*
deleting *10-40*
maintaining overview *10-37*
Script Maintenance screen
fields and functions *5-36*
function keys *5-37*
illustration *5-36*
running a script from *6-8*

Script Manager
  overview *5-6*
script name
  effect of deleting from the library list *5-52*
script usage statistics
  viewing *10-35*
scripts
  monitoring overview *10-29*
Select list
  removing a queue from *4-22*
Select screen (for queues)
  function keys *4-7*
  illustration *4-7*
Set ID map *7-23*
  adding *7-42*
  changing *7-75*
  deleting from a document specifier table *7-78*
shell
  definition of *1-8*
source agent
  destination of files *2-12*
  in an inbound NCPDP process flow *3-75, 3-79*
  in an inbound process flow *3-18, 3-23*
  in an outbound application flow *3-38, 3-43, 3-58*
  source of files *2-12*
Source Setup dialog box
  for inbound NCPDP process flow *3-77*
  for inbound process flow *3-20*
  for outbound application flow *3-40, 3-60*
sql compiler
  requirements for Informix Life Cycle table *8-32*
SQL error *8-25*
srvr_deq command
  arguments *4-20*
  deleting a queue entry with *4-19*
  format *4-19*
srvr_enq command
  adding entries to a queue with *4-17*
  arguments *4-18*
  format *4-17*
standard-to-application process flow *3-18*
standard-to-standard process flow *3-18*
startserver
  functions *10-22*
  when to use *10-22*
startserver.sh
  functions *10-22*
  script recovery *10-22*
  when to use *10-22*
status
  checking data manager *10-17*
  viewing a data manager's *10-12*
status field (on Data Manager Control screen) *10-18*
STEPS group
  description *5-20*
  example *5-20*
  general format *5-20*
  relationship to PROCS group *5-20*
stopserver
  function *10-19*
  when to use *10-19*
stopserver.sh
  function *10-19*
  when to use *10-19*
supporting files
  creating for a PCM process flow *3-12*
Sybase Life Cycle table *8-18*

T

then (reserved word) *5-24*
third-party products *1-8*
tracker command
  arguments *9-41*
  command line format *9-41*
  running Translation Traffic report with *9-41*
tracker program *9-34*
tracking data (with Life Cycle) *9-1*
Trading Partner List report (Life Cycle) *9-32*
Trading Partner Records dialog box *3-97*
transaction register
  length of file name *1-10*
Translation Activity report (Life Cycle) *9-26*
Translation Options dialog box *3-27, 3-47, 3-64, 3-83*
translation script *3-18, 3-38, 3-45, 3-58, 3-75*
  creating *5-47*
  definition *5-46*
  Life Cycle activities *9-15*
  naming convention for *5-46*
Translation Traffic report (Life Cycle) *9-34*
translator *3-18, 3-38, 3-49, 3-66, 9-10*
  called by translation script *5-46*
  relationship to Data Flow Administration *1-5*
translator program lftran *3-75*

U

UNIX directory structure *1-7*
UNIX file permissions *1-6*
UNIX mail script
  adding to a script *5-55*
  deleting *5-57*

editing *5-55*
example of *5-54*
location *5-54*
overview *5-54*
using with scripts *5-54*
UNIX operating environment *1-6*
UNIX user group *1-6*
upstream data manager
as data manager that writes to a queue *4-4*
USE_RECON_IDS
setting Life Cycle values *8-46*

V

variables
defining in a script *5-11*
guidelines for using in a script *5-34*
referencing in a script *5-12*
version
displaying *1-18*
VT100 terminal emulator
function key emulation *1-17*

W

WORK_DIRECTORY
setting for translation data manager *5-49*

X

XL_MODEL_SCR
setting for translation data manager *5-49*
using to specify name of translation script *5-46*
XL_RUN_DIR
setting for translation data manager *5-49*
xlld *9-10*
called by translation script *5-46*
command line format *9-23*
Life Cycle columns updated by *9-16*
log file *9-21, 9-22*
overview *9-15*
running from the command line *9-23*
running from translation scripts *9-15*
types of updates *9-17*
viewing log file *9-22*
ways to call *9-15*
xlld.pc
checking revision number *8-26*
XML Add screen *7-36*
fields and functions *7-36*
function keys *7-38*
illustration *7-36*
XML Mapping screen
fields and functions *7-68*
function keys *7-69*
purpose *7-68*