

# **Gentran:Server® for UNIX With EC Workbench**

## **Data Flow Administration Guide**

Version 6.1

***Sterling Commerce***  
*An IBM Company*

# Copyright Notice

**March 2011**

Licensed Materials - Property of Sterling Commerce©

Copyright Sterling Commerce, an IBM Company 1991-2011. All Rights Reserved

US Government Users Restricted Rights - Use, duplication or disclosure restricted by contract with Sterling Commerce

## **Sterling Commerce Software**

### **Trade Secret Notice**

THE GENTRAN:SERVER FOR UNIX SOFTWARE ("STERLING COMMERCE SOFTWARE") IS THE CONFIDENTIAL AND TRADE SECRET PROPERTY OF STERLING COMMERCE, INC., ITS AFFILIATED COMPANIES OR ITS OR THEIR LICENSORS, AND IS PROVIDED UNDER THE TERMS OF A LICENSE AGREEMENT. NO DUPLICATION OR DISCLOSURE WITHOUT PRIOR WRITTEN PERMISSION. RESTRICTED RIGHTS.

This documentation, the Sterling Commerce Software it describes, and the information and know-how they contain constitute the proprietary, confidential and valuable trade secret information of Sterling Commerce, Inc., its affiliated companies or its or their licensors, and may not be used for any unauthorized purpose, or disclosed to others without the prior written permission of the applicable Sterling Commerce entity. This documentation and the Sterling Commerce Software that it describes have been provided pursuant to a license agreement that contains prohibitions against and/or restrictions on their copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright notice.

As and when provided to any governmental entity, government contractor or subcontractor subject to the FARs, this documentation is provided with RESTRICTED RIGHTS under Title 48 CFR 52.227-19. Further, as and when provided to any governmental entity, government contractor or subcontractor subject to DFARs, this documentation and the Sterling Commerce Software it describes are provided pursuant to the customary Sterling Commerce license, as described in Title 48 CFR 227-7202 with respect to commercial software and commercial software documentation.

These terms of use shall be governed by the laws of the State of Ohio, USA, without regard to its conflict of laws provisions. If you are accessing the Sterling Commerce Software under an executed agreement, then nothing in these terms and conditions supersedes or modifies the executed agreement.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies. Gentrans and Gentrans:Server are registered trademarks of Sterling Commerce, Inc.

### **Third Party Software**

Portions of the Sterling Commerce Software may include products, or may be distributed on the same storage media with products, ("Third Party Software") offered by third parties ("Third Party Licensors"). Portions of this software are copyrighted by Data Direct Technologies, Inc. 1991-2002.

**Warranty Disclaimer**

This documentation and the Sterling Commerce Software which it describes are licensed either "AS IS" or with a limited warranty, as set forth in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

The Third Party Software is provided 'AS IS' WITHOUT ANY WARRANTY AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. FURTHER, IF YOU ARE LOCATED OR ACCESSING THIS SOFTWARE IN THE UNITED STATES, ANY EXPRESS OR IMPLIED WARRANTY REGARDING TITLE OR NON-INFRINGEMENT ARE DISCLAIMED.



# Table of Contents

## About This Guide

- ▶ Welcome ..... ii
- ▶ Contents of Chapters ..... iii
- ▶ Related Publications ..... v
- ▶ Documentation Conventions ..... viii

## Understanding the Basics

- ▶ Overview ..... 1-2
- ▶ Introduction ..... 1-2
- ▶ The Data Flow Administration Features ..... 1-4
- ▶ Operating Environment ..... 1-6
- ▶ File Name Conventions ..... 1-9

### Basic Procedures

- ▶ 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 Use a Shortcut Key to Initiate an Action ..... 1-18
- ▶ How to Display Version and Copyright Information ..... 1-19

### Miscellaneous Procedures

- ▶ How to Search For a Trading Partnership Record ..... 1-20

## Designing Basic Process Flows

### Overview

- ▶ Introduction ..... 2-2
- ▶ Process Flows ..... 2-5
- ▶ Data Managers ..... 2-7
- ▶ Example Inbound Process Flow ..... 2-11
- ▶ Scripts ..... 2-13
- ▶ Example Process Flow With Script ..... 2-14

### Designing a Process Flow

- ▶ Overview ..... 2-15
- ▶ How to Identify a Flow's Purpose ..... 2-16

- ▶ How to Select Data Managers ..... 2-18
- ▶ How to Add Other Components ..... 2-22
- ▶ Example: Designing an Inbound Process Flow ..... 2-23
- ▶ Inbound XML Process Flows ..... 2-26
- ▶ Process Flow Worksheet ..... 2-28

### **Stages in Basic Processes**

- ▶ Inbound Processing ..... 2-32
- ▶ Outbound Processing ..... 2-35

### **Building a Process Flow**

- ▶ The Flow of Work ..... 2-38

## 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) ..... 3-63
- ▶ Processing Agent Dialog Box (XML Flow) ..... 3-65
- ▶ Translation Options Dialog Box (XML Flow) ..... 3-67
- ▶ How to Set Up the Processing Agent (XML Flow) ..... 3-69
- ▶ Delivery Agent Dialog Box (XML Flow) ..... 3-71
- ▶ How to Set Up the Delivery Agent (XML Flow) ..... 3-75

## Creating an Inbound NCPDP Flow

- ▶ Overview ..... 3-79
- ▶ Source Dialog Box (Inbound NCPDP Flow) ..... 3-81
- ▶ How to Set Up the Source Agent (Inbound NCPDP Flow) ..... 3-84
- ▶ Processing Dialog Box (Inbound NCPDP Flow) ..... 3-86
- ▶ Translation Options Dialog Box ..... 3-88
- ▶ How to Set Up the Processing Agent (Inbound NCPDP Flow) ..... 3-90
- ▶ Delivery Dialog Box (Inbound NCPDP Flow) ..... 3-91
- ▶ How to Set Up the Delivery Agent (Inbound NCPDP Flow) ..... 3-94

## Completing a Flow

- ▶ Overview ..... 3-97
- ▶ Error Handling Dialog Box ..... 3-98
- ▶ How to Set Up Error Handling Instructions ..... 3-100
- ▶ Trading Partner Records Dialog Box ..... 3-102
- ▶ How to Add Trading Partnership Records to the Flow ..... 3-104
- ▶ How to Delete Trading Partnerships from the Trading Partner Records Dialog Box ... 3-107

## Using Flow Summaries

- ▶ The Flow Summary ..... 3-109
- ▶ Flow Summary Views ..... 3-111
- ▶ How to Expand and Collapse the Flow View ..... 3-114
- ▶ Flow Summary Reports ..... 3-115
- ▶ How to Print Flow Summary Reports ..... 3-117

## Maintaining Process Flows

- ▶ Overview ..... 3-119
- ▶ How to Edit a Process Flow ..... 3-120
- ▶ How to Delete Trading Partnerships From the Flow ..... 3-122
- ▶ How to Delete a Flow ..... 3-124

## Working with Data Managers

### Overview

- ▶ Introduction ..... 4-2

▶ Data Managers .....	4-5
▶ The Foreground Manager .....	4-9
<b>Configuring Data Managers</b>	
▶ The Flow of Work .....	4-10
▶ Data Manager Control Screen .....	4-12
▶ Data Manager Configuration Screen .....	4-17
<b>Procedures</b>	
▶ How to Copy a Data Manager .....	4-19
▶ How to Edit a Data Manager's Process Fields .....	4-22
▶ How to Delete a Data Manager .....	4-25
<b>Miscellaneous</b>	
▶ Using a Host Command Card Data Manager .....	4-28

## Maintaining Initialization Files

### Overview

▶ Introduction .....	5-3
▶ Data Manager Initialization Files .....	5-6

### Initialization File Structure

▶ Overview .....	5-8
▶ Parameters Section .....	5-10
▶ Errors Section .....	5-14
▶ Standard Error Types .....	5-16
▶ Handlers Section .....	5-19

### Parameters Subsections

▶ Overview .....	5-22
▶ Parameters and Personality Table .....	5-23
▶ Routing Parameters .....	5-29
▶ Line Manager Parameters .....	5-35
▶ Translator Parameters .....	5-36
▶ Miscellaneous Parameters .....	5-38
▶ Archive Handling Parameters .....	5-52
▶ Duplicates Checking Parameters .....	5-53

### Maintaining Data Manager Initialization Files

▶ Overview .....	5-54
▶ How to Access and Edit an Initialization File .....	5-55
▶ How to Copy or Delete an Initialization File .....	5-58

### Configuring Initialization Files for Specific Functions

▶ Overview .....	5-59
▶ How to Set the Destination File Name .....	5-60
▶ How to Set Error Notification .....	5-63



- ▶ How to Use a UNIX Mail Script to Send Messages ..... 5-67
- ▶ How to Use an Error Class ..... 5-71
- ▶ How to Control Concurrent Processing With File Locking ..... 5-73
- ▶ The File Locking Process ..... 5-75

### **Configuring for Real-time Processing**

- ▶ Overview ..... 5-77
- ▶ The Stages of Real-Time Processing ..... 5-78
- ▶ How to Configure Data Managers for Real-Time Processing ..... 5-82
- ▶ How to Configure the FTP Daemon to Notify a Data Manager ..... 5-84

## **Working with Configuration Records**

### **Overview**

- ▶ Introduction ..... 6-3
- ▶ Configuration Records ..... 6-5
- ▶ Ways to Create Configuration Records ..... 6-6

### **Creating Records Manually**

- ▶ The Data Manager Configuration Screen ..... 6-8
- ▶ How to Manually Create a Configuration Record ..... 6-13
- ▶ How to Open a Configuration Record ..... 6-15
- ▶ How to Copy a Configuration Record ..... 6-17

### **Using Patterns**

- ▶ Introduction ..... 6-19
- ▶ Token List ..... 6-22
- ▶ Categories in Patterns ..... 6-24
- ▶ Data Manager Pattern Configuration Screen ..... 6-25
- ▶ The Pattern and Meld Tasks ..... 6-30
- ▶ How to Create a New Pattern ..... 6-31
- ▶ How to Copy a Pattern ..... 6-37
- ▶ How to Add Another Data Manager to a Pattern ..... 6-45
- ▶ How to Add a New Pattern Configuration Record ..... 6-47
- ▶ How to Copy a Pattern Configuration Record In a Pattern ..... 6-51
- ▶ How to Edit a Pattern Configuration Record ..... 6-54
- ▶ How to Delete a Pattern ..... 6-56
- ▶ How to Use the Meld Function ..... 6-57

### **Using the Inbound Data Manager**

- ▶ How the Inbound Data Manager Creates Records ..... 6-61
- ▶ How to Have the Inbound Data Manager Create Records ..... 6-63

### **Maintaining Configuration Records**

- ▶ How to Verify Creation of Configuration Records ..... 6-66
- ▶ How to Edit a Configuration Record ..... 6-67

- ▶ How to Delete a Single Configuration Record . . . . . 6-69
- ▶ How to Delete a Range of Configuration Records . . . . . 6-70

## Using Queues

### Overview

- ▶ Introduction . . . . . 7-2
- ▶ Queues . . . . . 7-4
- ▶ The Queue Process . . . . . 7-6
- ▶ Queue Select Screen . . . . . 7-7
- ▶ Queue File Screen . . . . . 7-8

### Configuring Gentrans:Server to Use a Queue

- ▶ The Flow of Work . . . . . 7-10

### How to Create a Queue

- ▶ How to Configure the Downstream Data Manager . . . . . 7-13
- ▶ How to Enter Queue Data Into a Configuration Record . . . . . 7-16
- ▶ How to Configure the Upstream Data Manager . . . . . 7-18
- ▶ Maintaining Queue Entries . . . . . 7-21
- ▶ Overview . . . . . 7-21
- ▶ How to View Entries in a Queue . . . . . 7-22
- ▶ Add Queue Entry Screen . . . . . 7-24
- ▶ How to Add an Entry to a Queue . . . . . 7-26
- ▶ How to Delete an Entry From a Queue . . . . . 7-28

### Maintaining Queues

- ▶ Overview . . . . . 7-30
- ▶ How to Remove a Queue from the Select List . . . . . 7-31
- ▶ How to Delete a Queue . . . . . 7-32

## Archiving Your Data

### Overview

- ▶ Introduction . . . . . 8-2
- ▶ Archiving . . . . . 8-4
- ▶ The Archive Handler . . . . . 8-6
- ▶ Flow of Data Through Archive Handler . . . . . 8-8
- ▶ The Archiving Process . . . . . 8-9

### Configuring Your System to Archive Data

- ▶ Overview . . . . . 8-10
- ▶ How to Configure a Data Manager to Archive Data . . . . . 8-11
- ▶ How to Set a Configuration Record to Archive Data . . . . . 8-12
- ▶ How to Configure an Archive Handler . . . . . 8-13

### Managing Your Archives

» Overview .....	8-15
» The cl_arch Program .....	8-16
» The longterm Script .....	8-17
» How to Edit the Longterm Script .....	8-19
» How to Manually Purge Your Archives .....	8-22

### **Retrieving Archived Files**

» Overview .....	8-25
» The Retrieval Process .....	8-26
» The Archive Query Screen .....	8-27
» The Archive List Screen .....	8-29
» How to Select and Retrieve Archived Files .....	8-31
» The Retrieval Log .....	8-34
» How to Display the Retrieval Log .....	8-35

## **Working with Scripts**

### **Overview**

» Introduction .....	9-3
» Gentran:Server Scripts .....	9-5
» The Script Manager .....	9-6
» Ways to Use Scripts .....	9-7

### **Parts of a Gentran:Server Script**

» Introduction .....	9-8
» ENVIRON Group .....	9-11
» LOCKS Group .....	9-13
» DATA Group .....	9-17
» DPROCS Group .....	9-19
» STEPS Group .....	9-20
» PROCS Group .....	9-23
» RESULTS Group .....	9-27

### **Creating Gentran:Server Scripts**

» Overview .....	9-30
» The Flow of Work .....	9-32
» Guidelines for Writing Scripts .....	9-33
» The Script Maintenance Screen .....	9-36

### **Procedures**

» How to Add a Script with the Script Editor .....	9-38
» How to Add a Script with Another Editor .....	9-41
» How to Copy a Script .....	9-43
» How to Add Lock File Names to the LOCKS Directory .....	9-45

### **Working With Translation Scripts**

▶ Translation Scripts .....	9-46
▶ How to Create a Translation Script .....	9-47
<b>Maintaining Gentran:Server Scripts</b>	
▶ How to Edit a Script with the Script Editor .....	9-50
▶ How to Delete a Gentran:Server Script .....	9-52
<b>Working with UNIX Mail Scripts</b>	
▶ Overview .....	9-54
▶ How to Add or Edit a UNIX Mail Script .....	9-55
▶ How to Delete a UNIX Mail Script .....	9-57

## Running Scripts

### Overview

▶ Introduction .....	10-2
----------------------	------

### Running Scripts

▶ Overview .....	10-3
▶ How to Have a Data Manager Run a Script after Processing .....	10-5
▶ How to Have a Data Manager Run a Script for an Error .....	10-7
▶ How to Run a Script from the Command Line .....	10-9
▶ How to Run a Script from Another Script .....	10-10
▶ How to Run a Script from the Script Maintenance Screen .....	10-12

### Running Scripts on a Schedule

▶ Overview .....	10-14
▶ The Permanent Schedule .....	10-15
▶ The Permanent Schedule Maintenance Screen .....	10-16
▶ The Permanent Schedule Screen .....	10-19
▶ How to Create an Environment File .....	10-22
▶ How to Add a Script to the Permanent Schedule .....	10-23
▶ How to Copy a Schedule .....	10-25
▶ How to Change a Script's Processing Schedule .....	10-28
▶ How to Remove a Script From the Permanent Schedule .....	10-30

### Restarting Scripts Automatically

▶ Overview .....	10-32
▶ The Recover Script .....	10-33
▶ How to Make a Script Restart Automatically .....	10-34

## Defining the Document Reference Number

### Overview

▶ Introduction .....	11-3
▶ The Document Reference Number .....	11-5
▶ How the Document Reference Number is Set .....	11-6

▶ Document Specifier Tables .....	11-10
<b>Defining Document Reference Numbers</b>	
▶ Overview .....	11-12
▶ Document Reference Number Specifier Screen .....	11-13
<b>Creating a Document Specifier Table</b>	
▶ Overview .....	11-15
▶ How to Add a Document Specifier Table .....	11-16
▶ How to Display a Document Specifier Table .....	11-19
▶ How to Copy a Document Specifier Table .....	11-22
<b>Mapping Document Specifier Tables</b>	
▶ Overview .....	11-24
▶ EDI Add Screen .....	11-28
▶ APP Add Screen .....	11-33
▶ XML Add Screen .....	11-37
▶ NCPDP Add Screen .....	11-40
▶ How to Add a Map to a Document Specifier Table .....	11-43
▶ Using the Qualifier and Occurrence Fields .....	11-46
▶ How to Display a Map Picture .....	11-48
<b>Attaching a Trading Partnership Code to a Table</b>	
▶ How to Attach a Trading Partnership Code to a Table .....	11-51
▶ How to Verify Trading Partnership Code Attachments .....	11-55
<b>Maintaining Document Reference Number Tables</b>	
▶ Overview .....	11-57
▶ How to Change a Trading Partnership Code and Table Attachment .....	11-58
▶ How to Remove a Trading Partnership Code from a Table .....	11-59
▶ How to Delete a Document Specifier Table .....	11-61
<b>Maintaining Document Specifier Maps</b>	
▶ Overview .....	11-62
▶ EDI Mapping Screen .....	11-63
▶ APP Mapping Screen .....	11-66
▶ XML Mapping Screen .....	11-69
▶ NCPDP Mapping Screen .....	11-72
▶ How to Display a Document Specifier Map .....	11-74
▶ How to Change a Document Specifier Map .....	11-76
▶ How to Delete a Map from a Table .....	11-79

## Using Transaction Registers to Track Documents

### Overview

▶ Introduction .....	12-2
▶ Transaction Registers .....	12-3

- ▶ Transaction Register Screen ..... 12-6
- ▶ Transaction Register Search Screen ..... 12-8

### **Working with a Single Transaction Register**

- ▶ How to Search for an Entry ..... 12-9
- ▶ How to Add an Entry ..... 12-11
- ▶ How to Delete an Entry ..... 12-13
- ▶ How to Delete Multiple Entries ..... 12-14

### **Working with Multiple Transaction Registers**

- ▶ How to Add a Transaction Register ..... 12-18
- ▶ How to Delete a Transaction Register ..... 12-20

## Setting Up Life Cycle

### **Overview**

- ▶ Introduction ..... 13-2
- ▶ Database Software ..... 13-4
- ▶ Life Cycle Components ..... 13-6

### **Life Cycle Tables**

- ▶ Life Cycle Table lc221 ..... 13-8
- ▶ Life Cycle Table: Informix ..... 13-10
- ▶ Life Cycle Table: Oracle ..... 13-14
- ▶ Life Cycle Table: Sybase ..... 13-18
- ▶ LCDestInfo Table ..... 13-22

### **Life Cycle Configuration**

- ▶ Overview ..... 13-24
- ▶ Life Cycle Configuration Process ..... 13-25
- ▶ How to Set Up Life Cycle for Oracle ..... 13-28
- ▶ How to Set Up Life Cycle for Informix ..... 13-34
- ▶ How to Set Up Life Cycle for Sybase ..... 13-37
- ▶ Data Manager Settings for Life Cycle Activities ..... 13-40

### **Testing Life Cycle Setup**

- ▶ How to Test Outbound Processing ..... 13-43
- ▶ How to Test Inbound Processing ..... 13-45
- ▶ Troubleshooting Life Cycle ..... 13-47

## Tracking Data with Life Cycle Files

### **Overview**

- ▶ Introduction ..... 14-3
- ▶ Understanding Life Cycle Records ..... 14-5
- ▶ The Life Cycle Facility ..... 14-8
- ▶ The Life Cycle Table ..... 14-9

▶ LCDestInfo Table .....	14-10
<b>The Life Cycle Load Programs</b>	
▶ Overview .....	14-11
▶ The lclid and xlld Life Cycle Process .....	14-13
▶ The lclid Program .....	14-15
▶ How to Run lclid from the Command Line .....	14-17
▶ The xlld Program .....	14-18
▶ How xlld Updates the Life Cycle Table .....	14-19
▶ How xlld Updates the ODBC LCDestInfo Table .....	14-21
▶ How xlld Handles Functional Acknowledgments .....	14-23
▶ How xlld Handles Update Failures .....	14-26
▶ The xlld Log File .....	14-27
▶ How to Run xlld from the Command Line .....	14-28
▶ How Life Cycle Handles Inbound Database Audit Records .....	14-30
<b>Life Cycle Reports</b>	
▶ Overview .....	14-31
▶ The Translation Activity Report .....	14-33
▶ The FA Exception Report .....	14-35
▶ The FA Due Report .....	14-37
▶ The TP List Report .....	14-39
▶ The Translation Traffic Report .....	14-41
<b>Running Life Cycle Reports</b>	
▶ Overview .....	14-44
▶ How to Run Informix Life Cycle Reports .....	14-46
▶ How to Run Oracle Life Cycle Reports .....	14-48
▶ How to Run the Translation Traffic Report .....	14-49
<b>Monitoring Processes</b>	
<b>Overview</b>	
▶ Introduction .....	15-3
▶ The Screen Viewer .....	15-5
<b>Monitoring Data Manager Processes</b>	
▶ Overview .....	15-7
<b>Data Manager Control Screen</b>	
▶ Data Manager Log Files .....	15-11
▶ How to Check a Data Manager's Status .....	15-13
▶ How to View a Data Manager's Log File .....	15-14
<b>Starting and Stopping Data Managers</b>	
▶ Overview .....	15-16
▶ How to Use the Data Manager Control Screen .....	15-17

▶ How to Stop Data Managers .....	15-20
▶ How to Start Data Managers .....	15-23
<b>Maintaining Data Manager Log Files</b>	
▶ Overview .....	15-26
▶ How to Purge Data Manager Log Entries .....	15-27
▶ How to Delete a Data Manager's Log File .....	15-29
<b>Monitoring Scripts</b>	
▶ Overview .....	15-30
▶ Script Logs and Journals .....	15-31
▶ How to Check the Status of a Script .....	15-33
▶ How to View a Script Log or Journal .....	15-35
▶ How to Display a Script's Processing Time Statistics .....	15-36
<b>Maintaining Script Logs and Journals</b>	
▶ Overview .....	15-38
▶ How to Purge Entries from Journals .....	15-39
▶ How to Delete a Log or Journal .....	15-41

Glossary

Index



# About This Guide

---

## Contents

- ▶ Welcome ..... ii
  - ▶ Contents of Chapters ..... iii
  - ▶ Related Publications ..... v
  - ▶ Documentation Conventions ..... viii
-

# Welcome

Welcome to the *Gentran:Server for UNIX with EC Workbench Data Flow Administration Guide*.

---

## Who should use this guide

The *Data Flow Administration Guide* is for Gentran:Server users who develop, manage, and monitor your organization's electronic data interchange operations.

---

## In this guide

This manual:

- Introduces you to Gentran:Server data flow administration concepts
- Suggests ways to construct your organization's electronic data interchange (EDI) operations
- Shows you how to use Gentran:Server's process flow facilities to control the flow of your EDI data
- Explains how to monitor your EDI processes.

---

## In this preface

This preface:

- Describes the set of Gentran:Server user's 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 *Gentran:Server for UNIX with EC Workbench Data Flow Administration Guide* contains 15 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, Designing Basic Process Flows	Presents an overview of designing process flows.
Chapter 3, Creating a Flow with the PCM Wizard	Describes the procedures for using the Process Control Manager to create and maintain simple process flows.
Chapter 4, Working With Data Managers	Explains how to create and maintain data managers, which are also known as intelligent agents.
Chapter 5, Maintaining Initialization Files	Explains how to create and maintain data manager initialization files.
Chapter 6, Working With Configuration Records	Explains how to create and maintain configuration records.
Chapter 7, Using Queues	Describes queues and explains how to use them in your system.
Chapter 8, Archiving Your Data	Explains how to: <ul style="list-style-type: none"> <li>▶ Configure your system to archive data</li> <li>▶ Maintain your archived files</li> <li>▶ Retrieve archived data.</li> </ul>
Chapter 9, Working With Scripts	Describes the components of Gentran:Server scripts and explains how to create, maintain, and use scripts to control processes and carry out commands in your data flows.
Chapter 10, Running Scripts	Explains how to run scripts, monitor the results, and restart scripts automatically after a machine halt.

(Continued on next page)

<b>(Contd) Chapter</b>	<b>Contents</b>
Chapter 11, Defining the Document Reference Number	Describes how to specify the characters in a document that the inbound data manager, the appm data manager, and the translator use to determine the document reference number.
Chapter 12, Using Transaction Registers to Track Documents	Describes transaction registers and explains how to work with them to track documents as they pass through Gentran:Server.
Chapter 13, Setting Up Life Cycle	Explains how to configure your system to use the Gentran:Server Life Cycle feature. Life Cycle enables you to load Gentran:Server event files to a relational data base.
Chapter 14, Tracking Data With Life Cycle Files	Explains how to use the Gentran:Server Life Cycle feature to load process event records to an optional relational database.
Chapter 15, Monitoring Processes	Explains how to: <ul style="list-style-type: none"><li>▶ View data manager log files</li><li>▶ Maintain data manager log files</li><li>▶ Monitor script processes</li><li>▶ Maintain script logs and journals.</li></ul>

---

## Related Publications

### Gentran:Server documentation

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

Document	Description
Upgrade and Data Conversion Guide	Instructions for upgrading from previous versions of Gentran:Server Workstation and Gentran:Server for UNIX. Also includes instructions for converting the files that are part of the upgrade.
Installation Checklist	Description of the recommended sequence in which you should install and configure system components.
Gentran:Server for UNIX Getting Started Guide	Instructions for installing the Gentran:Server software and performing setup tasks, such as setting up security.  Instructions for starting and exiting Gentran:Server and for setting preferences and default values. Also includes instructions for checking files in and out and saving files.
Application Integration User's Guide	Instructions for performing mapping and translation tasks using the Gentran:Server Application Integration Mapper.
Mapping and Translation Guide	Instructions for performing mapping and translation tasks using the Gentran:Server Visual Mapper.  <b>Note</b> This guide is provided only if you maintain maps created with Gentran:Server version 5.3 or prior.
NCPDP User's Guide	Instructions for mapping and translating NCPDP files with the Application Integration system.
XML User's Guide	Instructions for mapping and translating XML files with the Application Integration system.  <b>Note</b> This guide is provided only if your organization has the Gentran:Server XML translation option.  (Continued on next page)

<b>(Contd) Document</b>	<b>Description</b>
ODBC User's Guide	<p>Instructions for mapping and translating ODBC files with the Application Integration system.</p> <p><b>Note</b> This guide is provided only if your organization has the Gentran:Server ODBC translation option.</p>
GENCOD User's Guide	<p>Instructions for mapping and translating GENCOD files with the Application Integration system and the Visual Mapper.</p>
VDA User's Guide	<p>Instructions for mapping and translating VDA files with the Application Integration system and the Visual Mapper.</p>
Technical Reference Guide	<p>Describes processes, lists command-line commands in alphabetical order, and describes file record layouts and data type formats.</p>
Maintenance and Troubleshooting Guide	<p>Instructions for maintaining your Gentran:Server installation. Also provides troubleshooting information to help determine the cause and solution of problems that may occur.</p>
Advanced Data Distribution Guide	<p>Instructions for configuring and using the Gentran:Server Advanced Data Distribution product.</p> <p><b>Note</b> This guide is provided only if you have Gentran:Server with Advanced Data Distribution.</p>
FTP Daemon User's Guide	<p>Instructions for configuring and using the FTP Daemon tool with the Advanced Data Distribution product.</p>
Online Help	<p>Context-sensitive help screens describing the Gentran:Server dialog boxes for the mapping and translation features. Also includes procedures for using the mapping and translation and the data flow administration software.</p>
Readme file	<p>Information about recent enhancements included with this software release. This file is in the <i>/readme</i> directory on the Windows client computer.</p>

**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 Gentran:Server Life Cycle audit tracking facility  <b>Note:</b> Support for Oracle databases does not include support for the Oracle Exadata platform.	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. <b>Example</b> <i>Gentran: Server for UNIX and Workstation Application and Integration User's Guide</i>
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. <b>Example</b> A <b>password</b> 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. <b>Example</b> <i>&lt;scriptname&gt;.scr</i>

(Continued on next page)



## 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.  <b>Example</b> 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.  <b>Example</b> {-f <filename>} means you must enter the f parameter followed by a filename.
[ ]	Brackets indicate an optional part of a statement. Do not enter the brackets.  <b>Example</b> [-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.  <b>Example</b> -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.  <b>Example</b> (n) means that you should type a number enclosed by parentheses.



---

# Understanding the Basics

---

<b>Contents</b>	<b>Overview</b>
	▶ Introduction ..... 2
	▶ The Data Flow Administration Features ..... 4
	▶ Operating Environment ..... 6
	▶ File Name Conventions ..... 9
	<b>Basic Procedures</b>
	▶ The Data Flow Administration Main Menu ..... 12
	▶ How to Start and Exit Data Flow Administration ..... 14
	▶ How to Select an Option From a Menu or List ..... 16
	▶ How to Use Function Keys to Initiate an Action ..... 17
	▶ How to Use a Shortcut Key to Initiate an Action ..... 18
	▶ How to Display Version and Copyright Information ..... 19
	<b>Miscellaneous Procedures</b>
	▶ How to Search For a Trading Partnership Record ..... 20

---

## Overview

## Introduction

**In this chapter** This chapter contains basic information about the 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 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.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
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 <i>host</i> .
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.  <b>Example</b> The CTRL + A sequence accesses the UNIX command line.
Trading Partnership search	The 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**      Gentran:Server Data Flow Administration software is a comprehensive product that centralizes your organization's **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 the Gentran:Server Data Flow Administration software 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**      Gentran:Server's Data Flow Administration software is easy to use. You do not need programming knowledge to accomplish most tasks. However, to fully understand and use Data Flow Administration's capabilities, you should be familiar with the UNIX operating system and UNIX scripting commands.

---

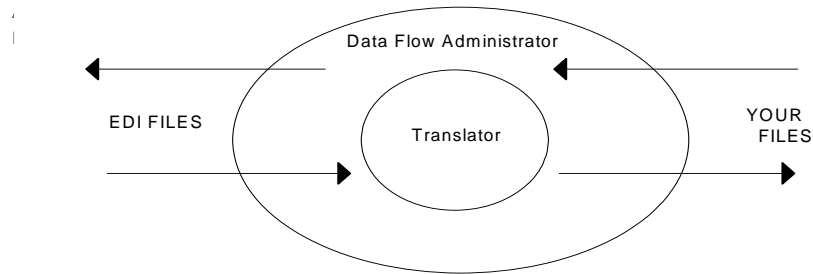
**Connects networks**      Gentran:Server's Data Flow Administration is the software product that connects separate data communications networks. It passes information between networks and processes data to prepare it for Gentran:Server's translator.

---

(Continued on next page)

---

**Illustration** This illustration shows the relationship between the Gentran:Server Data Flow Administration and the translator.



**Note**

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

---

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

---

# Operating Environment

---

## Introduction

Gentran:Server's Data Flow Administration software 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 Gentran:Server administrative login ID is usually the only member of the 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 *Gentran:Server for UNIX Getting Started Guide* for information about creating an administrative user login account.

---

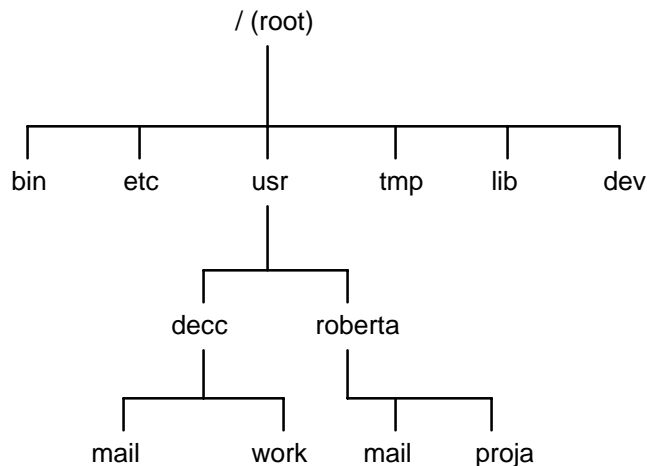
(Continued on next page)



---

## 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**.

---

(Continued on next page)

---

**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**

**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.

Gentran:Server's Data Flow Administration software 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**

Gentran:Server uses third-party database drivers and software to enable 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 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, 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, Gentran:Server displays a message to alert you of the problem.

---

(Continued on next page)

**File name length**

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

<b>File Type</b>	<b>Maximum Length of Name</b>
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

(Continued on next page)

## Character limitations

This table lists the characters that 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 with Advanced Data Distribution (Mailbox).

```

GENTRAN:Server
-----
| Sched | Script | Audit | DataMgr | Pattrn | Util | Mailbox | Exit |
|-----|-----|-----|-----|-----|-----|-----|-----|
Control/Configure Data Managers
  
```

**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.

(Continued on next page)


<b>(Contd) Option</b>	<b>Function</b>
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.
DataMgr	Displays the Data Manager Control screen.
Pattern	Accesses the Pattern feature.
Util	Displays the Utilities menu.
Mailbox	Displays the Advanced Data Distribution menu.
Exit	Exits the Gentran:Server Data Flow Administration software.

---

## How to Start and Exit Data Flow Administration

### Starting Data Flow Administration

Use this procedure to start the Data Flow Administration software.

Step	Action
1	<p>Is Gentran:Server running?</p> <ul style="list-style-type: none"> <li>▶ If YES, continue with Step 2.</li> <li>▶ If NO, double-click on the Gentran:Server icon to start the system.</li> </ul>
2	<p>Click the <b>Workbench</b> button on the Gentran:Server toolbar.</p> <div style="text-align: center;">  </div> <p><b>System Response</b> Gentran:Server prompts you for a login name and password.</p>
3	<p>Type the login ID and password of the owner of the environment and then press ENTER.</p> <p><b>System Response</b> Gentran:Server displays a “connecting to Server” message and then displays the copyright screen.</p>
4	<p>Press any key on the keyboard.</p> <p><b>System Response</b> Gentran:Server displays the Data Flow Administration main menu.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> GENTRAN:Server ----- Sched Script Audit DataMgr Pattern Util Mailbox Exit ----- Control/Configure Data Managers           </pre> </div> <p><b>WARNING</b> <b>If you receive the message server: command not found then check the PATH to make sure it contains \$EDI_ROOT and \$EDI_ROOT/bin.</b></p>

(Continued on next page)



## Exiting Data Flow Administration

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

Step	Action
1	<p>Do you want to save your changes on the current screen?</p> <ul style="list-style-type: none"> <li>▶ If YES, press the function key that corresponds to SAVE (usually F10).</li> <li>▶ If NO, continue with Step 2.</li> </ul> <p><b>Reference</b> See the <a href="#">How to Use Function Keys to Initiate an Action</a> in this chapter for information about using function keys.</p>
2	<p>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.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>GENTRAN:Server</p> <p>Sched Script Audit DataMgr Pattern Util Mailbox Exit</p> <p>Control/Configure Data Managers</p> </div>
3	<p>Select <b>Exit</b> and press ENTER.</p> <p><b>Note</b> You can also exit Gentran:Server Data Flow Administration by pressing E for exit or by pressing the ESC key twice.</p>

## 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 **DataMgr** from the main menu, press the TAB key or an arrow key until **DataMgr** 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 D to choose **DataMgr** 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.

**Example**

To select **arch** from the Data Manager Configuration screen, press the TAB key or the A key until **arch** is selected.

```
dmc Data Manager Configuration
Name A      Status  T Description
-----
alm n       0 l Async Line Manager
appm y      0 m Application Data Manager
appt n      0 x Application Translator Data Manager
arch n      0 a -Darch -Aarch -d0
base n      0 u Base Manager Model
dnld y      0 d UDF Data Manager
```

## How to Use Function Keys to Initiate an Action

**Introduction** Gentran:Server 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.

Select the script you want to copy

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

Press F4 to copy selected 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 Use a Shortcut Key to Initiate an Action

### Introduction

You can use some **shortcut keys** on any Gentran:Server Data Flow Administration screen. Other shortcut keys are available only on specific screens.

### All Data Flow Administration screens

Use these shortcut keys on any Data Flow Administration screen.

Shortcut Key	Function
CTRL+A	Goes to the UNIX shell  <b>Exiting the UNIX shell</b> To exit the shell and return to Data Flow Administration, you must type the <b>exit</b> command.
CTRL+T	Displays time of day.
ESC ESC	Exits the screen.

### All screens except Advanced Data Distribution

Use this shortcut key from any screen except Advanced Data Distribution screens.

Shortcut Key	Function
CTRL+E	Displays the Remove Data Manager TP Configurations screen. Use this screen to delete Trading Partnership configuration records based on a pattern.

### Data Manager Pattern Configuration screen

Use this shortcut key from the Data Manager Pattern Configuration screen.

Shortcut Key	Function
CTRL+D	Displays the Data Manager Selection screen. Use this screen to select another data manager within a pattern.

# How to Display Version and Copyright Information

---

**Introduction** If you call a Gentran:Server technical support representative, you may be asked for the version number of your software. The version number is on the Gentran:Server copyright screen.

---

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

---

**Procedure** Use this procedure to display version and copyright information.

Step	Action
1	Select <b>Util</b> from the main menu. <b>System Response</b> Gentran:Server displays the Utility menu.
2	Select <b>About...</b> from the Utilities menu. <b>System Response</b> The system displays the copyright screen.

---

# Miscellaneous Procedures

## How to Search For a Trading Partnership Record

**Introduction**    You can use the Trading Partnership Search feature to find and select Trading Partnership records.

You can search for specific values such as:

- ▶ Trading Partnership codes
- ▶ Interchange or Group IDs
- ▶ Organization codes
- ▶ Standard versions or Set IDs
- ▶ Categories
- ▶ Trading Partnership names.

### Starting a Trading Partnership search

You can search for Trading Partnership records any time the **TP search key** is offered at the bottom of a screen.

```

Archive Data (y/n) ? n
Error Handling Class
↓
Resource Group
Priority
F2:Select F5:Find F6:TP Search F7:Next F8:Prev F9:Quit

```

### Automatic starts

For some tasks you perform, the Trading Partnership search feature starts automatically.

#### Example

When you use the meld function to generate configuration records from a pattern, Gentran:Server automatically starts the Trading Partnership search at the appropriate time.

(Continued on next page)

**Procedure** Use this procedure to search for Trading Partnership records.

Step	Action
<p>1</p>	<p>Check the function keys listed at the bottom of the screen for the function key labeled TP search and then press the key.</p> <p><b>Example</b> If the TP search key is F6, then press F6.</p> <p><b>System Response</b> Gentran:Server displays the Trading Partnership Search options screen.</p> <div data-bbox="792 772 1263 1104" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <pre> Trading Partnership Search ----- Do you wish to enter a range of ...  Trading Partnership Codes ?      N Interchange and/or Group Ids ?   N Organization Codes ?            N Standard Version and/or Set Ids ? N User Defined Categories ?       N  Target an Inbound Mapping Table ? N Target an Outbound Mapping Table ? N Search by Trading Partnership Name ? N                     </pre> </div> <p><b>Note</b> For some tasks, Gentran:Server displays this screen automatically.</p>
<p>2</p>	<p>Select the criteria you want to use by entering <b>y</b> (Yes) or <b>n</b> (No) for each option.</p> <p><b>System Response</b> For each <b>y</b> entry, Gentran:Server prompts you to enter the search criteria.</p> <p><b>Example</b> If you type a <b>y</b> next to Trading Partnership Codes, Gentran:Server displays a screen for the starting and ending Trading Partnership codes.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>	
3	Type a value for one or more of the fields.  <b>Tips</b> To display all the Trading Partnership codes, type <b>000</b> in the Starting field and type <b>zzz</b> in the Ending field.  To use the same value in the Starting and Ending fields, type the value in the Starting field and press ENTER to fill in the same value in the Ending field.	
4	Press F10 to continue.  <b>System Response</b> Gentran:Server displays the Trading Partnership Search screen.	
5	Repeat Steps 2 through 4 for each search criterion that you want to use.  <b>Note</b> The User Defined Categories option is available only if you already have defined categories.	
6	Press F10 to continue.  <b>System Response</b>	
	<b>IF...</b>	<b>THEN Gentran:Server...</b>
	One or more records meet the criteria you entered	Lists all Trading Partnership records that satisfy the search. Continue with Step 7.
	No records meet the criteria you entered	Displays the message: No Trading Partnership Records Found within the Selected Criteria.  Press ESC to return to the Trading Partnership Search screen.  (Continued on next page)



(Contd) Step	Action																																																																						
7	<p>Enter <b>y</b> in the Proc Record (Process Record) field to select a record. To select all records, press F2.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Trading Partnership Search</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">TP Code</th> <th style="text-align: left;">TP Name/Description</th> <th style="text-align: left;">Std Version</th> <th style="text-align: left;">Set Id</th> <th style="text-align: left;">Proc Record</th> </tr> </thead> <tbody> <tr> <td>4097out</td> <td></td> <td>ANA1_8</td> <td>SRMHDR</td> <td>[N]</td> </tr> <tr> <td>INEND1911</td> <td>Sample Flow TP for 1911/invoic</td> <td>001911</td> <td>INVOIC</td> <td>[N]</td> </tr> <tr> <td>INEND210</td> <td>Sample Flow TP for M2_8/210</td> <td>M2/8</td> <td>210</td> <td>[N]</td> </tr> <tr> <td>INEND210FA</td> <td>Sample Flow FA for M2_8/210</td> <td>M2/8</td> <td>997</td> <td>[N]</td> </tr> <tr> <td>INEND837</td> <td>Sample Flow TP for 03032/837</td> <td>003032</td> <td>837</td> <td>[N]</td> </tr> <tr> <td>INEND837FA</td> <td>Sample Flow FA for 03032/837</td> <td>003032</td> <td>997</td> <td>[N]</td> </tr> <tr> <td>INEND850</td> <td>Sample Flow TP for 02040/850</td> <td>002040</td> <td>850</td> <td>[N]</td> </tr> <tr> <td>INEND850FA</td> <td>Sample Flow FA for 02040/997</td> <td>002040</td> <td>997</td> <td>[N]</td> </tr> <tr> <td>OUTBND02856</td> <td>Sample Flow for TP 856 Div 2</td> <td>003030</td> <td>856</td> <td>[N]</td> </tr> <tr> <td>OUTBND03856</td> <td>Sample Flow for TP 856 Div 3</td> <td>003030</td> <td>856</td> <td>[N]</td> </tr> <tr> <td>OUTBOUND856</td> <td>Sample Flow for TP 856</td> <td>003030</td> <td>856</td> <td>[N]</td> </tr> <tr> <td>TDCC204-1</td> <td>Sample Flow for 1st TP for 204</td> <td>M2/8</td> <td>204</td> <td>[N]</td> </tr> <tr> <td>TDCC204-2</td> <td>Sample Flow for 2nd TP for 204</td> <td>M2/8</td> <td>204</td> <td>[N]</td> </tr> </tbody> </table> <p style="text-align: center; font-size: small;">Esc-QUIT F1-HELP F2-MARK ALL F3-UNMARK ALL F10-CONTINUE</p> <p><b>Note</b> You can select one or more records.</p> </div>	TP Code	TP Name/Description	Std Version	Set Id	Proc Record	4097out		ANA1_8	SRMHDR	[N]	INEND1911	Sample Flow TP for 1911/invoic	001911	INVOIC	[N]	INEND210	Sample Flow TP for M2_8/210	M2/8	210	[N]	INEND210FA	Sample Flow FA for M2_8/210	M2/8	997	[N]	INEND837	Sample Flow TP for 03032/837	003032	837	[N]	INEND837FA	Sample Flow FA for 03032/837	003032	997	[N]	INEND850	Sample Flow TP for 02040/850	002040	850	[N]	INEND850FA	Sample Flow FA for 02040/997	002040	997	[N]	OUTBND02856	Sample Flow for TP 856 Div 2	003030	856	[N]	OUTBND03856	Sample Flow for TP 856 Div 3	003030	856	[N]	OUTBOUND856	Sample Flow for TP 856	003030	856	[N]	TDCC204-1	Sample Flow for 1st TP for 204	M2/8	204	[N]	TDCC204-2	Sample Flow for 2nd TP for 204	M2/8	204	[N]
TP Code	TP Name/Description	Std Version	Set Id	Proc Record																																																																			
4097out		ANA1_8	SRMHDR	[N]																																																																			
INEND1911	Sample Flow TP for 1911/invoic	001911	INVOIC	[N]																																																																			
INEND210	Sample Flow TP for M2_8/210	M2/8	210	[N]																																																																			
INEND210FA	Sample Flow FA for M2_8/210	M2/8	997	[N]																																																																			
INEND837	Sample Flow TP for 03032/837	003032	837	[N]																																																																			
INEND837FA	Sample Flow FA for 03032/837	003032	997	[N]																																																																			
INEND850	Sample Flow TP for 02040/850	002040	850	[N]																																																																			
INEND850FA	Sample Flow FA for 02040/997	002040	997	[N]																																																																			
OUTBND02856	Sample Flow for TP 856 Div 2	003030	856	[N]																																																																			
OUTBND03856	Sample Flow for TP 856 Div 3	003030	856	[N]																																																																			
OUTBOUND856	Sample Flow for TP 856	003030	856	[N]																																																																			
TDCC204-1	Sample Flow for 1st TP for 204	M2/8	204	[N]																																																																			
TDCC204-2	Sample Flow for 2nd TP for 204	M2/8	204	[N]																																																																			
8	Press F10 to continue.																																																																						



---

# Designing Basic Process Flows

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	2
	▶ Process Flows .....	5
	▶ Data Managers .....	7
	▶ Example Inbound Process Flow .....	11
	▶ Scripts .....	13
	▶ Example Process Flow With Script .....	14
	<b>Designing a Process Flow</b>	
	▶ Overview .....	15
	▶ How to Identify a Flow's Purpose .....	16
	▶ How to Select Data Managers .....	18
	▶ How to Add Other Components .....	22
	▶ Example: Designing an Inbound Process Flow .....	23
	▶ Inbound XML Process Flows .....	26
	▶ Process Flow Worksheet .....	28
	<b>Stages in Basic Processes</b>	
	▶ Inbound Processing .....	32
	▶ Outbound Processing .....	35
	<b>Building a Process Flow</b>	
	▶ The Flow of Work .....	38

---

# Overview

## Introduction

### In this chapter

This chapter introduces the basic components of a process flow and describes how to design basic flows.

Gentran:Server enables you to create sophisticated process flows with numerous components. You can:

- ▶ Use the Process Control Manager wizard to create a simple inbound or outbound process flow
- ▶ Create each of the flow components individually with EC Workbench, building flows tailored to your organization's needs.

### Reference

See the [Creating a Flow with the PCM Wizard](#) chapter in this guide for information about using the Process Control Manager wizard.

### Cautions

**You cannot use EC Workbench to modify flow components you created with the PCM Wizard.**

### Key terms

This table lists the key terms used in this chapter.

Term	Description
archive	The process of capturing and storing a copy of a document after a data manager has processed it.
archive directory	The directory to which a data manager routes an archive copy of a document it processes. This directory is the archive handler's work directory.
archive handler	The special data manager designed to archive data.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
configuration record	<p>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:</p> <ul style="list-style-type: none"> <li>▶ Specifies the Trading Partnership code or file name that the data manager uses to identify data</li> <li>▶ Tells the data manager what to do with the data it identifies.</li> </ul>
data manager	<p>An intelligent agent program that periodically scans a directory or queue for data files and then processes the files it finds. Processing can include:</p> <ul style="list-style-type: none"> <li>▶ Routing data</li> <li>▶ Invoking scripts</li> <li>▶ Archiving data</li> <li>▶ Handling data errors.</li> </ul>
destination directory	The end location of data in a process flow step.
initialization file	The file you use to set 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.
personality	The data manager type, such as inbound, download, file, and so on. The personality determines what type of processing the data manager performs on the data.
process flow	A set of processing components that process and move data from one location in the Gentran:Server system to another.
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 a data manager uses to process files.
scan directory	See <i>work directory</i> .
script	<p>A set of commands that controls processes or performs some action.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Term</b>	<b>Description</b>
script directory	The directory that contains all Gentran:Server scripts.
Script Manager	The Gentran:Server program that directs the script interpreter to execute the commands in a script.
source directory	The starting location of data in a process flow. This is the work directory of the first data manager in the flow.
Trading Partner record	One of the records maintained in trading partner files: Trading Partnership record, Interchange Organization record, Group Organization record, Contact record.
UNIX mail script	A UNIX script that you can use to send electronic mail messages based on the results of a 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 in which a data manager looks for the files or queued files it is to process. This directory is sometimes referred to as the <i>scan directory</i> .

---

# Process Flows

---

**Definition** A **process flow** is a set of 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 processes and routes EDI data that a trading partner sends to you.
- ▶ **Outbound**, which processes and translates your application data into EDI data so that you can send it to a trading partner.

**Comment**

If you have the Gentran:Server XML translation option, you can create process flows to route inbound and outbound XML data.

---

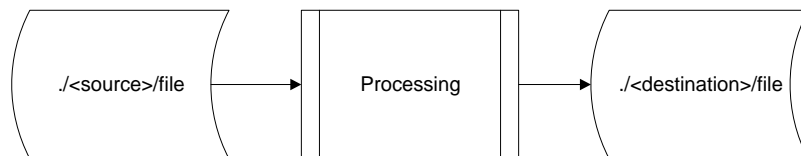
**Moves data from source to destination**

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.



---

(Continued on next page)

---

**Source** The source of files for a data manager in a process flow can be a:

- ▶ **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 a data manager in a process flow is usually 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 destination directories, each named for a set type.
- ▶ **TP Code**, which selects the Trading Partnership Code as the symbolic name of the destination directory. This selection results in multiple possible destination directories, each named for a Trading Partnership Code.
- ▶ **Categories**, which selects a Trading Partnership category as the symbolic name of the destination directory. This selection results in multiple possible destination directories, each named for a category.

With some data managers, you can use the wildcard character or combinations of symbols to create destination directories based on a file extension, file name, interchange ID, environment variable, standard version, and more.

**Reference**

See the [Working with Configuration Records](#) chapter in this guide for more information about specifying destination directories.

---



# Data Managers

---

**Definition** A **data manager** is an intelligent agent program that processes files. Processing can include:

- ▶ Translating data
- ▶ Routing data
- ▶ Invoking scripts
- ▶ Archiving data
- ▶ Handling data that contains errors
- ▶ Other types of processing that the data manager is designed to do.

---

**When processing starts** Data managers process files when one of the following happens:

- ▶ The data manager finds files in its work directory or queue during a periodic scan
- ▶ A data manager signals a second data manager that it has routed files to the second data manager's work directory or queue.

---

**Role of data managers** The data managers are the primary processing components in a process flow.

Data managers:

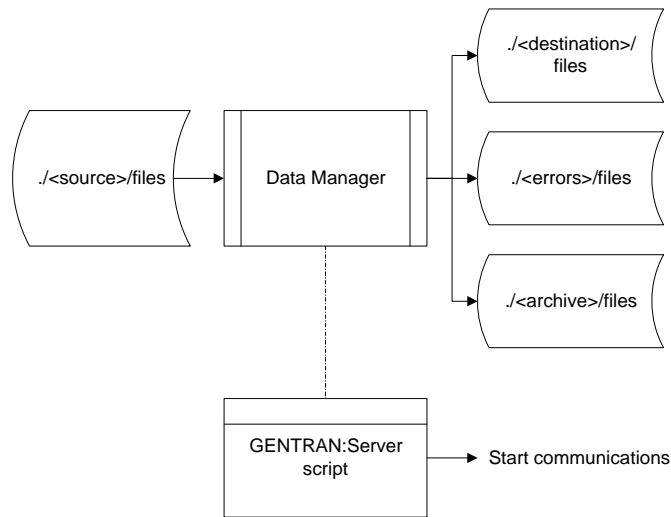
- ▶ Handle and route data between directories
- ▶ Invoke other process flow components, such as scripts.

---

(Continued on next page)

---

**Illustration** This illustration shows the role of the data manager in a process flow.



---

### Railroad track analogy

One way to view data flowing through Gentrans: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. Data managers are the components that serve as switching devices. They sort out data, reformat or transform it, and route it to the appropriate destination.

---

### Name limitations

Data manager names are limited to four characters.

---

(Continued on next page)

## Basic data manager types

This table lists the functions of the four major types of data managers.

Data Manager	Common Names	Functions
Inbound	<i>inbd</i> <i>edii</i> <i>edio</i>	<ul style="list-style-type: none"> <li>▶ Understands EDI data (X12, EDIFACT, etc.) and verifies that the data is properly enveloped, and constructed.</li> <li>▶ Can break up the input file by trading partner at the interchange, group, or set level.</li> <li>▶ Can record a document reference number, which is used to detect duplicate data and to retrieve archived data.</li> </ul>
application	<i>appm</i>	<ul style="list-style-type: none"> <li>▶ Can compare the data file to the application file definition as specified by the <i>&lt;filename&gt;.ddf</i> or <i>&lt;filename&gt;.app</i> file used in the map for this trading partner.</li> <li>▶ Can break up the input file by trading partner at the interchange, group, or set level.</li> <li>▶ Can record a document reference number, which is used to detect duplicate data and to retrieve archived data.</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

Data Manager	Common Names	Functions
translation	<i>xltr</i> <i>xlti</i> <i>xli1</i> <i>xli2</i> <i>xlto</i> <i>xlo1</i> <i>xlo2</i> <i>xdbi</i> <i>xdbo</i>	<p>Calls the Script Manager to execute a translation script. The translation script calls the translator and other runtime programs, such as <b>ediarc</b> and <b>envelope</b>.</p> <p>You can configure translation data managers to handle either EDI or application data.</p> <ul style="list-style-type: none"> <li>▶ You could use <b>xli1</b> for a translation data manager that handles inbound EDI data that will be translated to application data or to another EDI standard.</li> <li>▶ You could use <b>xlo1</b> for a translation data manager that handles application data that will be translated to EDI or to another application format.</li> </ul>
file	<i>file</i>	<ul style="list-style-type: none"> <li>▶ Recognize files by file name</li> <li>▶ Can copy and rename files</li> <li>▶ Able to generate archive files</li> <li>▶ Can be used to execute pre- or post-processing scripts</li> </ul>

### Reference

For a complete description of all data manager types, see the [Working with Data Managers](#) chapter in this guide.

### Optional XML data managers

If you have the Gentran:Server XML translation option, you also have available inbound XML data managers and outbound XML data managers.

## Example Inbound Process Flow

---

### Basic process flow

In its simplest form, a process flow consists of a series of data managers. Each data manager:

- Retrieves files from its own **work directory** or **queue**
- Processes the files
- Deposits the processed files into a specified destination directory.

---

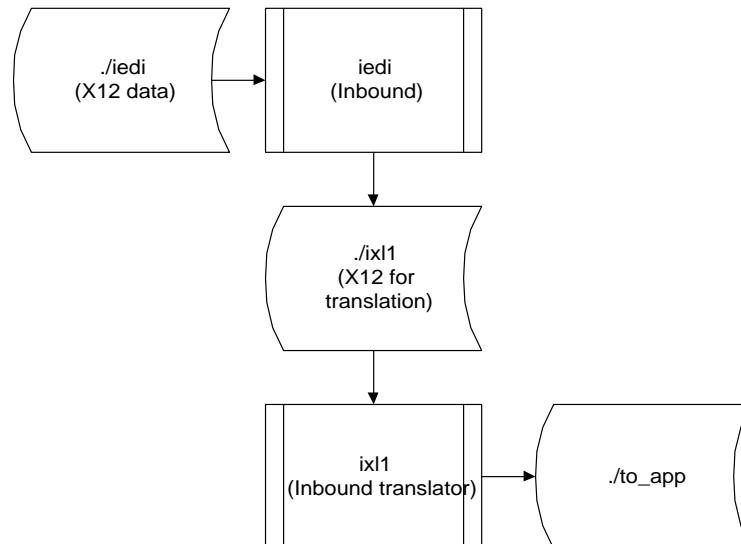
### Common directories

The destination directory for a data manager can be the source (work) directory for the next data manager in the process flow.

---

### Illustration

This illustration shows how process flow components move X12 EDI data in a Gentran:Server system.

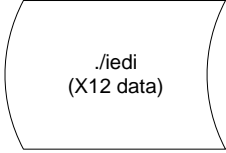
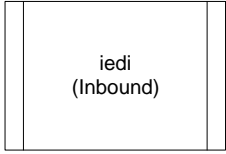
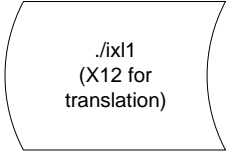
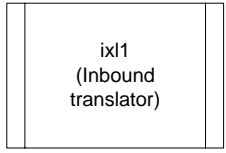



---

(Continued on next page)

### Parts of the process flow

This table describes the parts of example process flow.

Part	Description
 <p>.iedi (X12 data)</p>	<p>The source directory that contains data in X12 format. In this flow, this is also the <i>iedi</i> data manager's work directory.</p>
 <p>iedi (Inbound)</p>	<p>The inbound data manager that scans the <i>.iedi</i> directory for data and then processes the files it finds.</p>
 <p>.ixl1 (X12 for translation)</p>	<p>The <i>iedi1</i> data manager's destination directory and the <i>ixl1</i> data manager's work directory.</p> <p>This is where the <b>iedi1</b> inbound data manager sends files that it has processed. It is also the work directory that the <i>ixl1</i> data manager looks in for files to process.</p>
 <p>ixl1 (Inbound translator)</p>	<p>The inbound translation data manager. This data manager scans the <i>.ixl1</i> directory for files, translates the files it finds into application directory format, and routes the resulting files to the <i>.to_app</i> directory.</p>
 <p>.to_app</p>	<p>The destination directory for the inbound translation data manager.</p> <p>This is where the <b>ixl1</b> data manager sends the X12 data it has translated.</p>

# Scripts

---

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

---

**Definition** A 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 Gentran:Server what to do.

---

**Role of scripts** Gentran:Server scripts start and control processes in a process flow. They also perform processing actions that are out of the scope of a data manager.

You can use Gentran:Server scripts to:

- Pull data from host through an Ethernet or other host connection and deposit it into an inbound data manager's work directory
- Invoke data managers, other 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 [Working with Scripts](#) chapter in this guide for more information about creating and using Gentran:Server scripts.

---

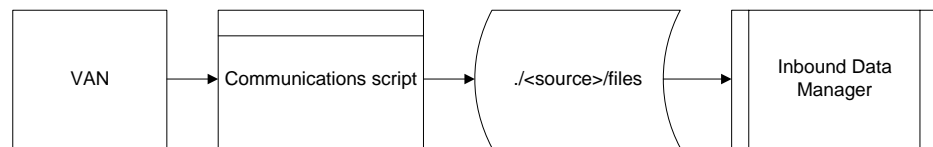
## Example Process Flow With Script

---

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

---

**Illustration**    This illustration shows how a Gentran:Server script moves EDI data from a VAN to the inbound data manager'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. <b>Reference</b> See <a href="#">How to Identify a Flow's Purpose</a>
2	Select the data managers for the flow. <b>Reference</b> See <a href="#">How to Select Data Managers</a>
3	Add other components. <b>Reference</b> See <a href="#">How to Add Other Components</a>

---

## How to Identify a Flow's Purpose

---

**Introduction**    The purpose of a process flow guides all 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 Gentran:Server or moving out of 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 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 interchange
- ▶ Translate the data into application data
- ▶ Move the data to the appropriate application software.

---

(Continued on next page)

**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.

**Example 3 - Inbound XML (If you have the Gentran:Server XML translation option)**

The purpose of my inbound XML process flow is to:

- Retrieve inbound XML data from my internet mailbox
  - Separate the XML data based on XML splitter table rules
  - Translate the data into application data
  - Move the data to the appropriate application software.
-

## How to Select Data Managers

**Introduction** When you design a process flow, you need to consider the personality types of the data managers you need to serve the purpose of the flow.

**Section criteria** Select the data manager 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 data manager's personality type** Use this table to select the data managers for your process flow. The four-character personality type shown in parentheses is the personality type you will see listed in the base initialization file. The base initialization file is the master file that contains all the possible parameters and values for every data manager.

Use this data manager	When you want to...
Inbound ( <i>inbd</i> )	Process inbound EDI data at the beginning of an inbound data flow.  <b>Comments</b> <ul style="list-style-type: none"> <li>▶ Input files can have any name.</li> <li>▶ The data manager determines how to process the data by:               <ol style="list-style-type: none"> <li>a) Extracting the Trading Partnership code from the six key fields of the EDI interchange</li> <li>b) Locating the Trading Partnership configuration record that tells the data manager what to do with the data.</li> </ol> </li> </ul> <b>Reference</b> See the <a href="#">Working with Configuration Records</a> chapter in this guide for information about Trading Partnership configuration records.  <div style="text-align: right; color: red;">(Continued on next page)</div>

<b>(Contd) Use this data manager</b>	<b>When you want to...</b>
Application <i>(appm)</i>	Process application data at the beginning of an outbound data flow.  <b>Comments</b> The data manager can sort the input file by Trading Partnership code at the set level.  Make sure that the input files are named for one of the following: <ul style="list-style-type: none"> <li>▶ Trading Partnership code</li> <li>▶ Application file that the data represents.</li> </ul>
Download <i>(dnld)</i>	Process files in translator Record File Layout format.  <b>Comments</b> <ul style="list-style-type: none"> <li>▶ Used at the beginning of an outbound process flow.</li> <li>▶ Sorts data by Trading Partnership code.</li> </ul>
Inbound translation <i>(xltr)</i>	Do one of the following: <ul style="list-style-type: none"> <li>▶ Translate EDI data to application format.</li> <li>▶ Translate EDI data in one standard format to another standard format.</li> </ul>
Outbound translation <i>(xltr)</i>	Do one of the following: <ul style="list-style-type: none"> <li>▶ Translate application data to EDI data.</li> <li>▶ Translate application data in one format to another application format.</li> </ul>
File <i>(file)</i>	Handle file based on the file name only, and not the file contents.  <b>Comments</b> Data manager determines how to process the data by: <ol style="list-style-type: none"> <li>1. Finding the file name</li> <li>2. Locating the File Name configuration record that tells the data manager what to do with the data.</li> </ol> <b>Reference</b> See the <a href="#">Working with Configuration Records</a> chapter in this guide for information about configuration records.  <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Use this data manager</b>	<b>When you want to...</b>
Host command card  <i>(hcmd)</i>	Process the data in a file according to the commands embedded in the file.  <b>Comments</b> The files this data manager processes contain separator characters that distinguish the commands in the file from the data in the file. You define the separator characters.  <b>Reference</b> See the <a href="#">Working with Data Managers</a> chapter in this guide for more information about host command card data managers.
Archive  <i>(arch)</i>	Make an archive copy of data to create a processing audit trail.  <b>Comments</b> The data is archived after the data manager has processed it.

### Optional XML data managers

If you have the Gentran:Server XML translation option, you can select an inbound or outbound XML data managers for your process flow.

<b>Use this data manager</b>	<b>When you want to...</b>
Inbound XML  <i>(xml)</i>	Process inbound XML data at the beginning of an inbound data flow.  <b>Comment</b> <ul style="list-style-type: none"> <li>▶ Input files can have any name.</li> <li>▶ The data manager determines how to process the data by:               <ol style="list-style-type: none"> <li>a) Splitting the XML file according to the XML splitter table rules</li> <li>b) Locating the Trading Partnership configuration record for each temporary XML file resulting from the splitting process. The configuration record tells the data manager what to do with the data.</li> </ol> </li> </ul> <b>Reference</b> See the <a href="#">Working with Configuration Records</a> chapter in this guide for information about Trading Partnership configuration records.  <div style="text-align: right; color: red;">(Continued on next page)</div>

<b>(Contd) Use this data manager</b>	<b>When you want to...</b>
Outbound XML  ( <i>xml</i> )	Process outbound XML data  <b>Comment</b> Data manager determines how to process the data by: <ol style="list-style-type: none"><li>1. Finding the file name</li><li>2. Locating the File Name configuration record that tells the data manager what to do with the data.</li></ol> <b>Reference</b> See the <a href="#">Working with Configuration Records</a> chapter in this guide for information about configuration records.

---

## How to Add Other Components

---

**Introduction**    After choosing the data managers for your flow, 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 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.

<b>IF you want to...</b>	<b>THEN add...</b>
Have 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.

---

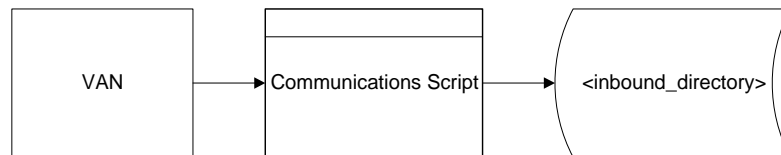
**Purpose of the process flow** The first step in creating a process flow is to decide the purpose of the flow. In this example, our purpose is to:

- ▶ 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 first data manager**

The data is in EDI format, so we start the flow with an inbound data manager. We name this data manager “inbd.”

**Comment**

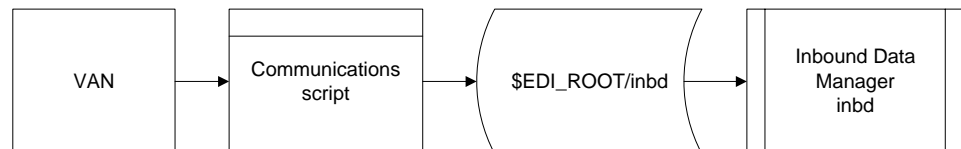
The inbound data manager, *inbd*, sorts the data in the input file based on interchange, group, or set, depending on the settings in the *inbd* initialization file. It also verifies the structure of the data in the file and identifies the Trading Partnership code.

---

(Continued on next page)

### Creating a directory for the files

The communications script needs a directory in which to deposit the X12 files and the inbound data manager needs the same directory to look in for files. Because this directory is the source or **work directory** for the data manager that starts our process flow, we name the directory for the inbound data manager so that the name is easy to remember.

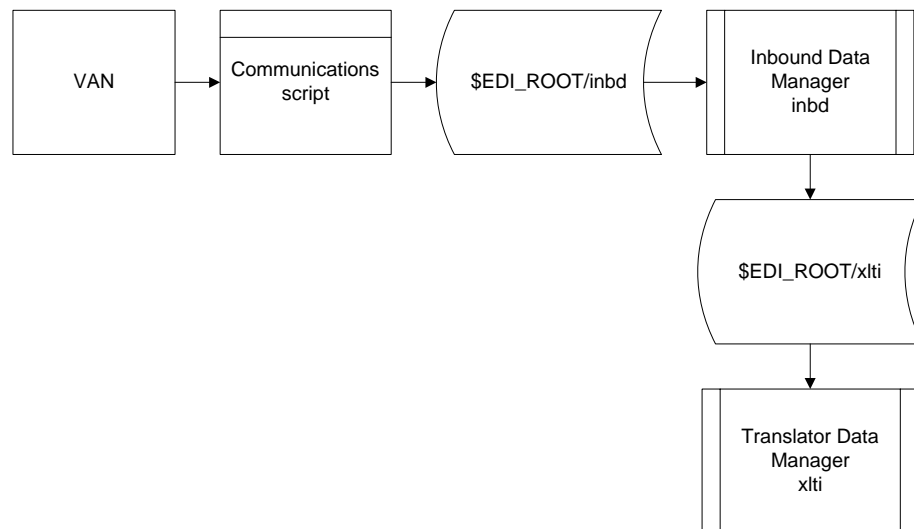


### Selecting the next data manager

We assume that no special processing is required, so we follow the inbound data manager with a translation data manager. The translation data manager translates the X12 data into a format that our application can understand. We name this data manager *xlti*.

#### Comment

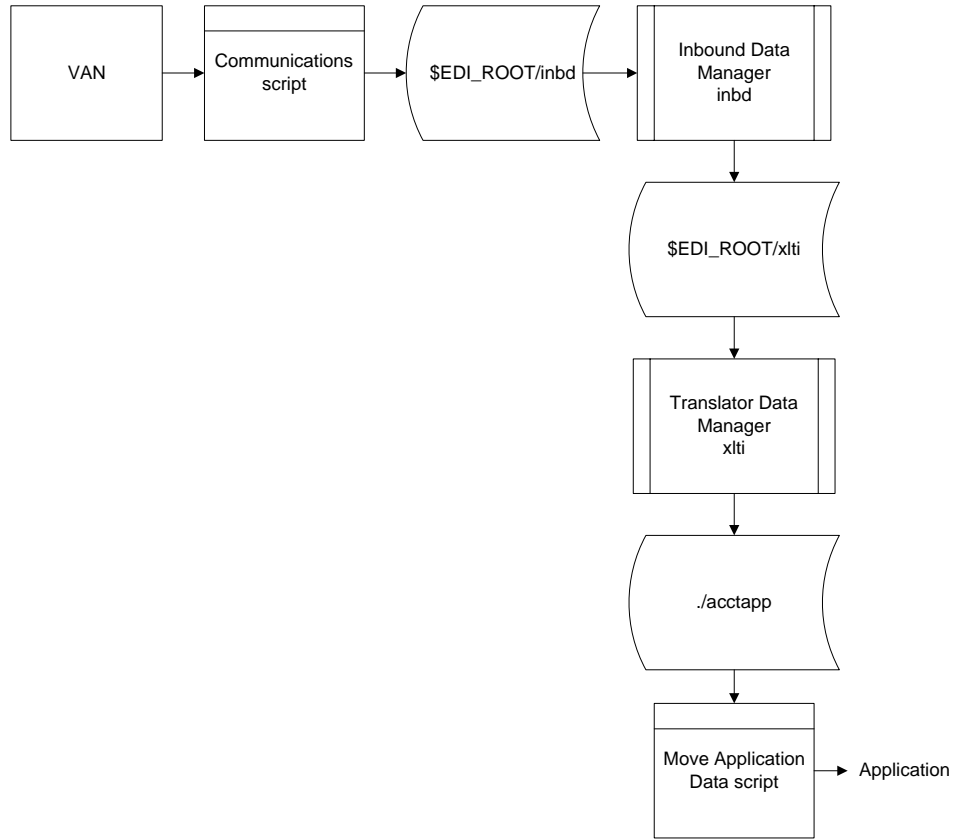
In this illustration, notice that the inbound data manager's destination directory is the same as the translator data manager's source or work directory. Also note that we named the directory for the translator data manager. Naming a data manager's work directory for the data manager that scans it keeps the relationship clear.



(Continued on next page)

**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.



## Inbound XML Process Flows

**Introduction**    If you have the Gentran:Server XML translation option, you can use the XML data manager to split XML files into smaller documents, and then route and translate each document separately.

**Stage table**    This table describes the stages in the process of configuring your system to route inbound XML documents.

Stage	Description
1	<p>Create a map for inbound XML data.</p> <p><b>Reference</b> See the section <a href="#">Creating a Map</a> in the <i>Gentran:Server for UNIX and Workstation XML User's Guide</i>.</p>
2	<p>Save the side of the map that describes the layout of the XML document as a file definition (DDF).</p> <p><b>Reference</b> See the topic <a href="#">How to Save a File Definition</a> in the <i>Gentran:Server for UNIX and Workstation Application Integration User's Guide</i>.</p>
3	<p>Configure the elements that you want the system to use to split the XML file into smaller documents or sets.</p> <p><b>Reference</b> See the section <a href="#">Configuring XML Elements to Split Files</a> in the <i>Gentran:Server for UNIX and Workstation XML User's Guide</i>.</p>
4	<p>Create the Trading Partnership rules that the system will use to determine the Trading Partner Code in an XML document</p> <p><b>Reference</b> See the topic <a href="#">How to Define XML TP Rules</a> in the <i>Gentran:Server for UNIX and Workstation XML User's Guide</i>.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Stage	Description
5	<p>Link the Trading Partnership rules from Stage 4 and the elements you configured in Stage 3 to the file definition you created in Stage 2.</p> <p><b>Reference</b> See the section <a href="#">Linking Rules to a Trading Partnership Code</a> in the <i>Gentran:Server for UNIX and Workstation XML User's Guide</i>.</p>
6	Create the XML flow, using the XML data manager to split the XML file and route each document according to the destination specified in the Trading Partnership record.

### How the XML data manager routes XML documents

This table describes how the XML data manager routes XML documents.

Stage	Description
1	The XML data manager uses the XML splitter tables (xmlspl1, xmlspl2, and xmlspl3) to determine how to break the file into smaller documents.
2	The data manager uses the element IDs to find the Trading Partnership rules.
3	The data manager reads the XML document to find the string for each Trading Partnership rule.
4	The data manager uses the strings to find the corresponding Trading Partner Code in the cross reference table, xmlxref.
5	Once the data manager knows the Trading Partner Code, it retrieves the routing information for the Trading Partnership and routes the XML document to the specified directory or queue.

## 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 (inbound, outbound)	
Purpose	
Communication connections	
Names of application sending or receiving files to and from Gentran:Server	
Scripts needed to pull record layout files from application?	

(Continued on next page)

<b>(Contd) Item</b>	
Scripts needed 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	<p>(Continued on next page)</p>

<b>(Contd) Item</b>	
Types of error notification	
Where error messages will be directed	

**Data managers**

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

Order	Type	Name

(Continued on next page)



---

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

Name	Function

---

**Directories** Use this table to record the names of the directories that you need to create.

Name	Description

---



# Stages in Basic Processes

## Inbound Processing

---

**Introduction** The inbound data manager handles data files in EDI format.

---

**Processing stages** This table describes the stages in a typical inbound process flow.

Stage	Description	
1	The inbound data manager scans for files in its work directory or queue at regular intervals, looking for valid EDI envelope information. The data manager processes each data set that it encounters as a transaction.	
	IF the data manager finds...	THEN it checks...
	Interchange segments	For the group ID and set ID.  <b>Comment</b> If the data manager encounters an ISA interchange acknowledgment (TA1) segment, it does not check for the group ID and set ID.
Invalid or incomplete data	Its initialization file for the directory and action specified for a STRUCT_ERROR type error.  (Continued on next page)	

<b>(Contd) Stage</b>	<b>Description</b>	
2	When the inbound data manager recognizes an envelope, it uses the codes that make up the envelope to identify the Trading Partnership code in the Gentran:Server Trading Partnership file.	
	<b>IF...</b>	<b>THEN...</b>
	The data manager does not find a Trading Partnership code in the Trading Partnership file	The data manager checks its initialization file to determine what action to take: <ul style="list-style-type: none"> <li>▶ If the ADD_TP flag is ON AND a model Trading Partnership record is set up for the version and set ID, then the inbound data manager creates Trading Partnership records and configuration records.</li> <li>▶ If the ADD_TP flag is OFF or if a model Trading Partnership record does not exist for the version and set ID, then the inbound data manager takes the action specified in its initialization file for a NO_TP type error.</li> </ul>
3	The inbound data manager uses the Trading Partnership code to identify the appropriate configuration record.	
	<b>IF the data manager...</b>	<b>THEN it...</b>
	Finds the configuration record	Directs the file to the destination directory.
Does not find a matching configuration record	Checks the initialization file for the action and directory specified for a NO_CONFIG type error.  (Continued on next page)	

<b>(Contd) Stage</b>	<b>Description</b>	
4	The data manager checks the document specifier file linked to the Trading Partnership code to find the unique document ID fields in the transaction. The data manager checks its transaction register for a match.	
	<b>IF...</b>	<b>THEN the data manager...</b>
	The data manager does not find a match in the transaction register	Makes an entry in the transaction register. The data manager writes all sets to the transaction register, regardless of the routing method (Interchange, Group, or Set).
	You do not allow duplicates and the data manager finds a matching entry in the transaction register or within the routing method (interchange or group)	Checks its initialization file for the action and directory specified for a DUPLICATES type error.
	You allow duplicates and the data manager finds a matching entry in the transaction register	Updates the date/time field in the transaction register.
5	If the archive flag is ON, the data manager moves a copy of the data to the directory specified in the ARCHIVE_DIRECTORY parameter of its initialization file. This is the archive handler's work directory.	
6	The data manager makes an entry in its log and Life Cycle event file for each data set. Once end-of-file is reached, the system deletes the original data file.	
7	<p>The script manager invokes the script specified in the configuration record for the Trading Partnership code.</p> <p><b>Comment</b> The point at which the script is invoked is determined by the value in the SCRIPT_RUN_SWITCH parameter of the data manager's initialization file.</p>	
8	If Gentran:Server is configured to use the Life Cycle feature, the system invokes the Life Cycle load programs and deletes the Life Cycle event file.	

# Outbound Processing

**Introduction** The download or outbound data manager is designed to handle data files in record file layout format. The format is defined by the Gentran:Server translator for outbound translation.

**Processing stages** This table describes the stages in a typical outbound processing flow.

Stage	Description	
1	The download data manager scans its work directory or queue for outbound record layout file-formatted transactions at regular intervals. The data manager processes each data set it encounters as a transaction.	
	<b>IF the data manager finds...</b>	<b>THEN it...</b>
	A complete file	Compares the Trading Partnership code and modifier code to the codes established in the configuration record.  <b>Comment</b> The data manager processes each transaction separately according to the Trading Partnership code and the modifier.
	Files with syntactical or structural errors	Checks the initialization file for the action and directory for a STRUCT_ERROR type error.
2	If the Trading Partnership code and modifier exist, the download data manager compares the document reference number to those in its transaction register.	

(Continued on next page)

<b>(Contd) Stage</b>	<b>Description</b>	
3	The download data manager checks the transaction register for duplicate sets within the routing method (interchange or group).	
	<b>IF the data manager...</b>	<b>THEN it...</b>
	Does not find duplicate sets	Enters the transaction in the transaction register and continues.
Finds duplicate sets and you do not allow duplicates	Checks the initialization file for the action and directory for DUPLICATE type errors.	
4	The download data manager uses the Trading Partnership code to identify the appropriate configuration file.	
	<b>IF the data manager...</b>	<b>THEN it...</b>
	Finds a configuration record for the Trading Partnership code	<ul style="list-style-type: none"> <li>▶ Directs the file to the destination directory named in the configuration record.</li> <li>▶ Strips the Trading Partnership code modifier from the record layout file.</li> </ul>
Does not find a matching configuration record	Checks its initialization file for the action and directory specified for a NO_CONFIG type error.	
5	If its initialization file is set for archiving data and the configuration record is set for archiving, the download data manager moves a copy of the transaction to the archive directory specified in the ARCHIVE_DIRECTORY parameter.	
6	The data manager makes an entry in its process log and Life Cycle event file for each data set. Once end-of-file is reached, Gentran:Server deletes the original file in the work directory.	

(Continued on next page)

<b>(Contd) Stage</b>	<b>Description</b>
7	<p>The script manager invokes the script specified in the configuration record for the Trading Partnership code.</p> <p><b>Comment</b> The point at which the script is invoked is determined by the value in the SCRIPT_RUN_SWITCH parameter of the data manager's initialization file.</p>
8	<p>If Gentran:Server is configured to use the Life Cycle feature, the system invokes the Life Cycle load programs and deletes the Life Cycle event file.</p>

---

# Building a Process Flow

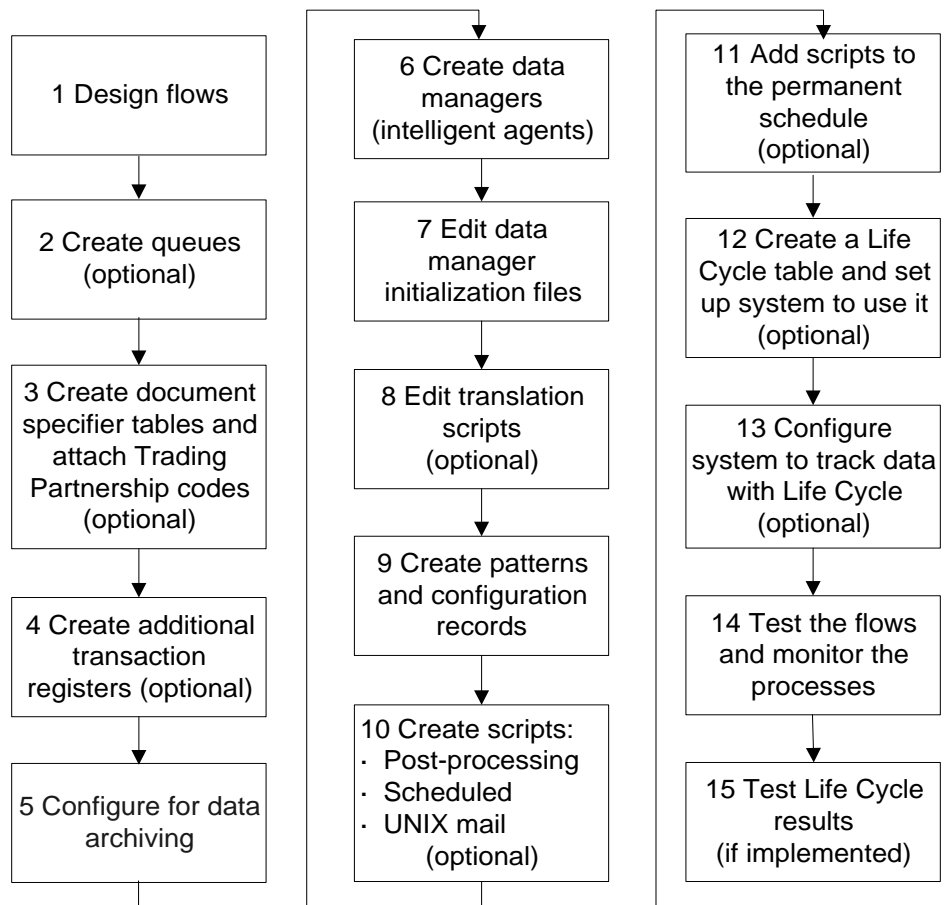
## The Flow of Work

### Introduction

After you design a process flow and identify its components, you can start building the flow.

### Process flow diagram

This diagram shows the stages in building a process flow.



(Continued on next page)



**The building process**

This table describes the tasks in building a process flow.

Task	Description
1	Design the process flows. <b>Reference</b> See <a href="#">Designing a Process Flow</a> .
2	Create any queues you want to use in the process flows. (This task is optional.) <b>Reference</b> See the <a href="#">Using Queues</a> chapter in this guide.
3	Create document specifier tables and attach Trading Partnership codes. (This task is optional.) <b>Reference</b> See the <a href="#">Defining the Document Reference Number</a> chapter in this guide.
4	Create any additional transaction registers. (This task is optional.) <b>Reference</b> See the <a href="#">Using Transaction Registers to Track Documents</a> chapter in this guide.
5	Configure for data archiving. (This task is optional.) <b>Reference</b> See the <a href="#">Archiving Your Data</a> chapter in this guide.
6	Create the data managers (intelligent agents). <b>Reference</b> See the <a href="#">Working with Data Managers</a> chapter in this guide.
7	Edit the data manager initialization files. <b>Reference</b> See the <a href="#">Maintaining Initialization Files</a> chapter in this guide.
8	Edit the translation scripts. (This task is optional.) <b>Reference</b> See the <a href="#">Working with Scripts</a> chapter in this guide.  (Continued on next page)

<b>(Contd) Task</b>	<b>Description</b>
9	<p>Create the configuration records manually or use patterns to generate them.</p> <p><b>Reference</b> See the <a href="#">Working with Configuration Records</a> chapter in this guide.</p>
10	<p>Create the Gentran:Server post-processing scripts, scheduled scripts, and UNIX mail scripts you want to use in the process flows.</p> <p><b>Reference</b> See the <a href="#">Working with Scripts</a> chapter in this guide.</p>
11	<p>Add Gentran:Server scripts to the Permanent Schedule. (This task is optional.)</p> <p><b>Reference</b> See the <a href="#">Running Scripts on a Schedule</a> section in the <a href="#">Running Scripts</a> chapter in this guide.</p>
12	<p>Create a Life Cycle database table and set up your system to use it. (This task is optional.)</p> <p><b>Reference</b> See the <a href="#">Setting Up Life Cycle</a> chapter in this guide.</p>
13	<p>Configure your system to track data with the Life Cycle feature. (This task is optional.)</p> <p><b>Reference</b> See the <a href="#">Tracking Data with Life Cycle Files</a> chapter in this guide.</p>
14	<p>Test the process flows and monitor Gentran:Server processes.</p> <p><b>Reference</b> See the <a href="#">Monitoring Processes</a> chapter in this guide.</p>
15	<p>Test Life Cycle results (if you implemented Life Cycle).</p>

---

# Creating a Flow with the PCM Wizard

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	3
	▶ The Process Control Manager Wizard .....	5
	▶ Process Flows .....	7
	▶ The Flow of Work .....	9
	<b>Beginning a Flow</b>	
	▶ Overview .....	11
	▶ Creating the Supporting Files .....	12
	▶ Flow Identification Dialog Box .....	14
	▶ How to Name and Describe the Flow .....	16
	<b>Creating an Inbound Flow</b>	
	▶ 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
	<b>Creating an Outbound Application Flow</b>	
	▶ 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
	<b>Creating an XML Flow</b>	

▶ Overview .....	58
▶ Source Setup Dialog Box (XML) .....	60
▶ How to Set Up the Source Agent (XML Flow) .....	63
▶ Processing Agent Dialog Box (XML Flow) .....	65
▶ Translation Options Dialog Box (XML Flow) .....	67
▶ How to Set Up the Processing Agent (XML Flow) .....	69
▶ Delivery Agent Dialog Box (XML Flow) .....	71
▶ How to Set Up the Delivery Agent (XML Flow) .....	75
<b>Creating an Inbound NCPDP Flow</b>	
▶ Overview .....	79
▶ Source Dialog Box (Inbound NCPDP Flow) .....	81
▶ How to Set Up the Source Agent (Inbound NCPDP Flow) .....	84
▶ Processing Dialog Box (Inbound NCPDP Flow) .....	86
▶ Translation Options Dialog Box .....	88
▶ How to Set Up the Processing Agent (Inbound NCPDP Flow) ...	90
▶ Delivery Dialog Box (Inbound NCPDP Flow) .....	91
▶ How to Set Up the Delivery Agent (Inbound NCPDP Flow) .....	94
<b>Completing a Flow</b>	
▶ Overview .....	97
▶ Error Handling Dialog Box .....	98
▶ How to Set Up Error Handling Instructions .....	100
▶ Trading Partner Records Dialog Box .....	102
▶ How to Add Trading Partnership Records to the Flow .....	104
▶ How to Delete Trading Partnerships from the Trading Partner Records Dialog Box .....	107
<b>Using Flow Summaries</b>	
▶ The Flow Summary .....	109
▶ Flow Summary Views .....	111
▶ How to Expand and Collapse the Flow View .....	114
▶ Flow Summary Reports .....	115
▶ How to Print Flow Summary Reports .....	117
<b>Maintaining Process Flows</b>	
▶ Overview .....	119
▶ How to Edit a Process Flow .....	120
▶ How to Delete Trading Partnerships From the Flow .....	122
▶ How to Delete a Flow .....	124

# Overview

## Introduction

**In this chapter** This chapter describes the procedures for using the Process Control Manager 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: <ul style="list-style-type: none"> <li>▶ Specifies the Trading Partnership code or file name that the data manager is to use to identify data</li> <li>▶ Tells the data manager what to do with the data it has identified.</li> </ul>
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: <ul style="list-style-type: none"> <li>▶ Routing data</li> <li>▶ Invoking scripts</li> <li>▶ Archiving data</li> <li>▶ Handling data errors.</li> </ul>
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.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
flow type	<p>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.</p> <p><b>Examples</b></p> <ul style="list-style-type: none"> <li>▶ Standard-to-standard</li> <li>▶ Standard-to-application</li> <li>▶ Application-to-application</li> <li>▶ Application-to-standard</li> </ul>
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 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.

### Reference

For more information about configuration records, see the [Working with Configuration Records](#) chapter in this guide.

### WARNING

**You cannot access and edit individual configuration records with the Gentran:Server Process Control Manager product.**

**You cannot use EC Workbench to modify flow components you created with the PCM Wizard.**

---

**(Continued on next page)**

---

### 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 Gentran:Server for UNIX, you can run the Process Control Manager once in a session for each host and environment combination.
- If you start 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. Gentran:Server restricts editing options to the first Client session you opened.

### Reference

See the [Running Multiple Sessions](#) topic in the *Gentran:Server for UNIX Getting Started Guide* for more information about process limitations and file locking in multiple sessions.

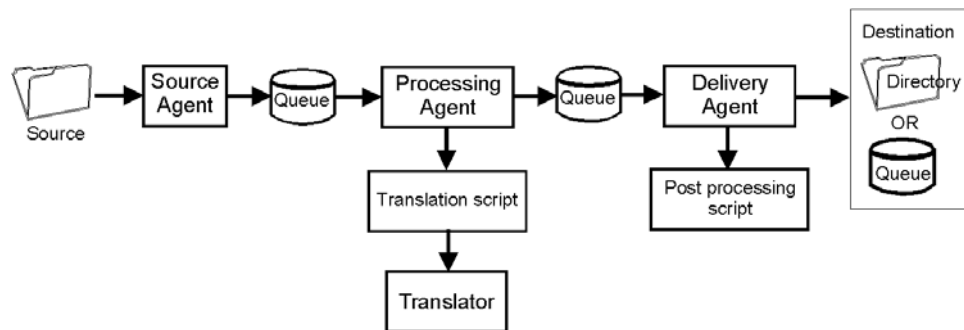
---



## 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.

(Continued on next page)

**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 Gentran:Server XML translation option or 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.

Task	Description	
1	Create the supporting files.  <b>Reference</b> See <a href="#">Creating the Supporting Files</a> .	
2	Name the new process flow.  <b>Reference</b> See <a href="#">How to Name and Describe the Flow</a> .	
3	Set up the data managers (agents) for the process flow. The procedure depends on the format of the input (source) file.	
	<b>IF the input file format is...</b>	<b>THEN see this section...</b>
	An EDI standard	<a href="#">Creating an Inbound Flow</a>
	An application	<a href="#">Creating an Outbound Application Flow</a>
	XML (and you have the XML translation option)	<a href="#">Creating an XML Flow</a>
An NCPDP standard	<a href="#">Creating an Inbound NCPDP Flow</a>	
4	Complete the process flow.  <b>Reference</b> See <a href="#">Completing a Flow</a> .	

(Continued on next page)

**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 PCM 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.  <b>Reference</b> See <a href="#">Creating the Supporting Files</a> .
2	Name the flow and select the flow type.  <b>Reference</b> See <a href="#">How to Name and Describe the Flow</a> .

---

## 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.

---

(Continued on next page)

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

Task	Description
1	<p>Do you want the source data manager to scan a queue to detect files?</p> <ul style="list-style-type: none"> <li>▶ If YES, create the queue and then continue with Task 2.</li> <li>▶ If NO, go to Task 2.</li> </ul> <p><b>Reference</b> For instructions on creating queues, see the chapter <a href="#">Using Queues</a> in this guide.</p>
2	<p>Create the destination you want to use in the process flow. This can be any of the following:</p> <ul style="list-style-type: none"> <li>▶ Directory file name</li> </ul> <p>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.</p> <ul style="list-style-type: none"> <li>▶ Queue</li> <li>▶ Trading Partnership record</li> <li>▶ Trading Partnership category.</li> </ul> <p><b>References</b></p> <ul style="list-style-type: none"> <li>▶ For instructions on creating directories, refer to your UNIX manuals.</li> <li>▶ For instructions on creating queues, see the chapter <a href="#">Using Queues</a> in this guide.</li> <li>▶ For instructions on creating Trading Partnership records and Trading Partnership categories, see your <i>Gentran:Server for UNIX and Workstation Application Integration User's Guide</i>.</li> </ul>
3	<p>Do you want to run a Gentran:Server script after files are processed?</p> <ul style="list-style-type: none"> <li>▶ If YES, create the script and move it to the <code>./script</code> directory.</li> <li>▶ If NO, you are ready to create the process flow.</li> </ul> <p><b>References</b> For instructions on creating scripts, see the chapter <a href="#">Working with Scripts</a> in this guide.</p>

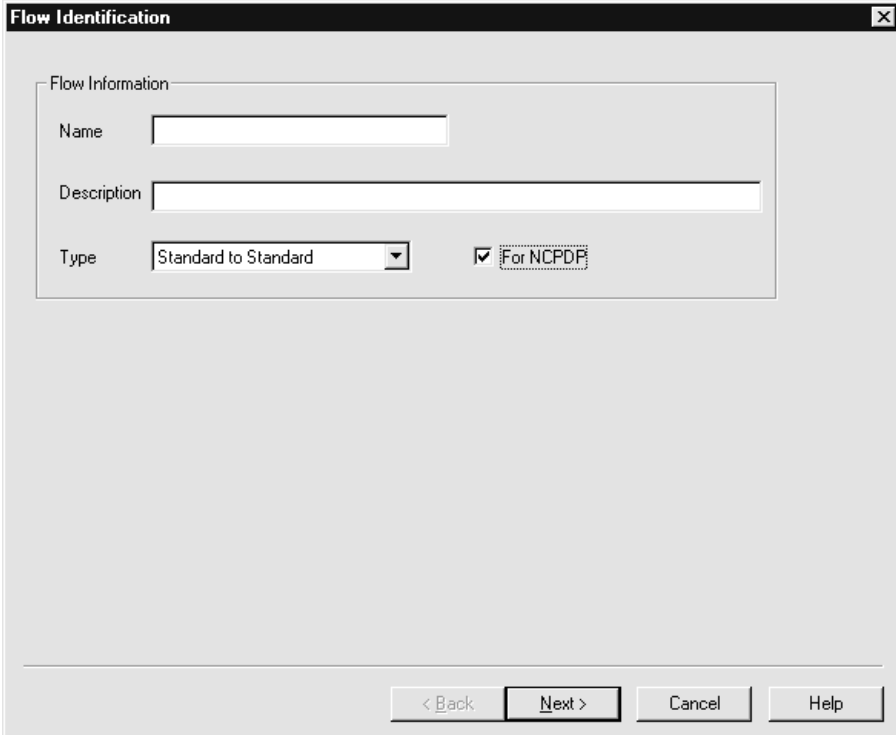
## Flow Identification Dialog Box

### 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 illustration shows the Flow Identification dialog box.



The screenshot shows a dialog box titled "Flow Identification" with a close button (X) in the top right corner. The dialog box contains a "Flow Information" section with the following fields:

- Name: A text input field.
- Description: A text input field.
- Type: A dropdown menu currently showing "Standard to Standard".
- For NCPDF: A checked checkbox.

At the bottom of the dialog box, there are four buttons: "< Back", "Next >", "Cancel", and "Help".



**Flow Identification fields and functions**






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

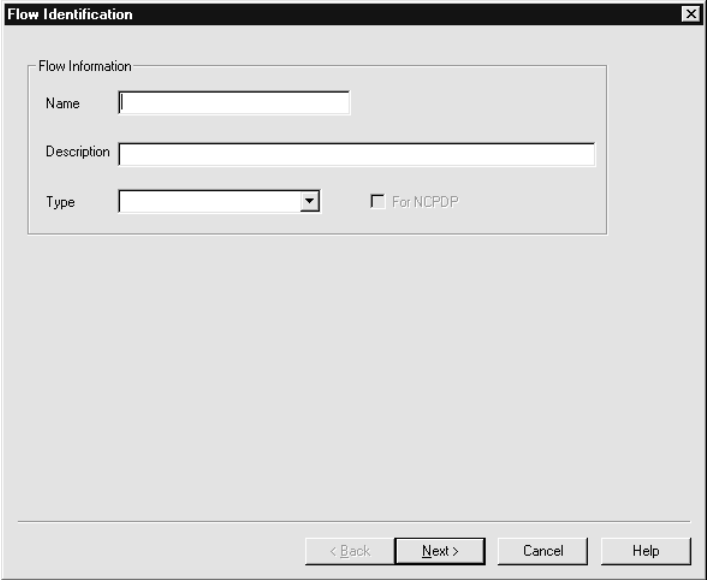
Field	Function
Name	Defines the name of the process flow. Maximum number of characters is 15. Do not use spaces in the name.
Description	Describes the process flow. Optional.
Type	Enables you to select type of process flow. <ul style="list-style-type: none"> <li>▶ Standard-to-standard</li> <li>▶ Standard-to-application</li> <li>▶ Application-to-standard</li> <li>▶ Application-to-application</li> </ul> <p><b>Note</b> If you have optional Gentran:Server products, such as the XML or SAP translation option you have additional flow types available.</p>
For NCPDP	Enables you to create an Inbound NCPDP Flow. <p><b>Note</b> Select the check box only if the source file is in an NCPDP format.</p>

## 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	<p>Click the <b>PCM</b> button on the Gentran:Server client toolbar to start the Process Control Manager wizard.</p> <div data-bbox="987 877 1068 953" style="text-align: center;">  </div> <p><b>System Response</b> Gentran:Server displays a tree that shows all the existing flows. This example has only one flow.</p> <div data-bbox="659 1121 1386 1318" style="border: 1px solid black; padding: 5px;"> <p>- GENTRAN:Server - Process Control Manager</p> <p>File Edit View Help</p> <p>  </p> <p>+  2nd flow - PCM test</p> </div> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
2	<p>Click <b>New</b> on the File menu to start the flow creation wizard.</p> <p><b>System Response</b> Gentran:Server displays the Flow Identification dialog box.</p> 
3	<p>Complete the boxes.</p> <p><b>WARNING</b></p> <p><b>You must name the flow and select a flow type. The description is optional, but we recommend that you include it.</b></p> <p><b>You cannot change the flow's name after you create and save the flow.</b></p>
4	<p>Click <b>Next</b> to continue to the Source setup dialog box.</p> <p><b>Reference</b> See the appropriate section in this chapter:</p> <ul style="list-style-type: none"> <li>▶ <a href="#">Creating an Inbound Flow</a></li> <li>▶ <a href="#">Creating an Outbound Application Flow</a></li> <li>▶ <a href="#">Creating an XML Flow</a></li> <li>▶ <a href="#">Creating an Inbound NCPDP Flow.</a></li> </ul>

# 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 Gentrans: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, **lfrtran**, 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: <ul style="list-style-type: none"> <li>▶ Receives EDI files</li> <li>▶ Splits EDI files by trading partner</li> <li>▶ Processes data and routes EDI data to the processing agent.</li> </ul>

(Continued on next page)

<b>(Contd) Stage</b>	<b>Description</b>
2	The processing agent runs the translation program, <b>lftran</b> , which translates the data and routes it to a delivery agent.
3	The delivery agent runs an embedded script named <code>&lt;dmname&gt;_gen_xltr.scr</code> . 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.

The screenshot shows the 'Source' dialog box with the following configuration:

- Agent Name:
- New File Detection:
  - Queue
  - Directory Scan
- Queue Name:
- Source Directory:
- Scan Frequency:
  - Once
  - Periodically
- Periodically settings: Every  Hour(s)  Minute(s)  Second(s)
- Split Files By:
  - Interchange
  - Group
  - Transaction Set

Buttons at the bottom: < Back, Next >, Cancel, Help

(Continued on next page)

**Source setup fields and functions**

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

Field	Function
Agent Name	<p>Defines the name of the source data manager. The maximum size is 4 characters.</p> <p><b>Note</b> 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.</p>
<b>New File Detection</b>	
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	<p>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.</p> <p><b>Example</b> ./sr03</p> <p><b>Note</b> 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.</p>
<b>Scan Frequency</b>	
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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Function</b>
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.
<b>Split Files By</b>	
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.

---

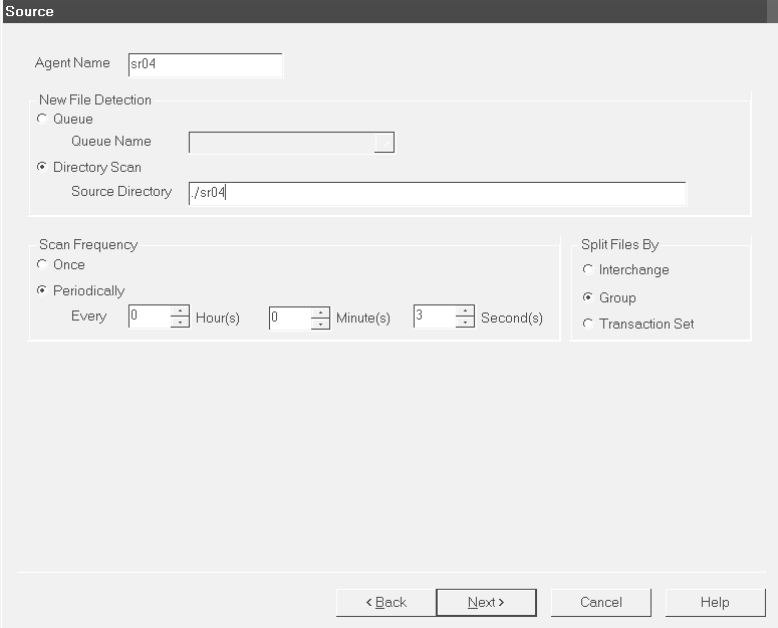


# How to Set Up the Source Agent (Inbound Flow)

**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	<p>Type the name of the source data manager in the Agent Name box.</p> <p><b>CAUTION</b>  <b>Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.</b></p>  <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
2	Select either <b>Queue</b> or <b>Directory Scan</b> as the type of source that you want the source data manager to examine for files to process.
3	<p>Did you select <b>Queue</b> in Step 2?</p> <ul style="list-style-type: none"> <li>▶ If YES, select the name of the queue from the drop-down list and continue with Step 4.</li> <li>▶ If NO, (the source is a directory), type the relative path name to the directory in the text box and continue with Step 4.</li> </ul> <p><b>WARNING</b></p> <p><b>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.</b></p>
4	<p>Click <b>Once</b> or <b>Periodically</b> to select the scan frequency.</p> <p><b>Note</b> The scan frequency you select applies to every data manager in the flow.</p>
5	<p>Did you select <b>Periodically</b> in Step 3?</p> <ul style="list-style-type: none"> <li>▶ 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.</li> <li>▶ If NO, continue with Step 5.</li> </ul>
6	Click <b>Interchange</b> , <b>Group</b> , or <b>Transaction Set</b> to select how you want the data manager to group routed data.
7	<p>Click <b>Next</b> to continue to the Processing Agent dialog box.</p> <p><b>Reference</b> See <a href="#">How to Set Up the Processing Agent (Inbound Flow)</a></p>

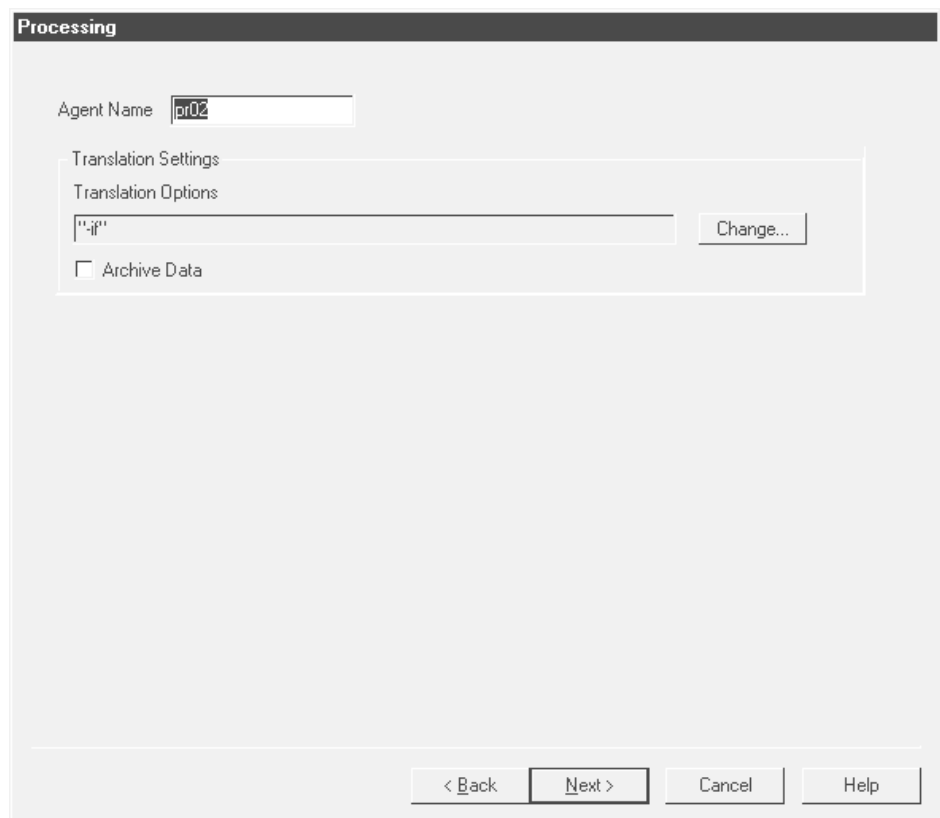
## Processing Agent Dialog Box (Inbound Flow)

### 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.



The screenshot shows a dialog box titled "Processing". It contains the following elements:

- An "Agent Name" text box containing the value "pr02".
- A "Translation Settings" section containing:
  - A "Translation Options" text box containing the value """.
  - A "Change..." button to the right of the "Translation Options" text box.
  - An unchecked checkbox labeled "Archive Data".
- At the bottom of the dialog box, there are four buttons: "< Back", "Next >", "Cancel", and "Help".

(Continued on next page)

### Processing Agent fields and functions

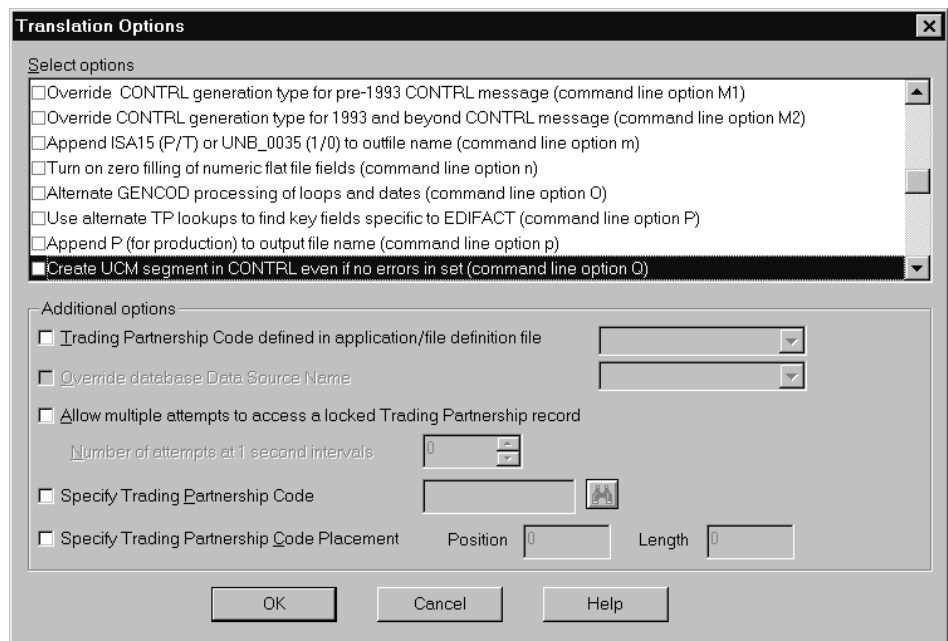
This table describes the fields of the Processing Agent dialog box and their functions.

Field	Function
Agent Name	<p>Defines the name of the processing data manager.</p> <p><b>Note</b> 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.</p>
<b>Translation Settings</b>	
Translation Options	Displays the currently selected translation options.
Change	<p>Displays the Translation Options dialog box.</p> <p><b>Reference</b> See the <a href="#">Translation Options Dialog Box</a> topic in this section.</p>
Archive Data	<p>Runs the <b>ediarc</b> program in the translation script. Archives the EDI-standard version of the file.</p> <p><b>Reference</b> See the <a href="#">ediarc</a> topic in the <a href="#">Command Reference</a> chapter of the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i> for more information about <b>ediarc</b>.</p> <p>See the <a href="#">Archiving Translation Data</a> chapter in the <i>Gentran:Server for UNIX and Workstation Application Integration User's Guide</i> for information about archiving translation data.</p>

## Translation Options Dialog Box

**Introduction** 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.



(Continued on next page)

### Translation Option fields and functions

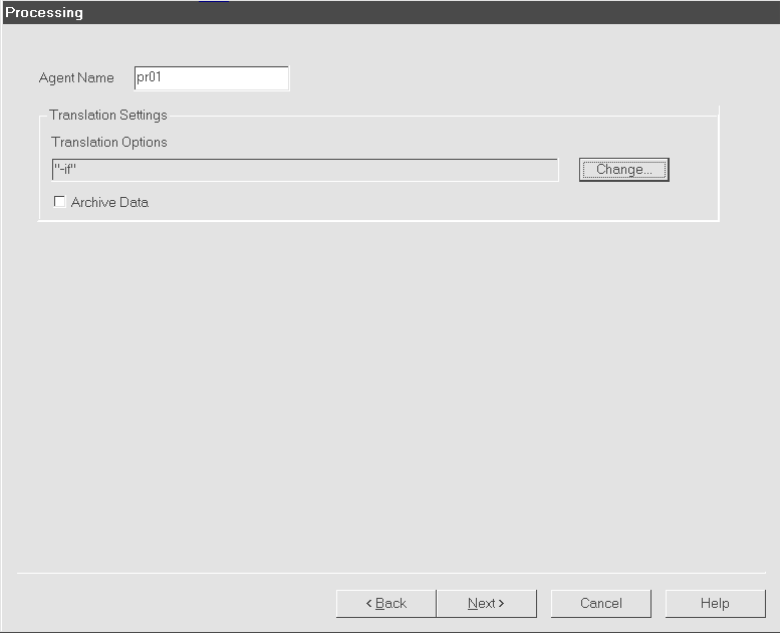
This table describes the fields of the Translation Options dialog box and their functions.

Field	Function
Select Options	Allows you to select parameters for the translation process.  <b>Reference</b> 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.
Override database Data Source Name	For Visual Mapper only, enables you to replace the ODBC DSN used to create the application file with the one you want to use for the current translation.  <b>Note</b> Your Gentran:Server system must have the optional ODBC translation capabilities.
Allow multiple attempts to access a locked Trading Partnership record	Sets the number of attempts 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	<p>Type the name of the processing data manager in the Agent Name text box.</p> <p><b>CAUTION</b>  <b>Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.</b></p> 
2	<p>Do you want to change the translation options?</p> <ul style="list-style-type: none"> <li>▶ If YES, click the <b>Change</b> button and complete the Translation Options dialog box.</li> <li>▶ If NO, continue with Step 3.</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Do you want the translation script to run the <b>ediarc</b> program?</p> <ul style="list-style-type: none"><li>▶ If YES, select <b>Archive Data</b> and then click <b>Next</b> to continue to the Delivery Agent dialog box.</li><li>▶ If NO, click <b>Next</b> to continue to the Delivery Agent dialog box.</li></ul> <p><b>Note</b> The <b>ediarc</b> program archives translation data.</p> <p><b>Reference</b> See <a href="#">How to Set Up the Delivery Agent (Inbound Flow)</a>.</p>

---



## 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.

The screenshot shows the 'Delivery' dialog box with the following fields and options:

- Agent Name:** Text field containing 'dl07'.
- Results Directory:** A group box containing:
  - Queue Output (with a dropdown menu)
  - Set Type
  - TP Code
  - Categories (with a dropdown menu)
  - User-Defined (with a text field)
- Results File:** A group box containing:
  - Set Type (with a text field)
  - User-Defined (with a text field)
  - TP Code
  - Categories (with a dropdown menu)
- Post Processing:** A group box containing:
  - Script Name:** Text field with a dropdown menu.
  - Run Script After:** Radio buttons for  Each Document and  All Documents.

At the bottom of the dialog box are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

(Continued on next page)

### Delivery agent fields and functions

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

Field	Function
Agent Name	<p>Defines the name of the delivery data manager.</p> <p><b>Note</b> 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.</p>
<b>Results Directory</b>	
Queue Output	<p>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.</p>
Set Type	<p>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.</p> <p><b>Note</b> This option is disabled if the output document is in an XML format.</p>
TP Code	<p>Selects Trading Partnership Code as the Results Directory (destination directory) in the configuration records.</p> <p>The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.</p>
Categories	<p>Enables you to specify a Trading Partnership category as the Results Directory (destination directory) in the configuration records.</p> <p>Select the category from the drop-down list box that is next to the Categories option.</p> <p>The Process Control Manager substitutes the actual category value in the configuration records.</p>

(Continued on next page)

<b>(Contd) Field</b>	<b>Function</b>
User Defined	<p>Enables you to specify the Results Directory (destination directory) for the configuration records.</p> <p>Enter the path in the text box that is next to the User Defined option.</p>
<b>Results File</b>	
Set Type	<p>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.</p> <p><b>Note</b> This option is disabled if the output document is in an XML format.</p>
TP Code	<p>Selects Trading Partnership Code as the Results File (output file name) in the configuration records.</p> <p>The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.</p>
Categories	<p>Enables you to specify a Trading Partnership category as the Results File (output file name) in the configuration records.</p> <p>Select the category from the drop-down list box that is next to the Categories option.</p> <p>The Process Control Manager substitutes the actual category value in the configuration records.</p>
<b>Post Processing</b>	
User Defined	<p>Enables you to specify the Results File (output file name) for the configuration records.</p> <p>Enter the path in the text box that is next to the User Defined option.</p>
Script Name	<p>Enables you to enter or select the name of the script you want to run after this data manager has processed the files.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Field</b>	<b>Function</b>
Each Document	Executes the post-processing Gentran:Server script after each document has been processed.
All Documents	Executes the post-processing 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.

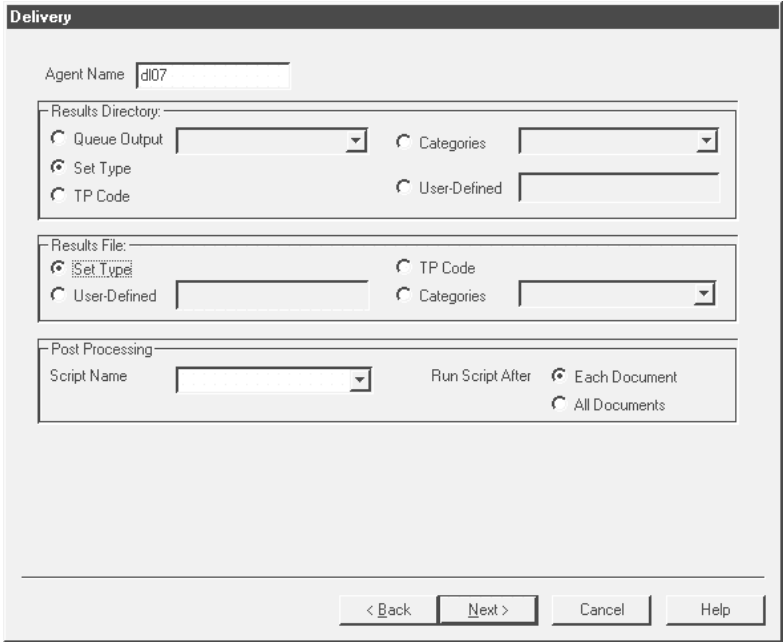
You can set:

- ▶ 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) Gentrans: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.

---

(Continued on next page)

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

Step	Action
1	<p>Type the name of the delivery data manager in the <b>Agent Name</b> box.</p> <p><b>CAUTION</b>  <b>Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.</b></p> 
2	<p>Choose the Results Directory by clicking <b>Set Type</b>, <b>TP Code</b>, <b>Categories</b>, or <b>User-Defined</b> to select the symbolic destination directory, or typing the path to the directory for the output.</p> <p><b>Comment</b>  The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. PCM creates directories if they do not exist.</p> <p><b>Example</b>  If you select <b>TP Code</b>, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Did you select <b>Categories</b> in Step 2?</p> <ul style="list-style-type: none"> <li>▶ If YES, select a category from the drop-down list.</li> <li>▶ If NO, continue with Step 4.</li> </ul>
4	<p>Did you select <b>User-Defined</b> in Step 2?</p> <ul style="list-style-type: none"> <li>▶ If YES, type the path to the directory in the text box that is below the User-Defined option.</li> <li>▶ If NO, continue with Step 5.</li> </ul>
5	<p>Choose the Results File name by clicking <b>Set Type, TP Code, Categories, or User-Defined</b> to select the symbolic destination file name.</p> <p><b>Comment</b> The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records.</p> <p><b>Example</b> If you select <b>TP Code</b>, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.</p>
6	<p>Did you select <b>Categories</b> in Step 5?</p> <ul style="list-style-type: none"> <li>▶ If YES, select a category from the drop-down list and continue with Step 7.</li> <li>▶ If NO, continue with Step 7.</li> </ul>
7	<p>Did you select <b>User-Defined</b> in Step 5?</p> <ul style="list-style-type: none"> <li>▶ If YES, type the complete file name in the text box that is below the <b>User-Defined</b> option and continue with Step 8.</li> <li>▶ If NO, continue with Step 8.</li> </ul>
8	<p>Do you want to execute a script after the translation process?</p> <ul style="list-style-type: none"> <li>▶ If YES, select the name of the script from the <b>Script Name</b> drop-down list and continue with Step 9.</li> <li>▶ If NO, continue with Step 9.</li> </ul>
9	<p>Click the <b>Each document</b> or <b>All documents</b> option to select when the system runs the script.</p>
10	<p>Click <b>Next</b> to continue to the Error Handling dialog box.</p> <p><b>Reference</b> See <a href="#">How to Set Up Error Handling Instructions</a> for instructions on completing the Error Handling dialog box.</p>

# 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 Gentrans: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.

---

(Continued on next page)



**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 <i>&lt;dmname&gt;_gen_xltr.scr</i> . 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.

The screenshot displays the 'Source' dialog box with the following configuration:

- Agent Name:** sr02
- New File Detection:**
  - Queue (Queue Name: [empty])
  - Directory Scan (Source Directory: ./source)
- Scan Frequency:**
  - Once
  - Periodically (Every: 0 Hour(s), 0 Minute(s), 3 Second(s))
- Split Files By:**
  - Interchange
  - Group
  - Transaction Set
- Filename Prefix is:**
  - Trading Partner Code
  - Application Filename

Navigation buttons at the bottom: < Back, Next >, Cancel, Help.

(Continued on next page)

**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.  <b>Note</b> 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.
<b>New File Detection</b>	
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.
<b>Scan Frequency</b>	
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.

(Continued on next page)

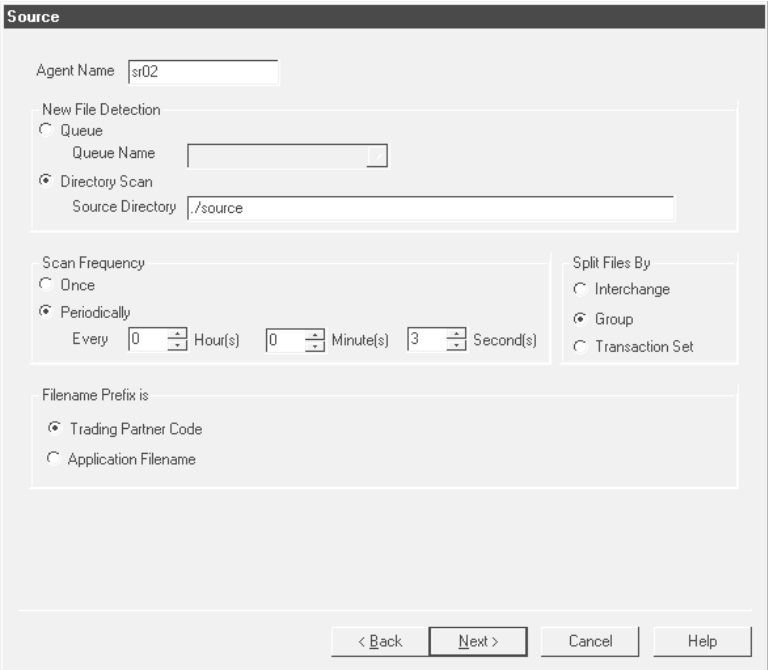
<b>(Contd) Field</b>	<b>Function</b>
Second(s)	Defines, in seconds, the frequency with which the data managers in the flow scan for new files. Value range is 0 to 59.  The default value is 3 seconds.
<b>Split Files By</b>	
Interchange	Selects interchange code as the splitting method to route files.
Group	Selects group code as the splitting method to route files.
Transaction Set	Selects transaction set as the splitting method to route files.
<b>Filename Prefix is</b>	
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	<p>Type the name of the source data manager in the Agent Name dialog box.</p> <p><b>CAUTION</b>  <b>Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.</b></p>  <p style="text-align: right;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
2	<p>Select either <b>Queue</b> or <b>Directory Scan</b> as the type of source that you want the source data manager to examines for files to process.</p> <p>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.</p>
3	<p>Click <b>Once</b> or <b>Periodically</b> to select the scan frequency.</p> <p><b>Note</b> The scan frequency you select applies to every data manager in the flow.</p>
4	<p>Did you select <b>Periodically</b> in Step 3?</p> <ul style="list-style-type: none"> <li>▶ If YES, complete the <b>Hour(s)</b>, <b>Minute(s)</b> and <b>Second(s)</b> boxes to select the frequency with which the data manager awakens and scans the queue or directory.</li> <li>▶ If NO, continue with Step 5.</li> </ul>
5	<p>Click <b>Interchange</b>, <b>Group</b>, or <b>Transaction Set</b> to select how the data manager groups routed data.</p>
6	<p>Select the filename prefix: <b>Trading Partner Code</b> or <b>Application Filename</b>.</p>
7	<p>Click <b>Next</b> to continue to the Processing Agent dialog box.</p> <p><b>Reference</b> See the <a href="#">How to Set Up the Processing Agent (Outbound Application)</a>.</p>

## 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.

The screenshot shows a dialog box titled "Processing". It contains the following elements:

- Agent Name:** A text field containing "p02".
- Translation Settings:** A section containing:
  - Translation Options:** A text field containing '"-of"' and a "Change..." button.
  - Archive Data:** A checkbox that is currently unchecked.
- Navigation Buttons:** At the bottom, there are four buttons: "< Back", "Next >", "Cancel", and "Help".

(Continued on next page)

### 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	<p>Defines the name of the processing data manager.</p> <p><b>Note</b> 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.</p>
<b>Translation Settings</b>	
Translation Options	Displays the currently selected translation options.
Change	<p>Displays the Translation Options dialog box.</p> <p><b>Reference</b> See the <a href="#">Translation Options Dialog Box (Outbound Application)</a> topic in this section.</p>
Archive Data	<p>Runs the <b>ediarc</b> program in the translation script. Archives the EDI-standard version of the file.</p> <p><b>Reference</b> See the <b>ediarc</b> topic in the <a href="#">Command Reference</a> chapter of the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i> for more information about <b>ediarc</b>.</p> <p>See the <a href="#">Archiving Translation Data</a> chapter in the <i>Gentran:Server for UNIX and Workstation Application Integration User's Guide</i> for information about archiving translation data.</p>



## Translation Options Dialog Box (Outbound Application)

**Introduction** 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 Options**

Select options

- Override CONTRL generation type for pre-1993 CONTRL message (command line option M1)
- Override CONTRL generation type for 1993 and beyond CONTRL message (command line option M2)
- Append ISA15 (P/T) or UNB\_0035 (1/0) to outfile name (command line option m)
- Turn on zero filling of numeric flat file fields (command line option n)
- Alternate GENCOD processing of loops and dates (command line option O)
- Use alternate TP lookups to find key fields specific to EDIFACT (command line option P)
- Append P (for production) to output file name (command line option p)
- Create UCM segment in CONTRL even if no errors in set (command line option Q)

Additional options

- Trading Partnership Code defined in application/file definition file
- Override database Data Source Name
- Allow multiple attempts to access a locked Trading Partnership record
  - Number of attempts at 1 second intervals
- Specify Trading Partnership Code
- Specify Trading Partnership Code Placement Position  Length

OK Cancel Help

(Continued on next page)

### Translation Option fields and functions

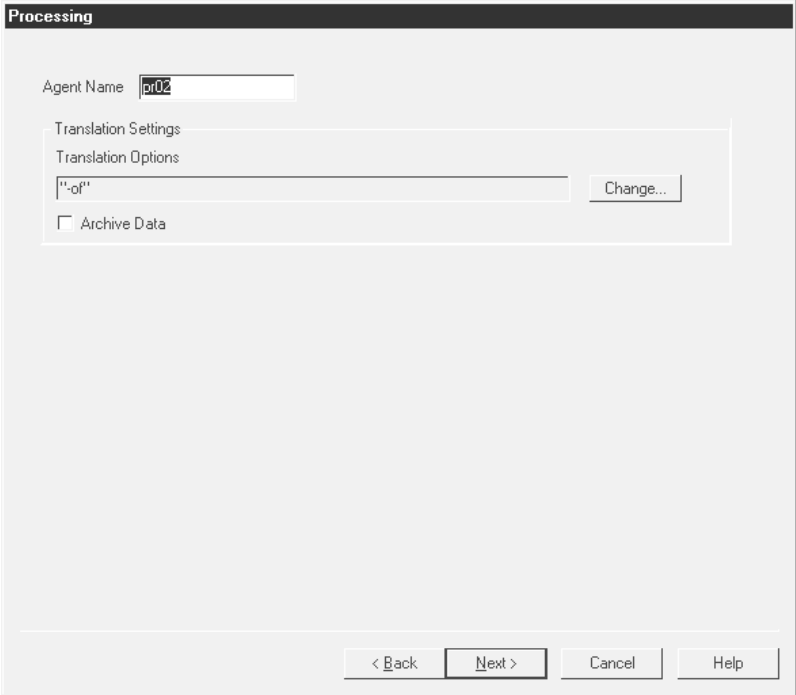
This table describes the fields of the Translation Options dialog box and their functions.

Field	Function
Select options	Enables you to select the translation options you want to apply to this flow.  <b>Reference</b> For a list of translation options, see the <a href="#">Iftran Overview</a> topic in the <a href="#">Command Reference</a> chapter of the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i> .
Trading Partnership code defined in application/file definition file	Enables you to select the application or file definition file. Used only for outbound translations.
Override database Data Source Name	For Visual Mapper only, enables you to replace the ODBC DSN used to create the application file with the one you want to use for the current translation.  <b>Note</b> Your Gentran:Server system must have the optional ODBC translation capabilities.
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 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	<p>Type the name of the processing data manager in the Agent Name box.</p>  <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
2	Do you want to change the translation options? <ul style="list-style-type: none"><li>▶ If YES, click the <b>Change</b> button and complete the Translation Options dialog box.</li><li>▶ If NO, continue with Step 3.</li></ul>
3	Do you want the translation script to run the <b>ediarc</b> program? <ul style="list-style-type: none"><li>▶ If YES, select <b>Archive Data</b> and then click <b>Next</b> to continue to the Delivery Agent dialog box.</li><li>▶ If NO, click <b>Next</b> to continue to the Delivery Agent dialog box.</li></ul> <p><b>Note</b> The <b>ediarc</b> program archives translation data.</p> <p><b>Reference</b> See the <a href="#">How to Set Up the Delivery Agent (Outbound Application)</a>.</p>

---

## 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.

The screenshot shows the "Delivery" dialog box with the following fields and options:

- Agent Name:
- Results Directory:
  - Queue Output
  - Set Type
  - TP Code
  - Categories
  - User-Defined
- Results File:
  - Set Type
  - User-Defined
  - TP Code
  - Categories
- Post Processing:
  - Script Name:
  - Run Script After:
    - Each Document
    - All Documents

Buttons at the bottom: < Back, Next >, Cancel, Help

(Continued on next page)

### Delivery Agent fields and functions

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

Field	Function
Agent Name	<p>Defines the name of the delivery data manager.</p> <p><b>Note</b> 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.</p>
<b>Results Directory</b>	
Queue Output	<p>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.</p>
Set Type	<p>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.</p> <p><b>Note</b> This option is disabled if the output document is in an XML format.</p>
TP Code	<p>Selects Trading Partnership Code as the Results Directory (destination directory) in the configuration records.</p> <p>The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.</p>
Categories	<p>Enables you to specify a Trading Partnership category as the Results Directory (destination directory) in the configuration records.</p> <p>Select the category from the drop-down list box that is next to the Categories option.</p> <p>The Process Control Manager substitutes the actual category value in the configuration records.</p>

(Continued on next page)

<b>(Contd) Field</b>	<b>Function</b>
User Defined	<p>Enables you to specify the Results Directory (destination directory) for the configuration records.</p> <p>Enter the path in the text box that is next to the User Defined option.</p>
<b>Results File</b>	
Set Type	<p>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.</p> <p><b>Note</b> This option is disabled if the output document is in an XML format.</p>
TP Code	<p>Selects Trading Partnership Code as the Results File (output file name) in the configuration records.</p> <p>The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.</p>
Categories	<p>Enables you to specify a Trading Partnership category as the Results File (output file name) in the configuration records.</p> <p>Select the category from the drop-down list box that is next to the Categories option.</p> <p>The Process Control Manager substitutes the actual category value in the configuration records.</p>
User Defined	<p>Enables you to specify the Results File (output file name) for the configuration records.</p> <p>Enter the path in the text box that is next to the User Defined option.</p>
<b>Post Processing</b>	
Script Name	<p>Enables you to enter or select the name of the script you want to run after this data manager has processed the files.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Field</b>	<b>Function</b>
Each Document	Executes the post-processing Gentran:Server script after each document has been processed.
All Documents	Executes the post-processing 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.

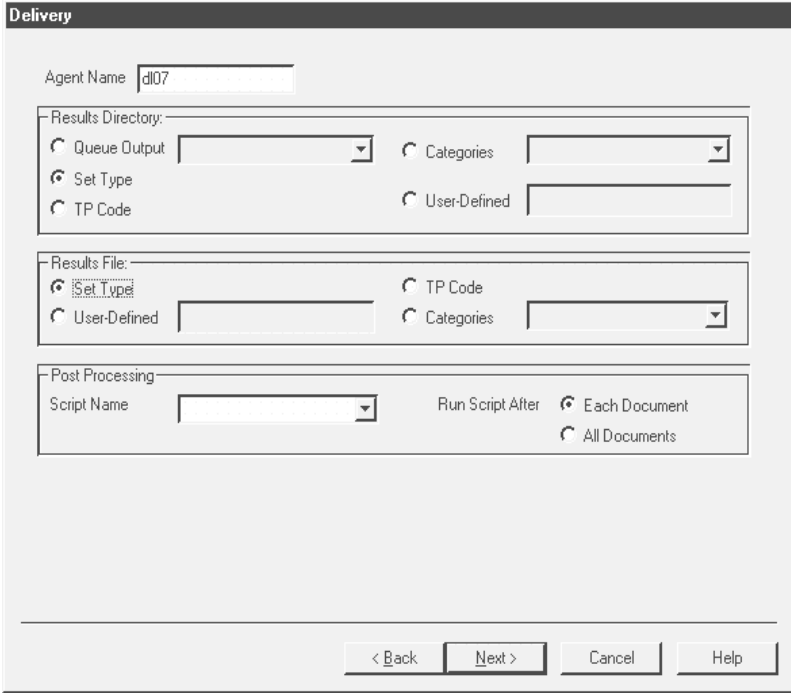
You can set:

- ▶ 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 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.

---

(Continued on next page)

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

Step	Action
1	<p>Type the name of the delivery data manager in the Agent Name box.</p> 
2	<p>Choose the Results Directory by clicking <b>Set Type</b>, <b>TP Code</b>, <b>Categories</b>, or <b>User-Defined</b> to select the symbolic destination directory, or type the path to the directory for the translation output.</p> <p><b>Comment</b> The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records.</p> <p><b>Example</b> If you select <b>TP Code</b>, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.</p>
3	<p>Did you select <b>Categories</b> in Step 2?</p> <ul style="list-style-type: none"> <li>▶ If YES, select a category from the drop-down list.</li> <li>▶ If NO, continue with Step 4.</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
4	<p>Did you select <b>User-Defined</b> in Step 2?</p> <ul style="list-style-type: none"> <li>▶ If YES, type the path to the directory in the text box that is below the <b>User-Defined</b> option.</li> <li>▶ If NO, continue with Step 5.</li> </ul>
5	<p>Choose the Results File name by clicking <b>Set Type, TP Code, Categories,</b> or <b>User-Defined</b> to select the symbolic destination file name.</p> <p><b>Comment</b> The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records.</p> <p><b>Example</b> If you select TP Code, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.</p>
6	<p>Did you select <b>Categories</b> in Step 5?</p> <ul style="list-style-type: none"> <li>▶ If YES, select a category from the drop-down list and continue with Step 7.</li> <li>▶ If NO, continue with Step 7.</li> </ul>
7	<p>Did you select <b>User-Defined</b> in Step 5?</p> <ul style="list-style-type: none"> <li>▶ If YES, type the complete file name in the text box that is below the <b>User-Defined</b> option and continue with Step 8.</li> <li>▶ If NO, continue with Step 8.</li> </ul>
8	<p>Do you want to run a post-processing script?</p> <ul style="list-style-type: none"> <li>▶ If YES, select the name of the script from the Script Name drop-down list and continue with Step 9.</li> <li>▶ If NO, continue with Step 9.</li> </ul>
9	<p>Click the <b>Each document</b> or <b>All documents</b> option to select when the system runs the script.</p>
10	<p>Click <b>Next</b> to continue to the Error Handling dialog box.</p> <p><b>Reference</b> See <a href="#">How to Set Up Error Handling Instructions</a> for instructions on completing the Error Handling dialog box.</p>

# 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.

---

(Continued on next page)

---

**Routing**

This table describes the routing in an XML flow.

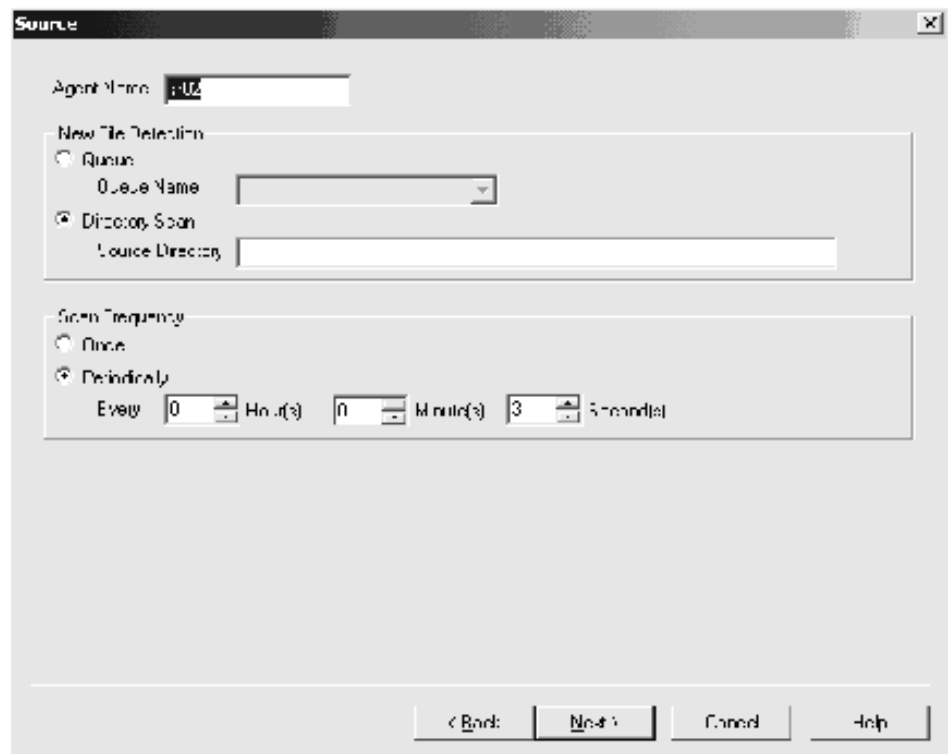
<b>Stage</b>	<b>Description</b>
1	The source agent, which is an XML data manager, receives XML data and routes it to the processing agent.
2	The processing agent, which is a translation data manager, runs the translator. After translation, the flow routes the translated data to the delivery agent.
3	The delivery agent, which is a translation data manager, runs an imbedded script named <code>&lt;dmname&gt;_gen_xltr.scr</code> that runs to generate a Life Cycle event record for auditing purposes.  <b>Note</b> 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.

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



(Continued on next page)

### 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.  <b>Note</b> 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.
<b>New File Detection</b>	
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.
<b>Scan Frequency</b>	
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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Function</b>
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.

---

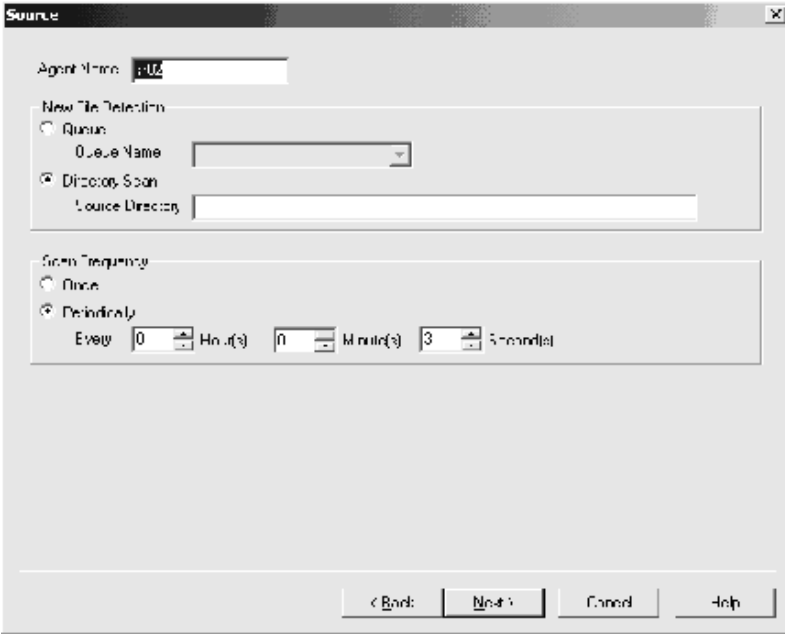


# How to Set Up the Source Agent (XML Flow)

**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	<p>Type the name of the source data manager in the Agent Name box.</p> <p><b>CAUTION</b>  <b>Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.</b></p>  <p style="text-align: right;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
2	<p>Do you want the data manager to scan a queue for files?</p> <ul style="list-style-type: none"><li>▶ If YES, click <b>Queue</b> and then select the name of the queue from the drop-down list.</li><li>▶ If NO, (you want the data manager to scan a directory), click <b>Directory Scan</b> and then type the relative path name to the directory in the text box.</li></ul>
3	<p>Click <b>Once</b> or <b>Periodically</b> to select the scan frequency.</p> <p><b>Note</b> The scan frequency you select applies to every data manager in the flow.</p>
4	<p>Did you select <b>Periodically</b> in Step 3?</p> <ul style="list-style-type: none"><li>▶ If YES, complete the <b>Hour(s)</b>, <b>Minute(s)</b> and <b>Second(s)</b> boxes to select the frequency with which the data manager awakens and scans the queue or directory.</li><li>▶ If NO, continue with Step 5.</li></ul>
5	<p>Click <b>Next</b> to continue to the Processing Agent dialog box.</p> <p><b>Reference</b> See <a href="#">How to Set Up the Processing Agent (XML Flow)</a>.</p>

---

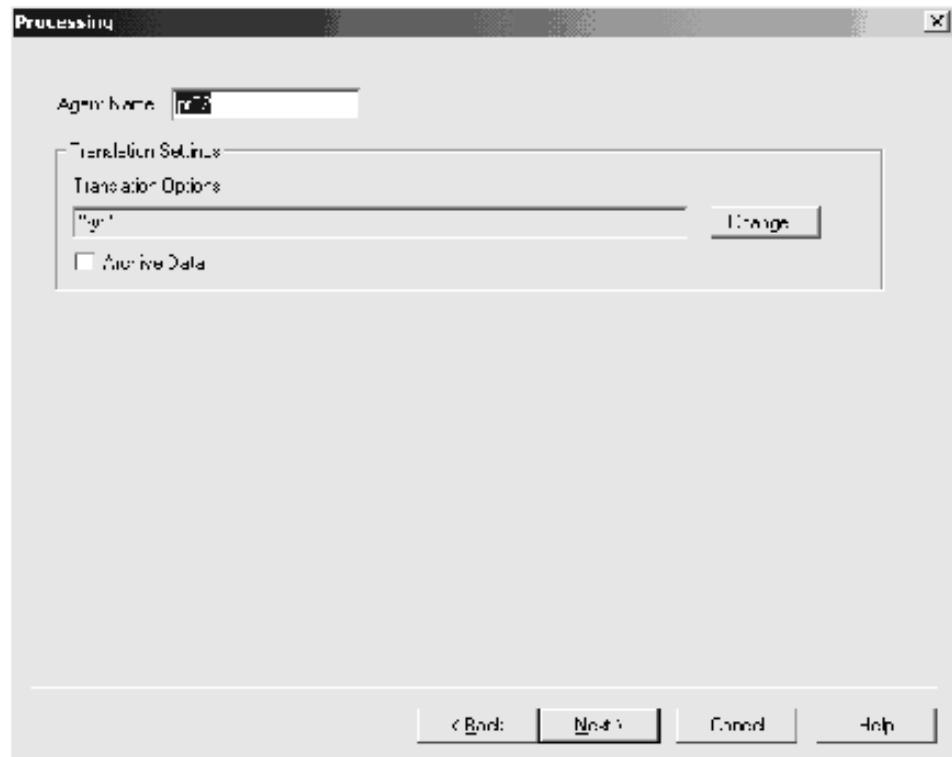
## Processing Agent Dialog Box (XML Flow)

### 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 dialog box

This illustration shows the Processing agent dialog box.



(Continued on next page)

### Processing Agent fields and functions

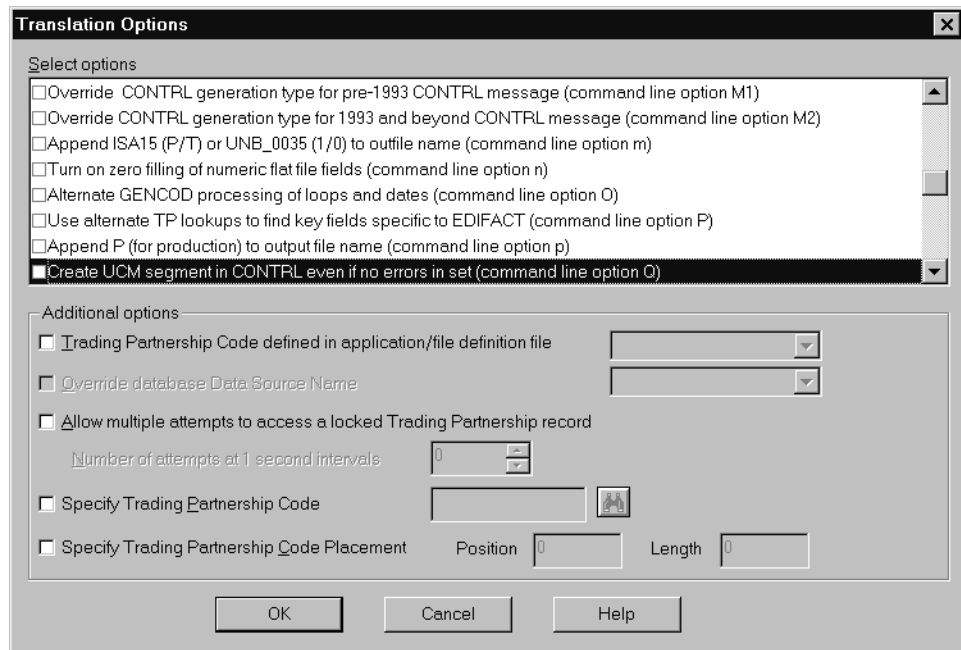
This table describes the fields of the Processing Agent dialog box in an XML flow and their functions.

Field	Function
Agent Name	<p>Defines the name of the processing data manager.</p> <p><b>Note</b> 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.</p>
Translation Options	<p>Displays the currently selected translation options.</p> <p><b>Note</b> When a flow has XML format as the source or destination format, the default value is -yf.</p>
Change	<p>Displays the Translation Options dialog box.</p> <p><b>Reference</b> See the <a href="#">Translation Options Dialog Box (XML Flow)</a> topic in this section.</p>
Archive Data	<p>If the output file is an EDI standard, runs the <b>ediarc</b> program in the translation script to archives the EDI-standard version.</p> <p><b>Reference</b> See the <a href="#">ediarc</a> topic in the <a href="#">Command Reference</a> chapter of the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i> for more information about <b>ediarc</b>.</p> <p>See the <a href="#">Archiving Translation Data</a> chapter in the <i>Gentran:Server for UNIX and Workstation Application Integration User's Guide</i> for information about archiving translation data.</p>

## Translation Options Dialog Box (XML Flow)

**Introduction** 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.



(Continued on next page)

### Translation Option fields and functions

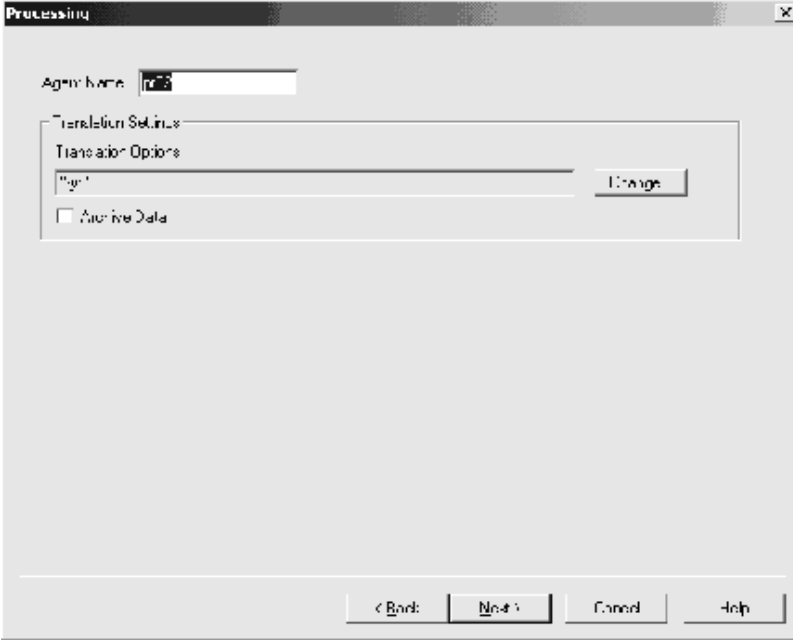
This table describes the fields of the Translation Options dialog box and their functions.

Use this field	To
Select options	Select the translation options you want to apply to this flow.  <b>Reference</b> For a list of translation options, see the <a href="#">Iftran Overview</a> topic in the <a href="#">Command Reference</a> chapter of the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i> .
Trading Partnership code defined in application/file definition file	Select the application or file definition file. Used only for outbound translations.
Override database Data Source Name	For Visual Mapper only, enables you to replace the ODBC DSN used to create the application file with the one you want to use for the current translation.  <b>Note</b> Your Gentran:Server system must have the optional ODBC translation capabilities.
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	<p>Type the name of the processing data manager in the Agent Name box.</p> 
2	<p>Do you want to change the translation options?</p> <ul style="list-style-type: none"> <li>▶ If YES, click the <b>Change</b> button and complete the Translation Options dialog box.</li> <li>▶ If NO, continue with Step 3.</li> </ul> <p><b>Note</b> We recommend that you do not change the translation options.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>If the output file is in an EDI standard format, do you want the translation script to run the <b>ediarc</b> program?</p> <ul style="list-style-type: none"><li>▶ If YES, select <b>Archive Data</b> and then click <b>Next</b> to continue to the Delivery agent dialog box.</li><li>▶ If NO, click <b>Next</b> to continue to the Delivery agent dialog box.</li></ul> <p><b>Note</b> The <b>ediarc</b> program archives EDI translation data.</p>
4	Continue with <a href="#">How to Set Up the Delivery Agent (XML Flow)</a> .

---



## 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.

The screenshot shows the 'Delivery' dialog box with the following fields and options:

- Agent Name:
- Results Directory:
  - Queue Output
  - Set Type
  - TP Code
  - Categories:
  - User-Defined:
- Results File:
  - Set Type
  - User-Defined
  - TP Code:
  - Categories:
- Post Processing:
  - Script Name:
  - Run Script After:
    - Each Document
    - All Documents

Buttons at the bottom: < Back, Next >, Cancel, Help

(Continued on next page)

**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.

The screenshot shows a dialog box titled "Delivery" with the following fields and options:

- Agent Name:** d09
- Results Directory:**
  - Queue Defini... (dropdown)
  - IP Code
  - Categories (dropdown)
  - User-Defined (text field)
- Results File:**
  - User-Defined (text field)
  - IP Code
  - Categories (dropdown)
- Post Processing:**
  - Script Name (dropdown)
  - Script Alter (checkbox)
  - Each Document
  - All Documents

Buttons at the bottom: < Back, Next >, Cancel, Help

(Continued on next page)

### 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	<p>Define the name of the delivery data manager.</p> <p><b>Note</b> 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.</p>
Queue Output	<p>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.</p>
Set Type	<p>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.</p> <p><b>Note</b> This option is disabled and does not appear on the dialog box if the output document is in an XML format.</p>
TP Code	<p>Select Trading Partnership Code as the Results Directory (destination directory) in the configuration records.</p> <p>The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.</p>
Categories	<p>Specify a Trading Partnership category as the Results Directory (destination directory) in the configuration records.</p> <p>Select the category from the drop-down list box that is next to the Categories option.</p> <p>The Process Control Manager substitutes the actual category value in the configuration records.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Use this field</b>	<b>To</b>
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.  <b>Note</b> 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 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 Gentran:Server script after each document has been processed.
All Documents	Execute the post-processing 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.

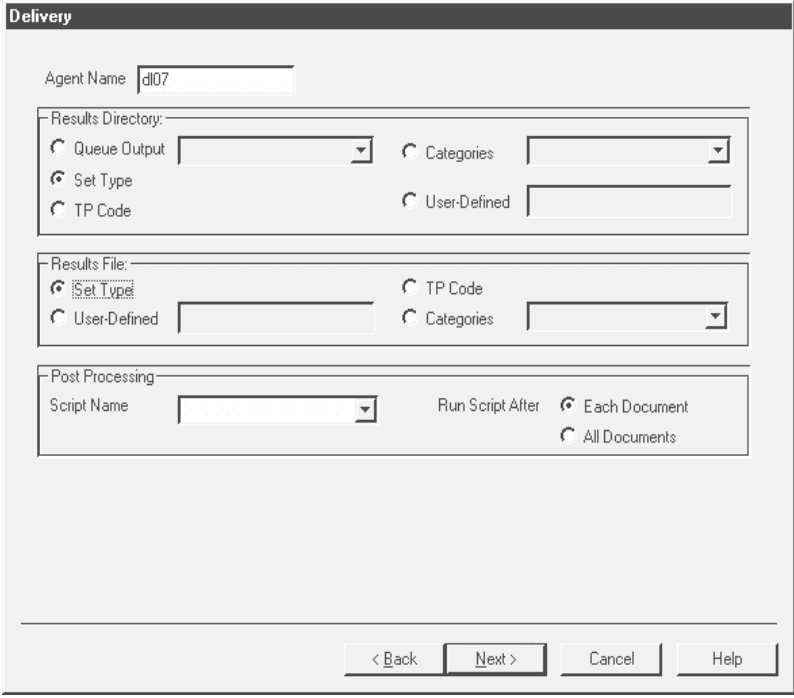
You can set:

- ▶ 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 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.

---

(Continued on next page)

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

Step	Action
1	<p>Type the name of the delivery data manager in the Agent Name box.</p> 
2	<p>Choose the Results Directory by clicking <b>Set Type</b>, <b>TP Code</b>, <b>Categories</b>, or <b>User-Defined</b> to select the symbolic destination directory, or type the path to the directory for the translation output.</p> <p><b>Comments</b> The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records.</p> <p>The <b>Set Type</b> option is not available if the destination format is XML.</p> <p><b>Example</b> If you select <b>TP Code</b>, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	Did you select <b>Categories</b> in Step 2? ▶ If YES, select a category from the drop-down list. ▶ If NO, continue with Step 4.
4	Did you select <b>User-Defined</b> in Step 2? ▶ If YES, type the path to the directory in the text box that is below the <b>User-Defined</b> option. ▶ If NO, continue with Step 5.
5	Choose the Results File name by clicking <b>Set Type, TP Code, Categories, or User-Defined</b> to select the symbolic destination file name.  <b>Comments</b> The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records.  The <b>Set Type</b> option is not available if the destination format is XML.  <b>Example</b> 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 <b>Categories</b> 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 <b>User-Defined</b> in Step 5? ▶ If YES, type the complete file name in the text box that is below the <b>User-Defined</b> 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 <b>Each document</b> or <b>All documents</b> option to select when the system runs the script.  <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
10	Click <b>Next</b> to continue to the Error Handling dialog box.  <b>Reference</b> See <a href="#">How to Set Up Error Handling Instructions</a> 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 **lftran** 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.

---

(Continued on next page)

## Routing direction

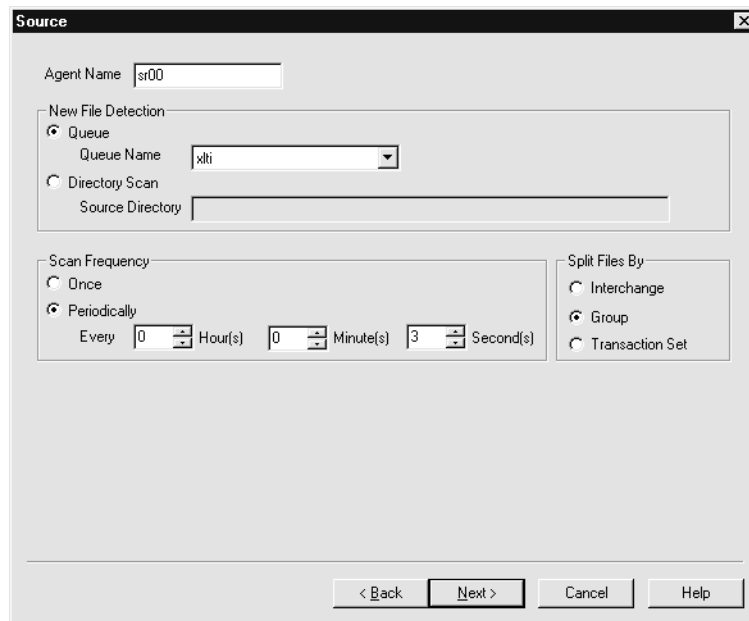
This table describes the routing direction in an inbound NCPDP flow.

Stage	Description
1	A source agent: <ul style="list-style-type: none"><li>Receives files in NCPDP format</li><li>Splits NCPDP files by trading partner</li><li>Processes data and routes data to the processing agent.</li></ul>
2	The processing agent starts the translator program, <b>lftran</b> , which translates the data and routes it to a delivery agent.
3	The delivery agent runs an embedded script named <code>&lt;dmname&gt;_gen_xltr.scr</code> . 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.



(Continued on next page)

### Source fields and functions

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

Field	Function
Agent Name	<p>Defines the name of the source data manager. The maximum size is 4 characters.</p> <p><b>Note</b> The system supplies a default name, which is based on file type selected on the <b>Flow Identification</b> dialog box. You can override the default name.</p>
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	<p>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.</p> <p><b>Example</b> ./sr03</p> <p><b>Note</b> 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.</p>
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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Function</b>
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 <b>Agent Name</b> box.  <b>Note</b> Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.  
2	Select either <b>Queue</b> or <b>Directory Scan</b> as the type of source that you want the source data manager to examine for files to process.  (Continued on next page)

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Did you select <b>Queue</b> in Step 2?</p> <ul style="list-style-type: none"> <li>▶ If YES, select the name of the queue from the drop-down list and continue with Step 4.</li> <li>▶ If NO, (the source is a directory), type the relative path name to the directory in the text box and continue with Step 4.</li> </ul> <p><b>CAUTION</b> 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.</p>
4	<p>Click <b>Once</b> or <b>Periodically</b> to select the scan frequency.</p> <p><b>Note</b> The scan frequency you select applies to every data manager in the flow.</p>
5	<p>Did you select <b>Periodically</b> in Step 3?</p> <ul style="list-style-type: none"> <li>▶ 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.</li> <li>▶ If NO, continue with Step 5.</li> </ul>
6	<p>Click <b>Interchange</b>, <b>Group</b>, or <b>Transaction Set</b> to select how the data manager groups routed data.</p>
7	<p>Click <b>Next</b> to continue to the <b>Processing</b> dialog box.</p> <p><b>Reference</b> See <a href="#">How to Set Up the Processing Agent (Inbound NCPDP Flow)</a>.</p>

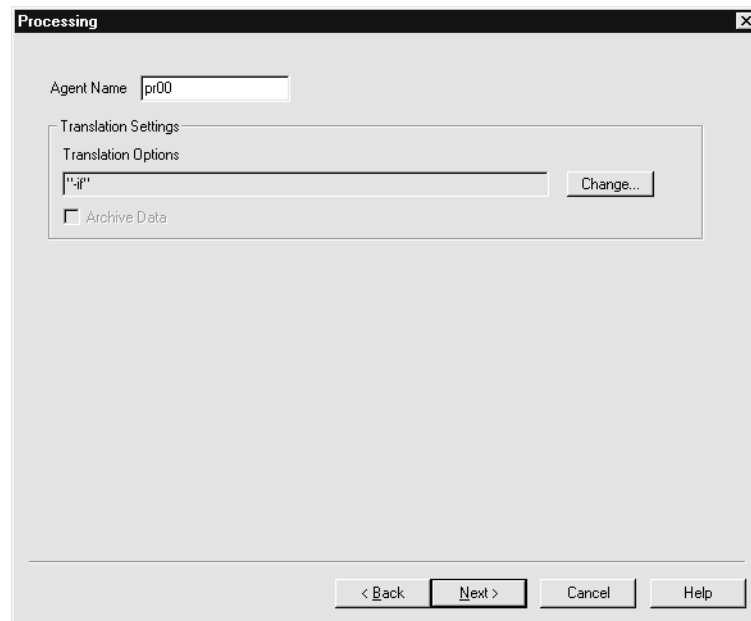
## Processing Dialog Box (Inbound NCPDP Flow)

### 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.



(Continued on next page)



**Processing  
Agent fields and  
functions**

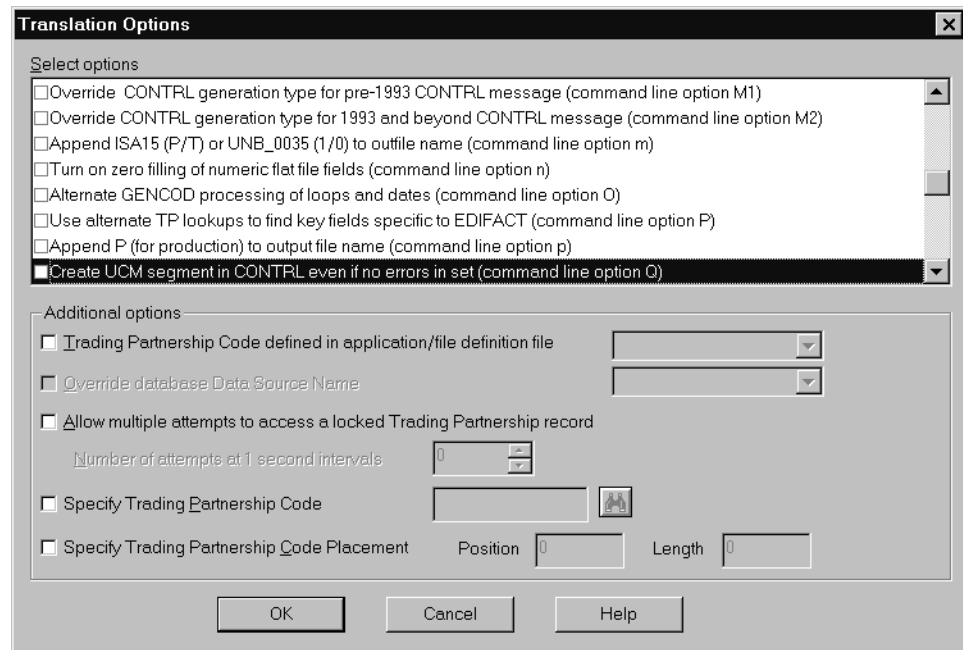
This table describes the fields of the **Processing** dialog box and their functions.

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

## Translation Options Dialog Box

**Introduction**    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.



(Continued on next page)

### Translation Option fields and functions

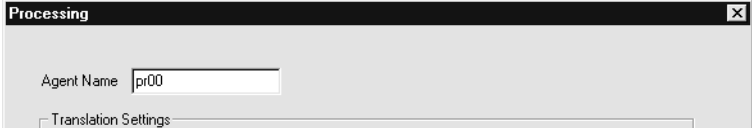
This table describes the fields of the **Translation Options** dialog box and their functions.

Field	Function
Select options	<p>Enables you to select the translation options you want to apply to this flow.</p> <p><b>Reference</b> For a list of translation options, see the <a href="#">Iftran Syntax</a> topic in the <a href="#">Command Reference</a> chapter of the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i>.</p>
Trading Partnership code defined in application/ file definition file	Enables you to select the application or file definition file. Used only for outbound translations.
Override database Data Source Name	<p>For Visual Mapper only, enables you to replace the ODBC DSN used to create the application file with the one you want to use for the current translation.</p> <p><b>Note</b> Your Gentran:Server system must have the optional ODBC translation capabilities.</p>
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	<p>Type the name of the processing data manager in the <b>Agent Name</b> box.</p> <p><b>Note</b> Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.</p> 
2	<p>Do you want to change the translation options?</p> <ul style="list-style-type: none"> <li>▶ If YES, click the <b>Change</b> button and complete the <b>Translation Options</b> dialog box.</li> <li>▶ If NO, continue with Step 3.</li> </ul>
3	<p>Do you want the translation script to run <b>ediarc</b>?</p> <ul style="list-style-type: none"> <li>▶ If YES, select <b>Archive Data</b> and then click <b>Next</b> to continue to the <b>Delivery</b> dialog box.</li> <li>▶ If NO, click <b>Next</b> to continue to the <b>Delivery</b> dialog box.</li> </ul> <p><b>Note</b> The <b>ediarc</b> program archives translation data.</p> <p><b>Reference</b> See <a href="#">How to Set Up the Delivery Agent (Inbound NCPDP Flow)</a>.</p>

## 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.

The screenshot shows the 'Delivery' dialog box with the following fields and options:

- Agent Name:
- Results Directory:
  - Queue Output
  - Set Type
  - TP Code
  - Categories
  - User-Defined
- Results File:
  - Set Type
  - User-Defined
  - TP Code
  - Categories
- Post Processing:
  - Script Name:
  - Run Script After:
    - Each Document
    - All Documents

Buttons at the bottom: < Back, Next >, Cancel, Help

(Continued on next page)

## Delivery fields and functions

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

Field	Function
Agent Name	<p>Defines the name of the delivery data manager.</p> <p><b>Note</b> The system supplies a default name, which is based on file type you selected on the <b>Flow Identification</b> dialog box. You can override the default name.</p>
Queue Output	<p>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.</p>
Set Type	<p>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.</p>
TP Code	<p>Selects Trading Partnership Code as the Results Directory (destination directory) in the configuration records.</p> <p>The Process Control Manager substitutes the actual Trading Partnership Code in the configuration records.</p>
Categories	<p>Enables you to specify a Trading Partnership category as the Results Directory (destination directory) in the configuration records.</p> <p>Select the category from the drop-down list box that is next to the Categories option.</p> <p>The Process Control Manager substitutes the actual category value in the configuration records.</p>
User Defined	<p>Enables you to specify the Results Directory (destination directory) for the configuration records.</p> <p>Enter the path in the text box that is next to the User Defined option.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Field</b>	<b>Function</b>
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 Gentran:Server script after each document has been processed.
All Documents	Executes the post-processing 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) Gentrans: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.

---

(Continued on next page)



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

Step	Action
1	<p>Type the name of the delivery data manager in the <b>Agent Name</b> box.</p> <p><b>Note</b>                      Gentran:Server supplies a default name. You may override the name. The maximum size is 4 characters.</p> 
2	<p>Choose the Results Directory by clicking <b>Set Type</b>, <b>TP Code</b>, <b>Categories</b>, or <b>User-Defined</b> to select the symbolic destination directory, or typing the path to the directory for the output.</p> <p><b>Comment</b>                      The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records. PCM creates directories if they do not exist.</p> <p><b>Example</b>                      If you select <b>TP Code</b>, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.</p>
3	<p>Did you select <b>Categories</b> in Step 2?</p> <ul style="list-style-type: none"> <li>▶ If YES, select a category from the drop-down list.</li> <li>▶ If NO, continue with Step 4.</li> </ul>
4	<p>Did you select <b>User-Defined</b> in Step 2?</p> <ul style="list-style-type: none"> <li>▶ If YES, type the path to the directory in the text box that is below the User-Defined option.</li> <li>▶ If NO, continue with Step 5.</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
5	<p>Choose the Results File name by clicking <b>Set Type, TP Code, Categories, or User-Defined</b> to select the symbolic destination file name.</p> <p><b>Comment</b> The Process Control Manager substitutes the actual value for the symbolic value when it creates the configuration records.</p> <p><b>Example</b> If you select <b>TP Code</b>, the Process Control Manager uses the actual Trading Partnership Code as the destination directory in the configuration records.</p>
6	<p>Did you select <b>Categories</b> in Step 5?</p> <ul style="list-style-type: none"> <li>▶ If YES, select a category from the drop-down list and continue with Step 7.</li> <li>▶ If NO, continue with Step 7.</li> </ul>
7	<p>Did you select <b>User-Defined</b> in Step 5?</p> <ul style="list-style-type: none"> <li>▶ If YES, type the complete file name in the text box that is below the <b>User-Defined</b> option and continue with Step 8.</li> <li>▶ If NO, continue with Step 8.</li> </ul>
8	<p>Do you want to execute a script after the translation process?</p> <ul style="list-style-type: none"> <li>▶ If YES, select the name of the script from the <b>Script Name</b> drop-down list and continue with Step 9.</li> <li>▶ If NO, continue with Step 9.</li> </ul>
9	<p>Click the <b>Each document</b> or <b>All documents</b> option to select when the system runs the script.</p>
10	<p>Click <b>Next</b> to continue to the <b>Error Handling</b> dialog box.</p> <p><b>Reference</b> See the <a href="#">Completing a Flow</a> section in this chapter for instructions on completing the <b>Error Handling</b> dialog box.</p>

# 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.

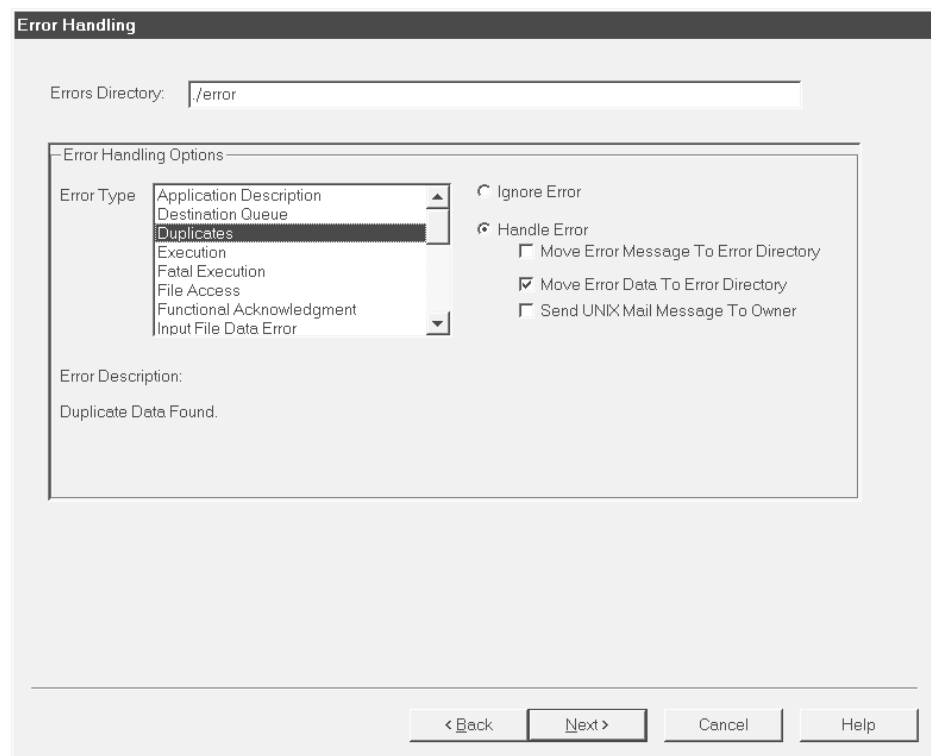
Task	Description
1	Select the error handling options. <b>Reference</b> See <a href="#">How to Set Up Error Handling Instructions</a> .
2	Select the Trading Partnership codes to use in the process flow. <b>Reference</b> See <a href="#">How to Add Trading Partnership Records to the Flow</a> .

---

## 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.



(Continued on next page)

## 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.  <b>Note</b> The default is <code>./error</code> .
Error Type	Select a type of error so that you can specify how you want 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.  <b>Reference</b> For instructions on how to add, edit, and delete UNIX mail_proc scripts, see the <a href="#">Working with UNIX Mail Scripts</a> section in the <a href="#">Working with Scripts</a> 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.**

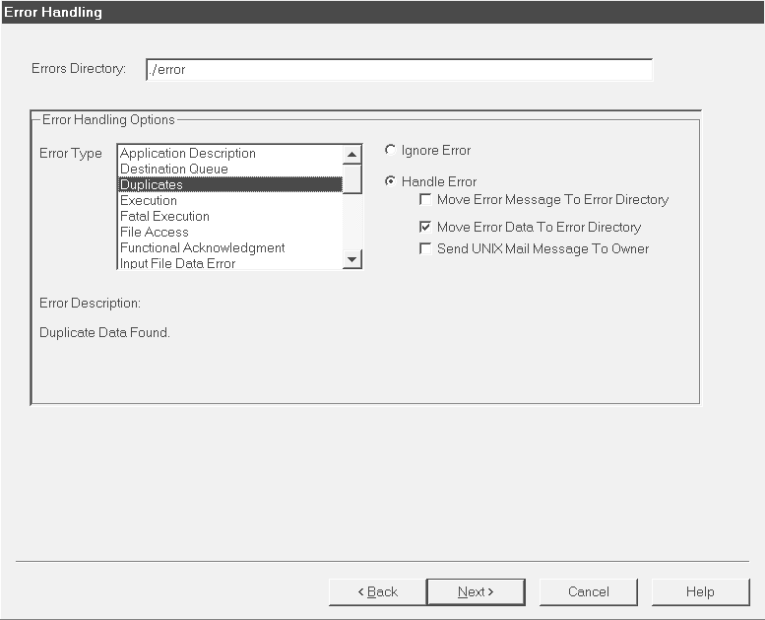
### Reference

See the [How to Use a UNIX Mail Script to Send Messages](#) topic in the [Maintaining Initialization Files](#) chapter of this guide for information about using UNIX mail scripts.

---

(Continued on next page)

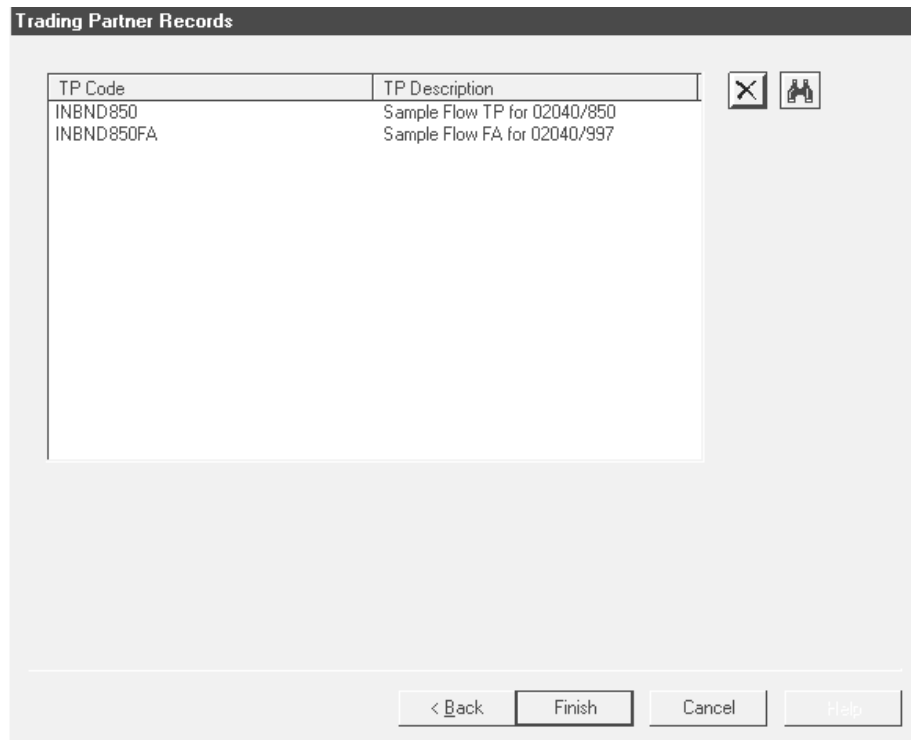
**Procedure** Use this procedure to set up error handling instructions for the process flow.

Step	Action
1	<p>Type the name of the directory to which you want errors routed.</p> 
2	<p>Select an error type from the <b>Error Type</b> list.</p>
3	<p>Do you want Gentran:Server to handle this type of error?</p> <ul style="list-style-type: none"> <li>▶ If YES, click <b>Handle Error</b> and then click on the way you want 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.</li> <li>▶ If NO, click <b>Ignore Error</b>.</li> </ul> <p><b>Note</b> If you select <b>Handle Error</b>, but do not select an instruction, Gentran:Server ignores the error.</p>
4	<p>Repeat Steps 2 and 3 until you have selected instructions for each error type.</p>
5	<p>Click <b>Next</b> to continue to the Trading Partner Records dialog box.</p> <p><b>Reference</b> See <a href="#">How to Add Trading Partnership Records to the Flow</a>.</p>

## 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.



(Continued on next page)



**Trading Partner  
Records dialog  
box fields and  
functions**

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

<b>Field</b>	<b>Function</b>
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.

**References**

For more information about configuration records, see the [Working with Configuration Records](#) chapter in this guide.


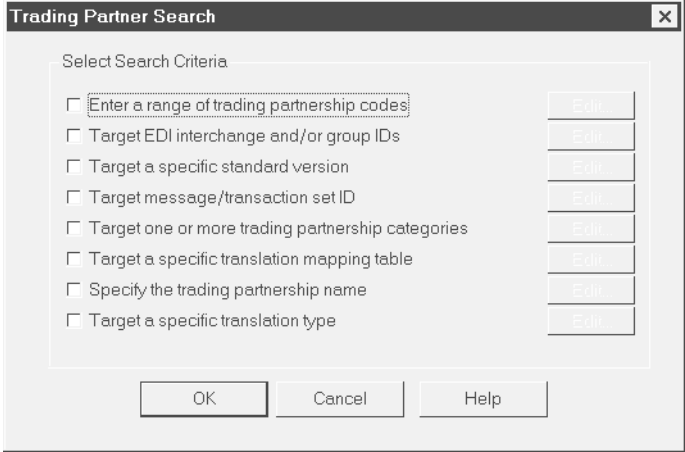
For information about Trading Partnership records, see the *Gentran:Server for UNIX and Workstation Application Integration User's Guide*.

---

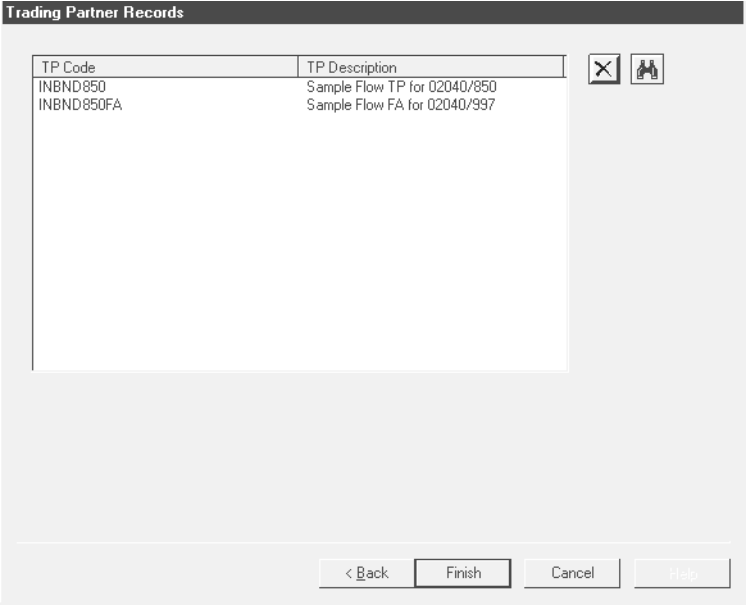
(Continued on next page)

**Adding Trading Partnerships**

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

Step	Action
1	<p>Click on the search icon.</p>  <p><b>System Response</b> Gentran:Server displays the Trading Partner Search dialog box.</p> 
2	<p>Search for the Trading Partnership code that you want to link to the flow.</p> <p><b>Reference</b> See the <a href="#">Working with Trading Partnerships</a> chapter in the <i>Gentran:Server for UNIX and Workstation Application Integration User's Guide</i> for instructions on using the Trading Partner Search dialog box.</p> <p><b>System Response</b> Gentran:Server displays the Trading Partner Search Results dialog box. This dialog box lists the Trading Partnership records that match the criteria you entered.</p> <p style="text-align: right;">(Continued on next page)</p>



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

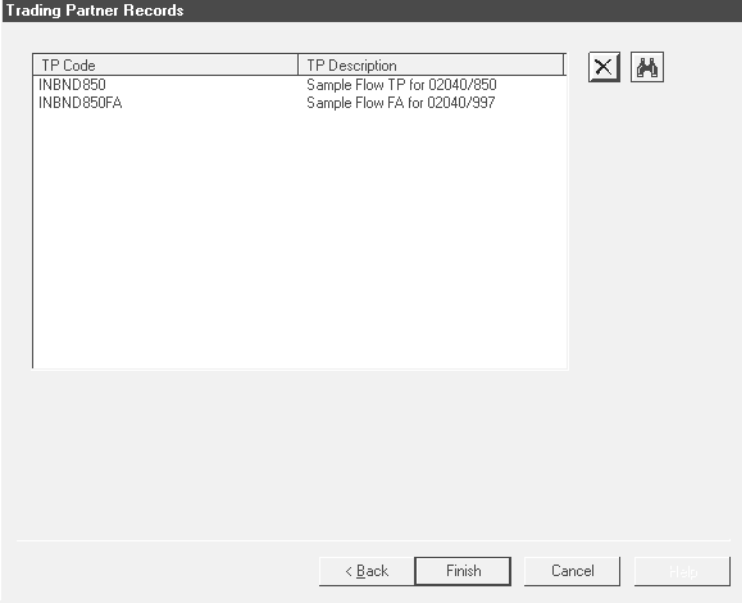
# How to Delete Trading Partnerships from the Trading Partner Records Dialog Box


**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	<p>Click the Trading Partnership code you want to delete on the Trading Partner Records dialog box.</p>  <p>(Continued on next page)</p>

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

---

# Using Flow Summaries

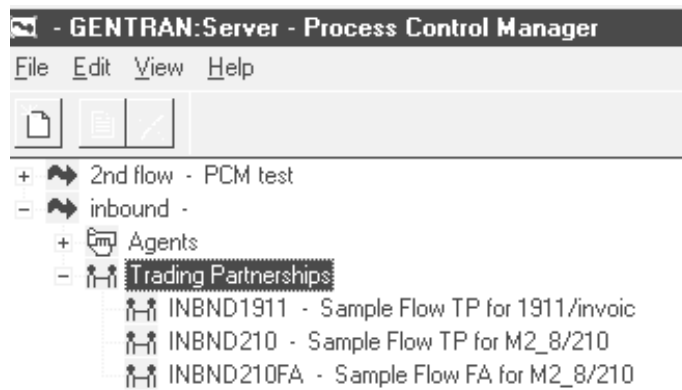
## The Flow Summary

---

**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.






---

(Continued on next page)

---

**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.

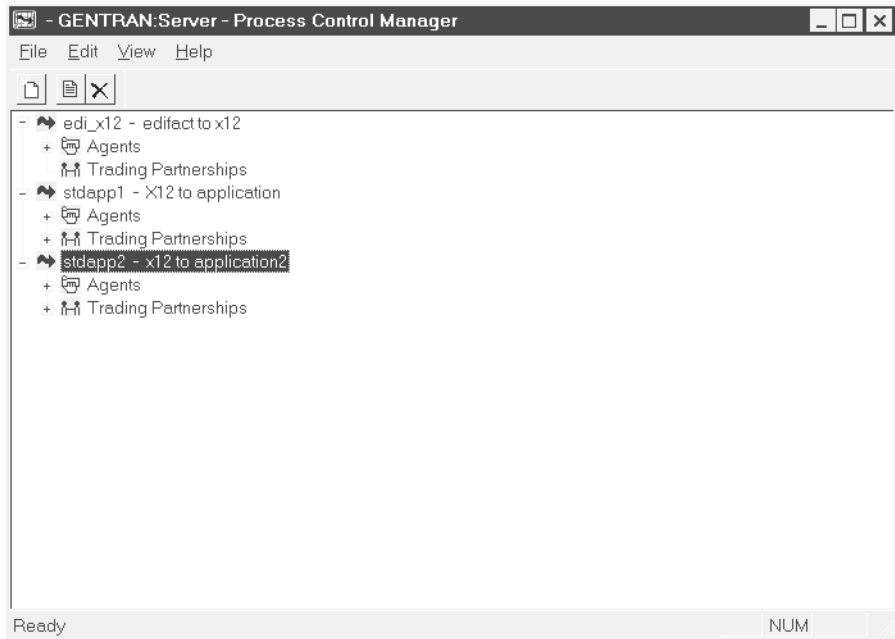


---

(Continued on next page)

**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.



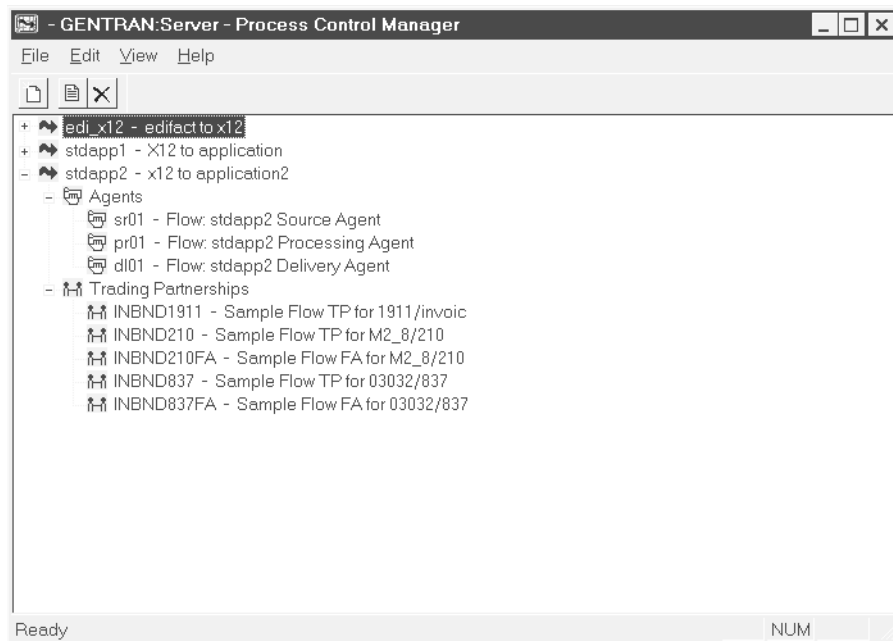
(Continued on next page)

**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 <b>Flow</b> icon. 
The individual data managers in the flow	Double-click on the <b>Agents</b> icon. 
The individual Trading Partnerships in the flow	Double-click on the <b>Trading Partnerships</b> icon. 
Only the flow names	Click <b>Refresh</b> on the View menu to collapse the view.

**Note**

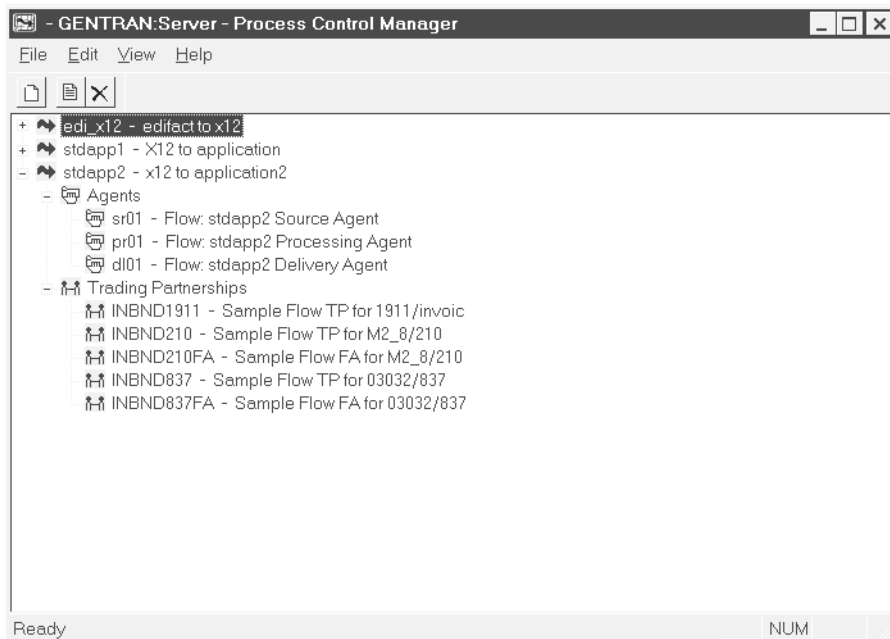
You can also use the plus (+) and minus (-) symbols to expand and collapse the view.

---

# 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.

(Continued on next page)

<b>(Contd) This report...</b>	<b>Summarizes...</b>
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 <b>PCM</b> button on the Gentran:Server main 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.	
	<b>IF you want to print...</b>	<b>THEN...</b>
	Basic flow Information	Click the flow. Do not expand the view.
	Expanded flow information	<ul style="list-style-type: none"> <li>▶ Double-click on the Flow icon to expand the view.</li> <li>▶ Click the Flow icon.</li> </ul>
	All agents Information	<ul style="list-style-type: none"> <li>▶ Double-click on the Flow icon to expand the view.</li> <li>▶ Click the Agents icon.</li> </ul>
	Single agent Information	<ul style="list-style-type: none"> <li>▶ Double-click on the Flow icon to expand the view.</li> <li>▶ Double-click on the Agents icon to expand the view.</li> <li>▶ Click the icon for the individual agent.</li> </ul>
	All Trading Partnerships Information	<ul style="list-style-type: none"> <li>▶ Double-click on the Flow icon to expand the view.</li> <li>▶ Click the Trading Partnerships icon.</li> </ul>
	Single Trading Partnership Information	<ul style="list-style-type: none"> <li>▶ Double-click on the Flow icon to expand the view.</li> <li>▶ Double-click on the Trading Partnerships icon to expand the view.</li> <li>▶ Click the icon for the individual Trading Partnership.</li> </ul>

(Continued on next page)



<b>(Contd) Step</b>	<b>Action</b>
3	Click <b>Print</b> on the File menu.  <b>System Response</b> Gentran:Server displays the Print dialog box.
4	Click OK.

---



# Maintaining Process Flows

## Overview

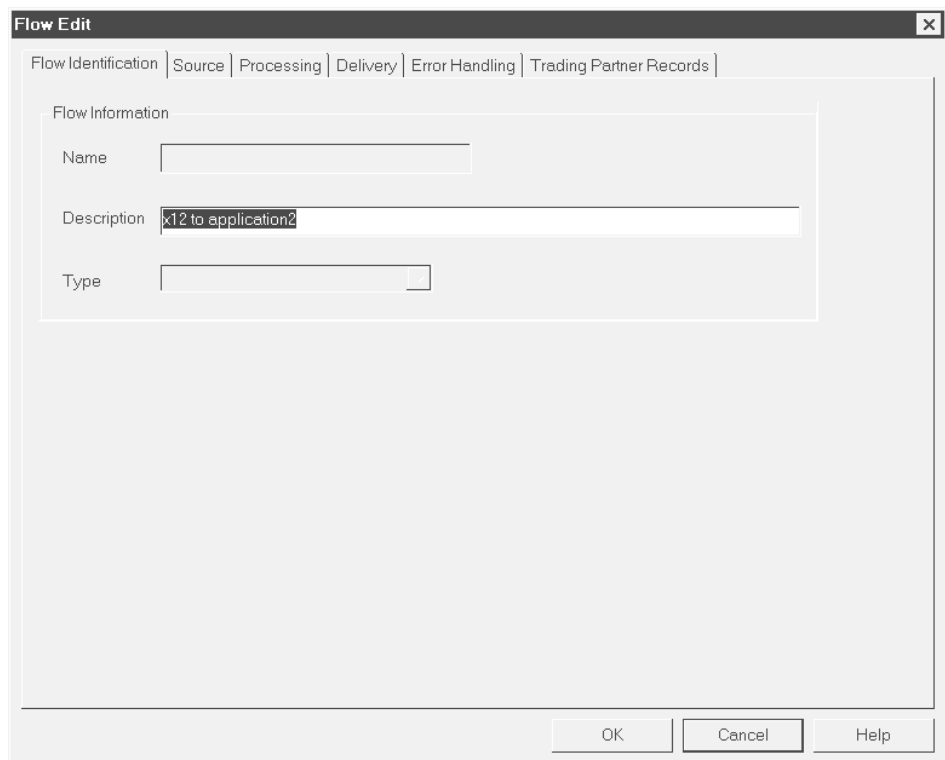
**Introduction** You can change most aspects of a process flow. To edit a flow, you change information on the flow's property sheet tabs.

### CAUTION

**You cannot use EC Workbench to modify flow components you created with the PCM Wizard.**

### 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.



The screenshot shows a dialog box titled "Flow Edit" with a close button (X) in the top right corner. The dialog has several tabs: "Flow Identification", "Source", "Processing", "Delivery", "Error Handling", and "Trading Partner Records". The "Flow Identification" tab is selected and active. It contains a "Flow Information" section with three fields: "Name" (empty text box), "Description" (text box containing "x12 to application?"), and "Type" (dropdown menu with a small arrow icon). At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

# 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.

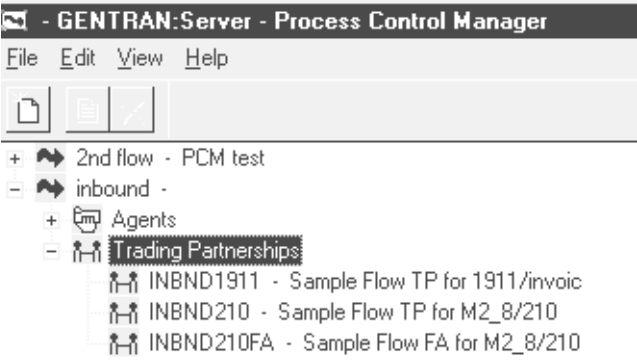
**References**

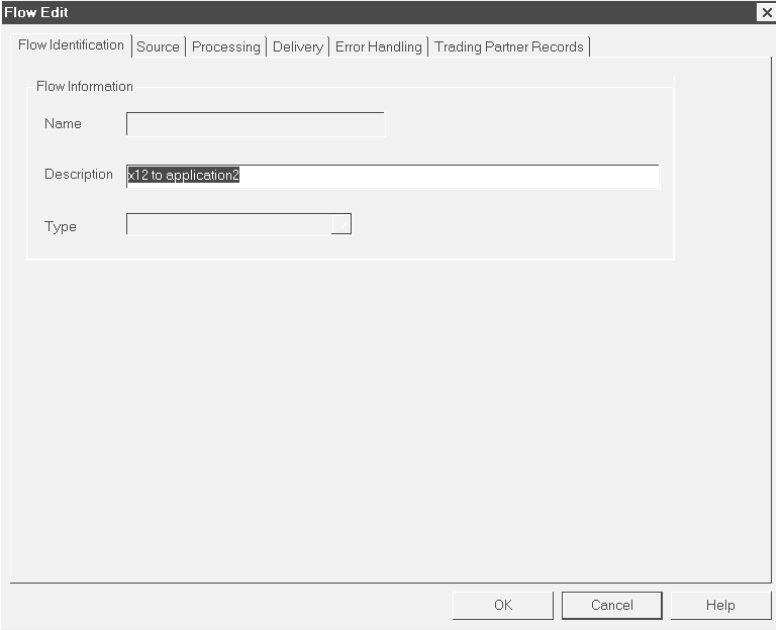
If you want to delete a Trading Partnership from a flow, see [How to Delete Trading Partnerships From the Flow](#).

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	<p>Start the Process Control Manager.</p> <p><b>System Response</b> Gentran:Server displays the Process Control Manager window.</p>  <p>The screenshot shows a window titled '- GENTRAN:Server - Process Control Manager'. It has a menu bar with 'File', 'Edit', 'View', and 'Help'. Below the menu bar are three icons: a document, a document with a pencil, and a document with a trash can. The main area is a tree view with the following structure:</p> <ul style="list-style-type: none"> <li>+ [Flow Icon] 2nd flow - PCM test             <ul style="list-style-type: none"> <li>- [Flow Icon] inbound -                 <ul style="list-style-type: none"> <li>+ [Agent Icon] Agents                     <ul style="list-style-type: none"> <li>- [TP Icon] <b>Trading Partnerships</b> <ul style="list-style-type: none"> <li>[TP Icon] INBND1911 - Sample Flow TP for 1911/invoic</li> <li>[TP Icon] INBND210 - Sample Flow TP for M2_8/210</li> <li>[TP Icon] INBND210FA - Sample Flow FA for M2_8/210</li> </ul> </li> </ul> </li> </ul> </li> </ul> </li> </ul>
2	<p>Click the flow that you want to edit.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

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

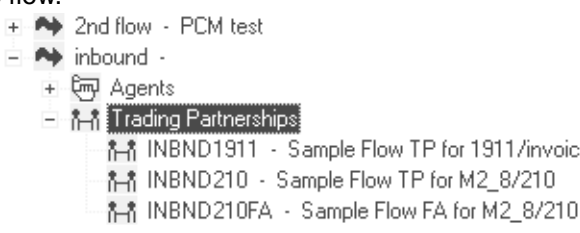
# How to Delete Trading Partnerships From the Flow

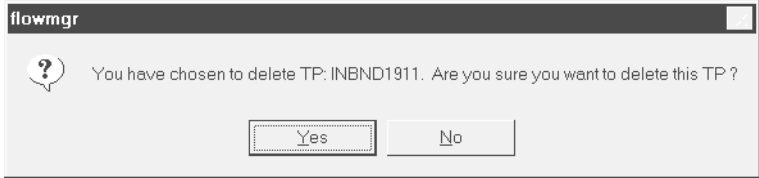
**Introduction**    This topic explains how to delete a Trading Partnership from a flow.

**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 <b>Trading Partnerships</b> to display the Trading Partnerships in the flow.  <p>The screenshot shows a tree view with the following structure:</p> <ul style="list-style-type: none"> <li>+ [icon] 2nd flow - PCM test                             <ul style="list-style-type: none"> <li>- [icon] inbound -                                     <ul style="list-style-type: none"> <li>+ [icon] Agents   <ul style="list-style-type: none"> <li>- [icon] <b>Trading Partnerships</b> <ul style="list-style-type: none"> <li>[icon] INBND1911 - Sample Flow TP for 1911/invoic</li> <li>[icon] INBND210 - Sample Flow TP for M2_8/210</li> <li>[icon] INBND210FA - Sample Flow FA for M2_8/210</li> </ul> </li> </ul> </li> </ul> </li> </ul> </li> </ul>
4	Click the Trading Partnership that you want to delete. <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
5	<p>Click <b>Delete</b> on the Edit menu.</p> <p><b>System Response</b> The Process Control Manager displays a confirmation prompt.</p> 
6	Click <b>Yes</b> to confirm the deletion.

## How to Delete a Flow

### Introduction

You can delete a flow if:

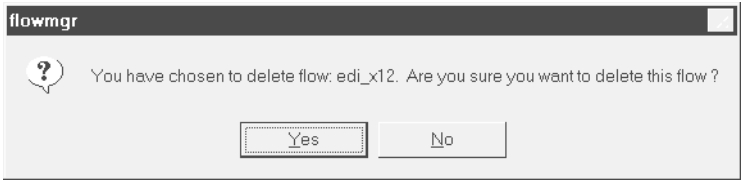
- ▶ Your application has changed
- ▶ You need to replace a flow.

### Consequences of deleting a flow

When you delete a flow, 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	<p>Click <b>Delete</b> on the Edit menu.</p> <p><b>System Response</b></p> <p>The Process Control Manager displays a confirmation prompt.</p> 
4	Click <b>OK</b> .

---

# Working with Data Managers

---

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	2
	▶ Data Managers .....	5
	▶ The Foreground Manager .....	9
	<b>Configuring Data Managers</b>	
	▶ The Flow of Work .....	10
	▶ Data Manager Control Screen .....	12
	▶ Data Manager Configuration Screen .....	17
	<b>Procedures</b>	
	▶ How to Copy a Data Manager .....	19
	▶ How to Edit a Data Manager's Process Fields .....	22
	▶ How to Delete a Data Manager .....	25
	<b>Miscellaneous</b>	
	▶ Using a Host Command Card Data Manager .....	28

---

# Overview

## Introduction

**In this chapter** This chapter explains how to create and maintain **data managers**, which are also known as **intelligent agents**.

**Key terms** This table lists the key terms used in this chapter.

Term	Description
base initialization file	The default initialization file that contains all the processing parameters for all data manager personality types except the archive data manager.
child process	A process that is started by and is part of another process. The other process is called the parent process.
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: <ul style="list-style-type: none"> <li>▶ Specifies the Trading Partnership code or file name that the data manager is to use to identify data</li> <li>▶ Tells the data manager what to do with the data it has identified.</li> </ul>
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: <ul style="list-style-type: none"> <li>▶ Routing data</li> <li>▶ Invoking scripts</li> <li>▶ Archiving data</li> <li>▶ Handling data errors.</li> </ul>
data manager log	A record of data manager activity produced when a data manager handles a data set.

(Continued on next page)



<b>(Contd) Term</b>	<b>Description</b>
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 for other data managers to run.
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.
Life Cycle	The 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. You can then use the records for auditing purposes.
Life Cycle event record	A document-tracking record produced when a data manager processes a file. The record contains the date, time, name, and source and destination of the data as it is passed through the data manager. It also contains the document reference number.
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.
pattern	One or a series of generic configuration records that describe the flow of data in a process.
personality	The data manager type that determines how the data manager processes data.
process flow	An exchange of data files between an application data manager and a translation data manager.
record file layout	<p>The user-defined file layout of an application file.</p> <p><b>Reference</b> See the <a href="#">File Record Layouts</a> chapter in the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i>.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Term</b>	<b>Description</b>
Trading Partner record	The collection of records maintained in trading partner files: Trading Partnership record, Interchange Organization record, Group Organization record, and Contact 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.
translation script	A Gentran:Server for UNIX script used to call the translator and other programs, such as <b>ediarc</b> and <b>envelope</b> .

---

# Data Managers

**Introduction** **Data managers** are the agents that cause data to flow from one point to another. They are the source and destination processing agents in a flow.

**Description and purpose** A data manager is an **intelligent agent**, an event-driven computer program that can operate without interaction from a person at a computer terminal.

The purpose of a data manager is to process data files. All data managers:

- ▶ Periodically scan a specified directory or queue to see if a data file has arrived. The arrival of a data file is the event that drives the data manager.
- ▶ Process the data based on the name of the file, the file's Trading Partnership code, or the contents of the file.

**Data manager personalities** Data managers have different personality types. The personality distinguishes the:

- ▶ Type of data to which the data manager is designed to respond
- ▶ Possible actions the data manager can be configured to take.

## Basic personality types

This table describes the basic data manager personality types.

Personality type/ Name	Description
<i>inbd</i> Inbound data manager	Processes EDI data. Can distinguish between EDI data and non-EDI data and sort them into distinct files for routing or further processing.
<i>dnld</i> Download data manager	Processes files in translator Record File Layout format. Checks files to make sure that they meet the minimum requirements for translation.  <b>Reference</b> See the <a href="#">File Record Layouts</a> chapter in the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i> .

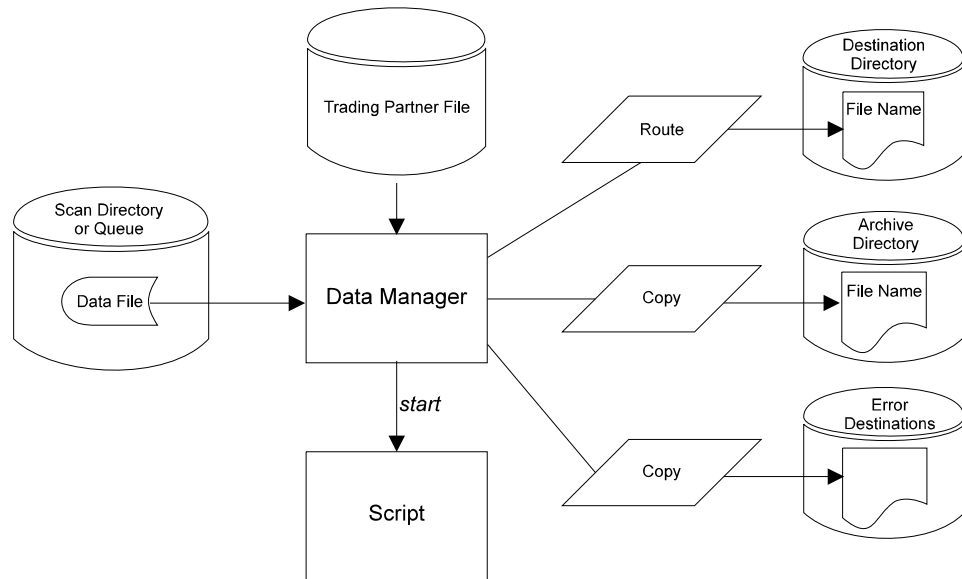
(Continued on next page)

<b>(Contd) Personality type/ Name</b>	<b>Description</b>
<i>file</i> File data manager	Reacts to the file name only, and not the file contents. The name of the file determines the action that the data manager takes.
<i>hcmd</i> Host command card data manager	Examines the data for commands marked by separators that distinguish the command from the data. Processes the data that follows the commands according to the instructions in the commands. Used primarily to route proprietary data.
<i>xltr</i> Translation data manager	Scans a directory or queue and calls a translation script to process the data it finds.  Uses the file name to route the data. Checks the INPUT_FN_APP_TP parameter in its initialization file to determine what the input file name represents.
<i>lnmn</i> Line manager data manager	Used only in the optional Advanced Data Distribution System to control asynchronous and bisynchronous communications lines. Starts and stops the line manager processes.  <b>Reference</b> See the <i>Gentran:Server for UNIX Advanced Data Distribution Guide</i> for information.
<i>appt/appt</i> Application data manager	Processes files that are in an application format your organization defined. Examines the files to make sure that they meet minimum requirements for translation.  <b>Reference</b> See the <a href="#">File Record Layouts</a> chapter in the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i> .
<i>ncpdp</i> NCPDP data manager	Processes files that are in NCPDP format. Extracts values of the six key fields to determine the Trading Partnership code. Uses the Trading Partnership record to route the data.
<i>xmli/xmlo</i> XML data manager (inbound or outbound)	If you have the Gentran:Server XML translation option, you have available the XML data manager personality, which processes files that are in XML format. Checks syntax and splits XML data.

(Continued on next page)

## Processing illustration

This illustration shows the possible processing actions of an inbound, download, application, or translation data manager. Note that this diagram depicts only the typical types of processing actions. It does not depict the order in which these events happen.



## Types of processing actions

The processing actions a data manager takes is based on the personality of the data manager and how you configure it to react to the type of data.

This table describes the possible processing actions of a data manager.

Action	Description
Route the data	Based on the type of data in the file, the data manager identifies the destinations for the data and routes the data to those locations. The destination is a directory or queue. If your organization purchased the optional Advanced Data Distribution System, a destination can be a mailbox ID or a distribution list.
Invoke a script	Runs a post-processing script based on the type of data it encounters.
Archive the data	Directs a copy of the data to an archive directory.

(Continued on next page)

<b>(Contd) Action</b>	<b>Description</b>
React to errors	<p>Detects and reacts to different data error conditions. In addition to redirecting the data in error, you can configure data managers to send the administrator e-mail notification that an error took place.</p> <p><b>Example</b> The data manager can copy duplicate data to a duplicates directory and copy files with syntax or structural errors to an error directory.</p>
Generate a Life Cycle record	Produces a Life Cycle event record, which includes the date, time, name, and location of the data as it is passed through the data manager.

### **Halts in processing**

---

During normal processing, a data manager periodically keeps track of its place in the file it is processing. If processing halts unexpectedly, such as in a power outage, the data manager “remembers” where it stopped. When the machine restarts, this recall ability enables the data manager to start processing at the beginning of the data set where it left off.

---

# The Foreground Manager

---

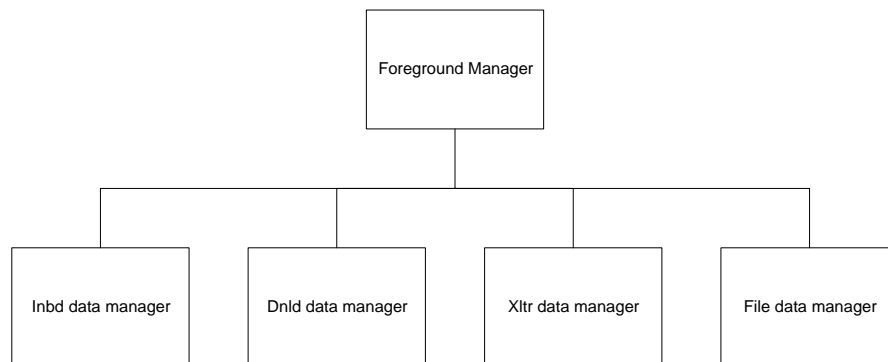
**Definition** The Foreground Manager (fmgr) is the parent process of the other data managers.

The Foreground Manager:

- Runs continuously under normal conditions, but can be stopped during system shutdown
- Does not need to be configured
- Maintains an activity log named *fmgr.l*
- Starts at machine start if **fmgr** is entered into the */etc/inittab* or */etc/rc* file.

---

**Illustration** This illustration shows the Foreground Manager as the parent process of other data managers.



---

**Relationship to other data managers**

All other data managers are:

- Configured through the Foreground Manager
- Started and stopped by the Foreground Manager (either automatically or by manually entered start and stop commands).

---

**Sends termination mail message**

If a data manager terminates without a Foreground Manager request, the Foreground Manager is configured to send a UNIX mail message to let the operator know about the termination.

---

# Configuring Data Managers

## The Flow of Work

---

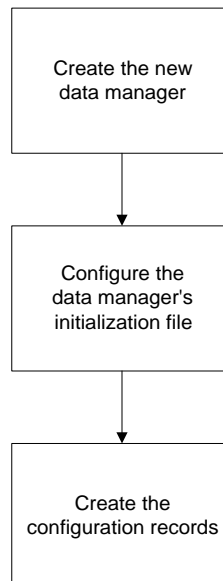
### Introduction

Data manager configuration is the process of creating a data manager (if it does not exist) and defining how it processes data. The process includes creating configuration records that link the data manager to a Trading Partnership code or file name.

---

### Process flow diagram

This illustration shows the process flow for configuring a data manager.



---

(Continued on next page)



## The configuration process

This table describes the tasks in the data manager configuration process.

Task	Description
1	<p>Create the data manager (if it does not exist).</p> <p>On the Data Manager Configuration screen, copy an existing data manager or template and then modify the field values. Remember that data managers are child processes of the Foreground Manager, <b>fmgr</b>.</p> <p><b>Reference</b> See the <a href="#">How to Copy a Data Manager</a> topic in this chapter.</p>
2	<p>Configure the data manager's initialization file.</p> <ol style="list-style-type: none"> <li>1. Open the data manager's initialization file and set the data manager's personality, processing parameters, and error-handling instructions.</li> <li>2. We recommend that you access the initialization file through the Data Manager Configuration screen. When you edit an initialization file accessed through the Data Manager Configuration screen, Gentran:Server: <ul style="list-style-type: none"> <li>— Creates the work directory (WORK_DIR parameter) and the archive directory (ARCHIVE_DIRECTORY parameter) if they do not exist</li> <li>— Alerts you if the data manager is running</li> <li>— Validates the changes.</li> </ul> </li> </ol> <p><b>Reference</b> See the <a href="#">Maintaining Initialization Files</a> chapter in this guide.</p>
3	<p>Create the configuration records.</p> <p>In these records, identify the data you want the data manager to handle and enter parameters that tell the data manager what to do with the data.</p> <p><b>Reference</b> See the <a href="#">Working with Configuration Records</a> chapter in this guide.</p>

## Data Manager Control Screen

### Introduction

You start data manager configuration activities from the Data Manager Control screen. The Data Manager Control screen lists and describes the data managers in your system. You use this screen to:

- Access the Query screen for an archive handler
- Start and stop data managers
- Open the default editor so that you can edit the data manager's UNIX mail script
- Access the Data Manager Configuration screen, which enables you to copy and delete data managers or access a data manager's initialization file for editing
- Update a data manager's running status
- Display a data manager's log file.

### Data Manager Control screen

This illustration shows an example of the Data Manager Control screen.

```

Data Manager Control
Name A   Status  T Description
-----
fmgr A   1239 F Foreground Manager (IPC Control)
alnm n   ***** l Async Line Manager
ap00 n   ***** m Flow: Test Q! flow Source Agent
ap01 n   ***** m Flow: 'nother fifteen Translate Agent
appm n   ***** m Application Data Manager
appt n   ***** x Application Translator Data Manager
arch n   ***** a -Darch -Aarch -d0
base n   ***** u Base Manager Model
cfin n   ***** i Flow: chris_flow Translate Agent
dnld n   ***** d UDF Data Manager
edii n   ***** i Inbound Data Manager
edio n   ***** i Outbound Data Manager
file n   ***** f File Data Manager
hcmd n   ***** h Host Command Card Data Manager
in00 n   ***** i Flow: flow Source Agent
in01 n   ***** i Flow: fifteen digits. Translate Agent
in02 n   ***** i Flow: test0505 Translate Agent
F2:Arch F3:Stop F4:EditMail F5:Config F6:Stat F7:Log F8:Start F9:Quit

```

(Continued on next page)

## 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 can use any combination of alphabetic and numeric characters.	
A (Autostart)	Determines whether the data manager starts automatically with the Foreground Manager. <ul style="list-style-type: none"> <li>▶ Y = Yes, starts automatically</li> <li>▶ N = No, does not start automatically</li> </ul>	
Status	Indicates the running status of the data manager. <b>Tip</b> Press F6 to update the status.	
	<b>IF the column displays...</b>	<b>THEN the data manager is...</b>
	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.

(Continued on next page)

(Contd) Field	Function	
T	Indicates the data manager type.	
	<p><b>Note</b> This field determines the type of Data Manager Configuration screen Gentran:Server displays for this data manager. The data manager's personality is set in its initialization file.</p>	
	Type Code	Description
	F	The Foreground Manager
	d	Download
	i	Inbound
	a	Archive
	f	File
	x	Translation
	l	Line manager (Advanced Data Distribution System only)
	h	Host command card
	m	Application
y	XML (if you have the XML translation option)	
Description	<p>Describes the data manager. Type up to 50 characters.</p> <p><b>CAUTION</b> <b>For an archive data manager, this field contains the archive data manager's processing parameters.</b></p> <p><b>Reference</b> For information about using this field to set the processing parameters for an archive data manager, see the <a href="#">Archiving Your Data</a> chapter in this guide.</p>	

(Continued on next page)

**Function keys of  
the Data  
Manager Control  
screen**

This table describes the function keys of the Data Manager Control screen.

Key	Function
F2	Displays the Query screen for the selected archive handler.
F3	Stops a running data manager.
F4	Opens the default editor so that you can edit the data manager's UNIX mail script.
F5	If pressed while <b>fmgr</b> is selected, displays the Data Manager Configuration screen, which enables you to copy, modify, and delete data managers.  If pressed while a data manager is selected, enables you to open, edit, and delete configuration records for the selected data manager.
F6	Updates the data manager's running status 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.

(Continued on next page)

---

**Control required to configure data managers**

There are two types of control available in the Data Manager Control screen (or the PCM wizard):

- ▶ Primary, which gives you the ability to configure, start, and stop data managers
- ▶ Secondary, which gives you the ability to view data manager records.

**CAUTION**

**You must have primary control to configure, start, or stop a data manager.**

**How control is granted**

Gentran:Server gives primary control to the first Gentran:Server user who invokes the Data Manager Control screen.

Any user who accesses the screen while the first user is using the screen has secondary control. Gentran:Server displays a message to the secondary users to let them know who has primary control.

When the first user leaves the screen, they relinquishes primary control. The next Gentran:Server user to access the screen gains primary control. A user who has secondary control must leave and return to gain primary control when it becomes available.

---

## Data Manager Configuration Screen

### Introduction

You start the data manager copy, delete, and edit functions from the Foreground Manager's Data Manager Configuration screen.

### Data Manager Configuration screen

This illustration shows an example of the Data Manager Configuration screen.

```

dmc Data Manager Configuration
Name A Status T Description
alnm n 0 l Async Line Manager
ap00 n 0 m Flow: Test Q! flow Source Agent
ap01 n 0 m Flow: 'nother fifteen Translate Agent
appm n 0 m Application Data Manager
appt n 0 x Application Translator Data Manager
arch n 0 a -Darch -Aarch -d0
base n 0 u Base Manager Model
cfin n 0 i Flow: chris_flow Translate Agent
dnld n 0 d UDF Data Manager
edii n 0 i Inbound Data Manager
edio n 0 i Outbound Data Manager
file n 0 f File Data Manager
hcmd n 0 h Host Command Card Data Manager
in00 n 0 i Flow: flow Source Agent
in01 n 0 i Flow: fifteen digits. Translate Agent
in02 n 0 i Flow: test0505 Translate Agent
in03 n 0 i Flow: happyhappyjoyjo Source Agent
F3:Del F4:Copy F5:Edit F9:Quit

```

### Function keys of the Data Manager Configuration screen

This table describes the function keys of the Data Manager Configuration screen.

Key	Function
F3	Deletes the selected data manager.
F4	Copies the selected data manager.
F5	Displays the Edit screen for the selected data manager so that you can modify the data manager's autostart, type, or description. You can also access the data manager's initialization file from the Edit screen.
F9	Exits the screen.

(Continued on next page)

**Using the Data  
Manager  
Configuration  
screen**

---

This table describes how to use the Data Manager Configuration screen.

<b>IF you want to...</b>	<b>THEN...</b>
Insert one or more characters	Place the cursor at the insertion point and type the characters.
Change a character	Select the character and type a new character over it.
Erase a character	Press the space bar.
Insert a space	Press the space bar.

---



## Procedures

### How to Copy a Data Manager

---

**Introduction**

Gentran:Server comes with a Foreground Manager and a template for each data manager personality. To add a data manager to your system, you must copy one of the templates or an existing data manager that handles data in a similar manner to the one you want to add.

**CAUTION**

**If you use the PCM wizard to create your process flows, the wizard creates the appropriate data managers for the type of flow. You cannot use EC Workbench to modify data managers you created with the PCM Wizard.**

---

**Files created when you copy a data manager**

When you copy a data manager, Gentran:Server copies these files and renames them for the new data manager:

- ▶ Configuration information
- ▶ Initialization file
- ▶ Mail\_proc UNIX mail file.

---

(Continued on next page)

**Procedure** Use this procedure to copy a data manager.

Step	Action
1	<p>Select <b>DataMgr</b> from the Gentran:Server host Main Menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen. This screen lists all the data managers that have been added to Gentran:Server.</p>
2	<p>Select <b>fmgr</b>, the Foreground Manager.</p> <pre> Data Manager Control Name A      Status  T Description ----- fmgr A      1239 F Foreground Manager (IPC Control) alrm n      ***** l Async Line Manager ap00 n      ***** m Flow: Test Q! flow Source Agent ap01 n      ***** m Flow: 'nother fifteen Translate Agent appm n      ***** m Application Data Manager appt n      ***** x Application Translator Data Manager arch n      ***** a -Darch -Aarch -d0 base n      ***** u Base Manager Model </pre>
3	<p>Press F5.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Configuration screen, which lists the data managers that are child processes of the Foreground Manager.</p>
4	<p>Select the name of the data manager you want to copy.</p> <pre> dmc Data Manager Configuration Name A      Status  T Description ----- alrm n      0 l Async Line Manager ap00 n 0      m Flow: Test Q! flow Source Agent ap01 n 0      m Flow: 'nother fifteen Translate Agent appm n      0 m Application Data Manager appt n      0 x Application Translator Data Manager </pre>
5	<p>Press F4 to copy the data manager.</p> <p><b>System Response</b> Gentran:Server displays the Copy screen.</p> <pre> Copy Name A      Status  T Description ----- alrm n      0 l Async Line Manager </pre> <p>F9:Quit F10:Save</p>
6	<p>Type the name of the new data manager in the Name field.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
7	<p>Make any necessary changes to the Autostart (A), Type (T), and Description fields.</p> <p><b>WARNING</b></p> <p>You can type only Y or N in the Autostart field. Also, be certain to use only a valid character in the Type field. Note that the Type field determines the type of Data Manager Configuration screen Gentran:Server displays for this data manager. You set the data manager's personality in its initialization file.</p> <p><b>Reference</b></p> <p>See the <a href="#">Data Manager Control Screen</a> for a list of valid characters for the Type field.</p>
8	Press F10 to save your changes.

**CAUTION**

To change the process fields for your new data manager, follow the instructions in the next topic.

---

## How to Edit a Data Manager's Process Fields

**Introduction** This topic explains how to change a data manager's description, type, and the flag that has the Foreground Manager start the data manager.

### References

- ▶ To change the data manager's personality, how it processes data, or how it handles errors, see the [Maintaining Initialization Files](#) chapter in this guide.
- ▶ To change the configuration record that identifies what the data manager processes, see the [Working with Configuration Records](#) chapter in this guide.

**The process fields** The process fields are displayed on the Data Manager Control screen and on the Data Manager Configuration screen. The process fields include:

- ▶ The name of the data manager (cannot be changed)
- ▶ Whether the data manager automatically starts with the Foreground Manager
- ▶ The data manager type (i, d, f, x, etc.)
- ▶ The description of the data manager.

### CAUTION

**Do not change the Type (T) field if configuration records have been created for this data manager. The Type controls the names of the fields in the data manager's configuration records.**

**Procedure** Use this procedure to edit a data manager's process fields.

Step	Action
1	Select <b>DataMgr</b> from the Gentran:Server host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.  <div style="text-align: right; color: red;">(Continued on next page)</div>

(Contd) Step	Action
2	<p>Select <b>fmgr</b>, the Foreground Manager.</p> <pre> Data Manager Control Name A Status T Description ----- fmgr A 1239 F Foreground Manager (IPC Control) alnm n ***** l Async Line Manager ap00 n ***** m Flow: Test Q! flow Source Agent ap01 n ***** m Flow: 'nother fifteen Translate Agent appm n ***** m Application Data Manager appt n ***** x Application Translator Data Manager arch n ***** a -Darch -Aarch -d0 </pre>
3	<p>Press F5.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.</p> <pre> dmc Data Manager Configuration Name A Status T Description ----- alnm n 0 l Async Line Manager ap00 n 0 m Flow: Test Q! flow Source Agent ap01 n 0 m Flow: 'nother fifteen Translate Agent appm n 0 m Application Data Manager appt n 0 x Application Translator Data Manager </pre>
4	<p>Select the data manager you want to edit.</p> <p><b>WARNING</b></p> <p><b>You cannot edit an active data manager. Before you continue, check the Status field to make sure the data manager is not running. If it is running, stop the data manager, wait until the data manager's status is inactive, and then edit it.</b></p> <p><b>Reference</b> See the <a href="#">Starting and Stopping Data Managers</a> section in the <a href="#">Monitoring Processes</a> chapter of this guide for instructions on stopping data managers.</p>
5	<p>Press F5 to edit the data manager.</p> <p><b>System Response</b> Gentran:Server displays the Edit screen.</p> <pre> Edit Name A Status T Description ----- alnm 0 l Async Line Manager F5:EditInit F9:Quit F10:Save </pre> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
6	Modify the fields.  <b>WARNING</b> <b>Do not change the type (T) field if configuration records have been created for this data manager.</b>
7	Press F10 to save your changes.

---

## How to Delete a Data Manager

### Introduction

If you no longer need a data manager, you can delete it. When you delete a data manager, Gentran:Server automatically deletes the data manager's log file, initialization file, associated configuration records, and UNIX mail\_proc file. Gentran:Server prompts you to keep or delete the work directory and archive directory.

### CAUTION

**When you delete a data manager with this procedure, Gentran:Server automatically deletes the files associated with the data manager. If you need these files for historical or other purposes, copy the data manager and rename it before you delete the original.**

**If other data managers use the same work directory or archive directory, do not delete them! Doing so could result in loss of data.**

### Deleting a data manager

Use this procedure to delete a data manager.

Step	Action
1	Select <b>DataMgr</b> from the Gentran:Server host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.
2	Select <b>fmgr</b> , the Foreground Manager.  <pre> Data Manager Control Name A      Status  T Description ----- fmgr A      1239 F Foreground Manager (IPC Control) alnm n      ***** l Async Line Manager ap00 n      ***** m Flow: Test Q! flow Source Agent ap01 n      ***** m Flow: 'nother fifteen Translate Agent appm n      ***** m Application Data Manager appt n      ***** x Application Translator Data Manager arch n      ***** a -Darch -Aarch -d0 base n      ***** u Base Manager Model cfin n      ***** i Flow: chris_flow Translate Agent dnld n      ***** d UDF Data Manager </pre>
3	Press F5.  <b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.  (Continued on next page)

(Contd) Step	Action
4	<p>Select the data manager that you want to delete.</p> <p><b>WARNING</b></p> <p><b>You cannot delete an active data manager. Before you continue, check the Status field. If the data manager is running, stop the data manager, wait until the data manager's status is inactive, and then delete it.</b></p> <p><b>To make a data manager stop after it has finished processing data in its scan directory, go to the Data Manager Control screen, select the data manager, and press F3.</b></p>
5	<p>Press F3 to delete the data manager.</p> <p><b>System Response</b> Gentran:Server displays a confirmation prompt.</p> <pre data-bbox="643 951 1403 1073"> Delete app4 Name A   Status   T Description app4 n           0 m New application data manager  Do you want to delete this entry? y/n █ F9:Quit</pre>
6	<p>Type <b>y</b> to confirm the deletion.</p> <p><b>System Response</b> Gentran:Server displays a confirmation prompt for the work (scan) directory.</p> <pre data-bbox="651 1325 1403 1444"> Delete app4           Confirmation Name A   Status   Scan Directory: appm app4 n           Do You Wish to Remove Directory ?  Do you want to delete this entry? y/n y</pre> <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
7	<p>Do you want to delete the work directory?</p> <ul style="list-style-type: none"><li>▶ If YES, press <b>y</b>.</li><li>▶ If NO, press <b>n</b>.</li></ul> <p><b>WARNING</b></p> <p><b>Do not delete the work directory if it is used by other data managers.</b></p> <p><b>System Response</b> Gentran:Server displays a confirmation prompt for the archive directory.</p>
8	<p>Do you want to delete the archive directory?</p> <ul style="list-style-type: none"><li>▶ If YES, press <b>y</b>.</li><li>▶ If NO, press <b>n</b>.</li></ul> <p><b>WARNING</b></p> <p><b>Do not delete the archive directory if it is used by other data managers.</b></p> <p><b>System Response</b> Gentran:Server deletes the data manager and either deletes or saves the work and archive directories according to your responses.</p>

## Miscellaneous

### Using a Host Command Card Data Manager

---

#### Introduction

**Host command card (hcmd)** data managers are used to route **proprietary** data. Proprietary data is data that is in a non-EDI-standard format.

---

#### Command cards

A host command card data manager is designed to handle data files that contain embedded information called **command cards**.

The command cards contain this information:

- ▶ Trading Partnership code
- ▶ Trading Partnership code modifier
- ▶ Document reference number.

---

#### Inserting command cards

You must insert or have your system insert the command cards into the file before routing the file to the host command card data manager. Usually, you create a script to insert the command cards.

You must insert a host command card before each set of records that you want routed together.

---

#### Host command card format

You must enclose a command card in a **delimiter string** to separate it from the data itself. The same delimiter string marks the start and the end of the command card.

This is the format of the command card and delimiter string:

```
delimiter_string tp_code modifier doc_ref_number delimiter_string (nl)
```

---

(Continued on next page)

**Format description**

This table describes the parts of the format.

Part	Description
delimiter_string	Any printable text string. The string cannot contain blanks, but may contain a control character, such as the dollar sign (\$) value.  You define the delimiter string in the CMD_STR parameter of the host command card data manager's initialization file.  <b>Example</b> \$CMD_STR\$
tp_code	The Trading Partnership code.
modifier	The Trading Partnership code modifier. The modifier is required for host command card data managers.
doc_ref_number	The document reference number. This is the identifier that uniquely identifies the document.
(nl)	New line. A new line must follow the ending delimiter string.

**Initialization file parameters**

You must set one of these personality-specific parameters in the host command card's initialization file:

- CMD\_STR, which defines the delimiter string that encloses command cards to separate them from the data in a file
- CMD\_STR\_ENV, which specifies an environment variable you want to represent the delimiter string. Omit the dollar sign (\$).

**Reference**

See the [Maintaining Initialization Files](#) chapter in this guide for information about setting parameters in a data manager's initialization file.

**Configuration records**

The configuration records you create for a host command card data manager must:

- Be based on the Trading Partnership code
- Include a Trading Partnership code modifier.

**Reference**

See the [Working with Configuration Records](#) chapter in this guide for information about creating configuration records.

(Continued on next page)

**Processing**

This table describes the processing stages of a host command card data manager.

---

<b>Stage</b>	<b>Description</b>
1	The host command card data manager picks up a file in its scan directory or queue and searches for a configuration record that matches the Trading Partnership code and modifier.
2	If a matching configuration record exists, the data manager directs the file to the destination specified in the configuration record.
3	The data manager passes the actual data that follows the command card until it detects another command card or reaches the end of the file. The data manager passes only the data, not the command card.

---

---

# Maintaining Initialization Files

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	3
	▶ Data Manager Initialization Files .....	6
	<b>Initialization File Structure</b>	
	▶ Overview .....	8
	▶ Parameters Section .....	10
	▶ Errors Section .....	14
	▶ Standard Error Types .....	16
	▶ Handlers Section .....	19
	<b>Parameters Subsections</b>	
	▶ Overview .....	22
	▶ Parameters and Personality Table .....	23
	▶ Routing Parameters .....	29
	▶ Line Manager Parameters .....	35
	▶ Translator Parameters .....	36
	▶ Miscellaneous Parameters .....	38
	▶ Archive Handling Parameters .....	52
	▶ Duplicates Checking Parameters .....	53
	<b>Maintaining Data Manager Initialization Files</b>	
	▶ Overview .....	54
	▶ How to Access and Edit an Initialization File .....	55
	▶ How to Copy or Delete an Initialization File .....	58
	<b>Configuring Initialization Files for Specific Functions</b>	
	▶ Overview .....	59
	▶ How to Set the Destination File Name .....	60
	▶ How to Set Error Notification .....	63
	▶ How to Use a UNIX Mail Script to Send Messages .....	67
	▶ How to Use an Error Class .....	71
	▶ How to Control Concurrent Processing With File Locking .....	73

- ▶ The File Locking Process ..... 75
- Configuring for Real-time Processing**
- ▶ Overview ..... 77
- ▶ The Stages of Real-Time Processing ..... 78
- ▶ How to Configure Data Managers for Real-Time Processing .... 82
- ▶ How to Configure the FTP Daemon to Notify a Data Manager ... 84

---

# Overview

## Introduction

**In this chapter** This chapter describes the parts of a data manager initialization file and explains how to create and maintain initialization files.

**Key terms** This table lists the key terms used in this chapter.

Term	Description
base initialization file	The default initialization file that contains all the processing parameters for all data manager personality types except the archive data manager. The file name is <i>base.init</i> .
beeper	The Sterling Commerce script that invokes the script specified in the <code>script_name</code> position of an error handler. Used when the error handler is configured to trigger a beeper whenever a specific type of error occurs.
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: <ul style="list-style-type: none"> <li>▶ Specifies the Trading Partnership code or file name that the data manager is to use to identify data</li> <li>▶ Tells the data manager what to do with the data it has identified.</li> </ul>
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: <ul style="list-style-type: none"> <li>▶ Routing data</li> <li>▶ Invoking scripts</li> <li>▶ Archiving data</li> <li>▶ Handling data errors.</li> </ul>

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
default value	The value that Gentran:Server uses if you do not specify a different value.
downstream data manager	A data manager that processes files that a previous (upstream) data manager has placed in its work directory or queue.
error class	An error category set for an error handler and entered into a configuration record. An error class enables Gentran:Server to handle the same type of error differently for different Trading Partnerships or file names.
error handler	A line in the HANDLERS section of a data manager's initialization file used to specify how you want a particular error handled.
FIFO	A first-in-first-out file that can be read only once before the data is removed.
Foreground Manager	The parent data manager of all other data managers. You can use Foreground Manager (fmgr) to configure the data managers. Fmgr must be running before other data managers can be run.
FTP daemon	A background program that controls file transfer protocol. File transfer protocol moves or copies files between computers.
initialization file	The configurable file that sets the data manager's personality and processing parameters.
IPC Trigger	Inter Process Control Trigger. The Gentran:Server feature that enables you to configure data managers to process files in real-time mode.
line manager	A special type of data manager used only with the optional Advanced Data Distribution system. It is used to transfer files to and from users' mailboxes.
lock	A way to configure a data manager's initialization file to lock access to files in its scan directory until external processing is complete. Used to ensure that two data managers or other processes do not process a file at the same time.  <p style="text-align: right;">(Continued on next page)</p>



<b>(Contd) Term</b>	<b>Description</b>
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. Gentran:Server creates the mail script and gives it the data manager's name.
personality	The data manager type that determines how the data manager processes data.
real-time processing	A system configuration that enables your system to move critical documents through the processing cycle as quickly as possible.
Trading Partnership code	A user-defined code that uniquely identifies a Trading Partnership record.
Trading Partnership	An arrangement with a specific trading partner to exchange information in a specific document type, described by a map file.
translation script	A Gentran:Server for UNIX script that calls the translator and other programs, such as <b>ediarc</b> and <b>envelope</b> .
upstream data manager	A data manager that processes files and then routes them to the work directory or queue of another data manager, which is called the downstream data manager.

## Data Manager Initialization Files

---

### Definition of initialization file

A data manager's **initialization file** is the configurable file that contains the data manager's personality and processing parameters. Every data manager type except the archive data manager has an initialization file.

### Reference

For information about setting the archive data manager's processing parameters, see the [Archiving Your Data](#) chapter in this guide.

---

### Purpose

The data manager's initialization file determines these main characteristics:

- ▶ Data manager's personality
- ▶ Directory or queue that the data manager scans for files
- ▶ Frequency with which the source directory or queue is scanned
- ▶ Type of processing action the data manager takes, including the way errors are handled
- ▶ Routing method (interchange, group, or set)
- ▶ Way in which a file's unique identification code is constructed
- ▶ Name of the archive directory and length of the archive period.

---

### File format

The initialization files are in ASCII text file format.

---

### Base initialization file

Gentran:Server has a default **base initialization file** that contains all the parameters for every data manager type except the arch (archive) personality.

When you access a data manager's initialization file, the system displays the initialization file that your organization configured for the data manager. If one is not yet configured, the system displays the base initialization file.

---

### Autocopy feature

If you copy a data manager, the system copies the initialization file and renames it for the new data manager.

---

(Continued on next page)

---

**Initialization file entries in the data manager log**

When a data manager starts, it records its initialization file parameters in the data manager's **log file**. You can view the log to see the parameters set for the data manager.

**Comment**

To view the log file, select the data manager from the Data Manager Control screen and press F7. See the [Monitoring Processes](#) chapter in this guide for more information.

---

**Path name**

All data manager initialization files are stored in the `$EDI_ROOT/conf.d` directory. Every initialization file has the suffix `.init`. The path name of an initialization file is

```
$EDI_ROOT/conf.d/<dmnm>.init
```

Where `<dmnm>` represents the name of the data manager.

**Example**

This is the path to the initialization file for the data manager `inbd`:

```
$EDI_ROOT/conf.d/inbd.init
```

---

# Initialization File Structure

## Overview

---

**Introduction** Initialization files contain symbols, a parameters section, an errors section, and a handlers section.

---

**Symbols** Symbols are used to designate certain elements in an initialization file.

This table describes the symbols and their functions.

Symbol	Function
#	Designates a comment line. A comment line is a phrase or sentence that explains the purpose or effect of the line that follows the comment. You can use as many comment lines as you want in the file. The comment delimiter (#) must appear in the first position (column one) of a line.
!	Designates a section in an initialization file. There are three sections: Parameters, Errors, and Handlers.
:	Designates a label for a parameter. You assign a value to a parameter to describe something you want done or define an item in the section.  <b>Example</b> Use the label :ARCHIVE_DIRECTORY to set the name of the directory that receives the data manager's archived data.  :ARCHIVE_DRIECTORY    arch

---

(Continued on next page)

**Sections** This table describes the three main sections in an initialization file and their functions.

Section	Function
!PARMS (Parameters)	<p>Sets the data manager's personality type, routing parameters, archive directory, allowance for duplicates, and other parameters such as the source type (directory or queue), the name of the data manager's work directory, and the scan delay period.</p> <p><b>Comment</b> The system sets any environment variables in the Parameters section at initialization time only. Changes you make to the data manager while it is running go into effect the next time it is started.</p>
!ERRORS	Sets the error classes.
!HANDLERS	<p>Describes how errors are handled.</p> <p><b>Examples</b> These are examples of actions a data manager can take to handle an error.</p> <ul style="list-style-type: none"> <li>▶ Move the file to a specified directory</li> <li>▶ Trigger a beeper</li> <li>▶ Call a script.</li> </ul> <p><b>Comment</b> The system sets environment variables from the Handlers error handler section during run time.</p>

### CAUTION

**Use each section name only once in an initialization file. For the best results, do not change the order in which the sections appear in the base initialization file.**

## Parameters Section

---

**Introduction** The Parameters section is the largest section. This section contains the parameters that determine how the data manager operates and which directories and files it uses.

---

**Subsections** The Parameters section consists of several subsections:

- ▶ Routing parameters
- ▶ Translator parameters (for xltr data managers)
- ▶ Line manager parameters (for lnmn data managers)
- ▶ Miscellaneous parameters
- ▶ Archive handling
- ▶ Duplicates checking.

**Reference**

For detailed descriptions of the parameters in each subsection, see [Parameters Subsections](#).

---

**Parameters subsection descriptions**

This table describes the subsections in the Parameters section.

Subsection	Function
Routing Parameters	Determines: <ul style="list-style-type: none"> <li>▶ Whether files are routed by interchange, group, or set, or by another method</li> <li>▶ How Life Cycle event files are handled</li> <li>▶ How the destination (output) file name is constructed.</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Subsection</b>	<b>Function</b>
Line Manager Parameters	<p>Sets processing information that is unique to line managers, which are used in Advanced Data Distribution operations:</p> <ul style="list-style-type: none"> <li>▶ Name of the directory to copy files from</li> <li>▶ Name of the script that initializes the line manager directories</li> <li>▶ Name of the directory in which the line manager runs</li> <li>▶ Name of the termination file</li> <li>▶ Line Manager execution command</li> <li>▶ Kill time for child processes.</li> </ul>
Translator Parameters	<p>Sets processing information that is unique to translator (<i>xltr/appt</i>) data managers:</p> <ul style="list-style-type: none"> <li>▶ Name of the translation script</li> <li>▶ Directory that the translator data manager or <i>xltr/appt</i> runs in</li> <li>▶ Run mode (application, record file layout, or edi)</li> <li>▶ File names and directory names to create and link to the run directory</li> <li>▶ Location of the Trading Partnership code in the input file</li> </ul>
Miscellaneous Parameters	<p>Sets basic processing values, including the:</p> <ul style="list-style-type: none"> <li>▶ Name of the directory or queue that the data manager looks in for files</li> <li>▶ Name of the directory that contains the transaction register</li> <li>▶ Number of seconds in the scan period</li> <li>▶ Data manager's personality type</li> <li>▶ Other pieces of information the data manager needs to process data.</li> </ul>
Archive Handling	<p>Determines how the data manager handles archiving. Sets the:</p> <ul style="list-style-type: none"> <li>▶ Name of the archive directory</li> <li>▶ Number of days archived data is held in the archive directory</li> <li>▶ Archiving of error files.</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

(Contd) Subsection	Function
Duplicates Checking	<p>Determines how the data manager handles duplicate data:</p> <ul style="list-style-type: none"> <li>▶ Allow, disallow, or not check for duplicate data</li> <li>▶ Use the direction of the data (in or out) as a transaction register key.</li> </ul>

### Parameters general format

This is the general format within a Parameters subsection.

```
#comment_line
:label_name      parameter_value
```

This table describes the parts of the format.

Part	Description
#comment_line	The line of text that explains the parameter setting on the next line.
:label_name	The name of the parameter being set.
parameter_value	The value of the parameter.

#### Comment

You can have any amount of space between the label\_name and the parameter\_value.

#### Note

For certain parameters, such as the LIFECYCLE\_EXEC\_LINE, the parameter\_value is on the line that follows the label\_name.

### Example 1

This is an example of the format for the parameter that sets the data manager's work directory.

```
# Configuration and work file location, default = base_dir
:WORK_DIRECTORY      inbd
```

(Continued on next page)



**Example 2**

---

This is an example of the format for the parameter that sets the command for executing Life Cycle.

```
:LIFE_CYCLE_EXEC_LINE  
#          lclld -flclld/base
```

---

## Errors Section

---

**Introduction** The Errors section lists the error classes that the data manager handles.

---

**General format** This is the general format of the Errors section.

```

:error_type          handler_name          class

```

This table describes the parts of the format.

Part	Description
:error_type	<p>The name of the error. All error types are defined for you in the base initialization file.</p> <p><b>Reference</b> See the <a href="#">Standard Error Types</a> topic in this chapter for a list of error types.</p> <p><b>Comment</b> If the error type begins with E_, the error causes the data manager to terminate. For example, the E_FILE_ACCESS error causes the data manager to terminate because a file could not be opened or read.</p>
handler_name	<p>The name of the handler for this kind of error and class.</p> <p><b>Comment</b> You use the handler_name in the first position of the Handlers section when you define how you want the data manager to handle the error type.</p>
class	<p>The error class for the error type and error handler combination. You determine the class name.</p> <p>You use error classes in configuration records to enable a data manager to handle the same type of error differently for different trading partners or file names.</p> <p><b>Reference</b> See <a href="#">How to Use an Error Class</a> for more information.</p>

---

(Continued on next page)

---

**Errors section  
examples****Example 1**

This is an example of the format for a duplicate error in the Errors section.

```
:DUPLICATE          dupl_a          default
```

In this example, the error\_type is DUPLICATE. The error is assigned to the handler action dupl\_a. The error class is default.

**Example 2**

This is an example of the format for a no Trading Partnership code error in the Errors section.

```
:NO_TP              notp            class1
```

In this example, the error\_type is NO\_TP. The error is assigned to the handler notp. The error class is class1.

**Example 3**

This is an example of the format for duplicate errors with two error classes.

```
:DUPLICATE          dup            default
:DUPLICATE          dup_special    special
```

---

## Standard Error Types

This table describes the standard error types.

**Note**

You cannot modify the standard error types.

Error Type	Description
DUPLICATE	Duplicate data found.
NO_CONFIG	No configuration record exists.
NO_TP	No Trading Partnership record exists.  <b>Note</b> If your system is configured to use the Trading Partnership Rules function to determine the Trading Partner Code, the system generates this error when it is unable to find an entry in the Cross Reference Table for the string and Trading Partner Code.
STRUCT_ERROR	The data is not in the correct format.
TRANSLATION_FAILED	The translation process failed. It may be because the script could not be run, the environment variables could not be set, or some other reason. Used with xltr/appt data managers only.
Q_FILE_MISSING	The data manager could not find a queued file.  <b>Possible causes</b> <ul style="list-style-type: none"> <li>▶ The file name may be invalid.</li> <li>▶ The file no longer exists.</li> <li>▶ The file was deleted prior to the queue update.</li> </ul>
ROUTING_ERROR	The file cannot be routed to the destination directory or mailbox.
DEST_Q_ERROR	The data manager could not write to the destination queue.
SRC_Q_ERROR	The data manager could not read from the source queue.

(Continued on next page)

<b>(Contd) Error Type</b>	<b>Description</b>
APP_ERROR	The application data manager could not access and retrieve the application file. Used with apm data managers only.
EXEC_ERROR	The data manager could not execute a child process.
INVALID_FN	A file name is invalid.
FA_ERROR	Error processing a functional acknowledgment.
E_ISAM	An ISAM file (such as the transaction register, the Trading Partner Configuration file, or the Advanced Data Distribution distribution list) could not be opened, read, or written to.
E_FILE_ACCESS	<p>A file could not be found or it could not be opened or read. The file can be any of the files called in the process, including temporary files.</p> <p><b>Note</b> Do not use any of the move actions (move_e, move_e_mm, or move_b) with this error.</p>
E_RW_SAME_QUEUE	The data manager is reading and writing to the same queue, causing a loop condition.
E_WRITE_ERROR	The data manager could not write to the output file or log.
E_MEM_ERROR	The data manager was not able to allocate memory.
E_LOCK_FAIL	<p>The number of LOCK_ATTEMPTS has been exceeded, but the data manager cannot access the input file because the file is still locked by another process.</p> <p><b>Note</b> Do not use any of the move actions (move_e, move_e_mm, or move_b) with this error.</p>
EXEC_FAIL_ERROR	<p>A child process failed.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Error Type</b>	<b>Description</b>
E_LINE_MANAGER	Line manager error detected. This error can occur if: <ul style="list-style-type: none"><li>▶ The system couldn't copy scripts to the run directory</li><li>▶ The initialization script failed</li><li>▶ Other problems listed in the data manager's log cause the line manager process to fail.</li></ul> Used only with <i>Inmn</i> data managers.
E_INIT_XLTR	Translation error. This error can occur if the system could not copy scripts to the run directory or for other reasons. The data manager log shows the cause of the failure. Used only with <i>xltr</i> and <i>appt</i> data managers.
NO_XML_SPLIT_TABLE_ERROR	The data manager could not find an entry for the XML elements in the XML splitter tables, <i>xmlspl1</i> , <i>xmlspl2</i> , and <i>xmlspl3</i> .

---

# Handlers Section

---

**Introduction** The Handlers section describes the actions to take when errors occur.

---

**General format** This is the format of the Handlers section.

```
:handler_name      action      dir/file_name or script_name
```

This table describes the parts in the format.

Part	Description
:handler_name	<p>The name of the handler. Handler names are not pre-defined. You can use any name for the handler.</p> <p><b>Examples</b> These are examples of handler names.</p> <ul style="list-style-type: none"> <li>▶ dup</li> <li>▶ exec_fail_error</li> <li>▶ notp</li> <li>▶ no_config</li> <li>▶ struct_e</li> <li>▶ q_err</li> </ul> <p><b>Comment</b> The handler name comes from the second position in the Errors section.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Part</b>	<b>Description</b>	
action	The action to be taken when this error is encountered. Use this table to select an action. These are the only actions available.	
	<b>Select</b>	<b>To</b>
	move_e	Move errors to directory specified in the third field.
	move_m	Move messages on the lines that follow the current line to the directory specified in the dir/file_name field.
	move_e_mm	Move errors to specified directory and mails the message that is on the line that follows this line.
	move_b	Move these items to the directory specified in the dir/file_name field: <ul style="list-style-type: none"> <li>▶ Data in error</li> <li>▶ The message that is on the line that follows this line.</li> </ul>
	script	Run the script specified in the dir/file_name field.
	beep	Invoke the beeper script and triggers the beeper number on the line that follows this line.  <b>Comment</b> You must type the beeper number on the next line, and you must type the name of the beeper script in the script_name position.
	ignore	Ignore the error; does not record it.
	mm	Mail the message specified on the lines that follow this line.  <span style="color: red;">(Continued on next page)</span>



(Contd) Part	Description
	<p><b>CAUTION</b></p> <p><b>Some actions do not apply to certain handlers. For example, you cannot use the <code>move_e</code> action for an <code>E_FILE_ACCESS</code> error. This is because the error indicates that the file cannot be accessed. If <code>Gentran:Server</code> cannot access the file, it cannot move it.</b></p> <p>Similarly, you should not use a move file action with the <code>E_LOCK_FAIL</code> handler. This is because <code>Gentran:Server</code> has to lock the file in order to move it.</p> <p>In both these examples, a valid action would be to have the handler send a mail message or run a script.</p>
dir/file_name	<p>The name of the destination directory and file name for the data in error or messages. If the action in the second field is <code>move_e</code>, <code>move_m</code>, <code>move_e_mm</code>, or <code>move_b</code>, this is the name of the file that receives the data set.</p> <p><b>WARNING</b></p> <p><b>If you want mail messages for <code>move_e_mm</code> actions to contain the file name to which the data in error was sent, use the environment variable <code>DEST_FN</code> in the <code>mail_proc</code> script.</b></p>
script_name	<p>If the action in the second field position is “script” or “beep,” this is the name of the script to be run.</p> <p><b>WARNING</b></p> <p><b>The script action runs only <code>Gentran:Server</code> scripts.</b></p>

**Handlers section example**

This is an example of the format for describing how to handle an error.

```
:e_struct          move_e          error/estruct
```

In this example, the data manager is instructed to move data with structural errors to the `error` directory and name the file `estruct`.

# Parameters Subsections

## Overview

---

**In this section**

The Parameters section controls the operations of the data manager. This section describes the Parameters subsections. Each topic describes the controls in one of the Parameters subsections.

---

**Parameter types**

There are six parameter subsections:

- ▶ Routing
- ▶ Line Manager
- ▶ Translator
- ▶ Miscellaneous
- ▶ Archive handling
- ▶ Duplicates checking.

---

**Start and end markings**

The start and end of each subsection is marked with a comment line.

**Example**

```
##### Routing Parameters start #####
```

---

# Parameters and Personality Table

**Introduction** The parameters you can use in a data manager’s initialization file depend on the data manager’s personality.

**WARNING**

**Data managers recognize only those parameters their personality is designed to use. If you include a parameter in a data manager’s initialization file that its personality does not recognize, the data manager ignores the parameter.**

**Summary table** This table summarizes the parameters you can use with each data manager personality type. The parameters are arranged by section and subsection.

**Note**

You must have the Gentran:Server XML translation option to use the xml data manager.

Parameter	Personality								
	inbd	dnld	file	hcmd	xltr	Inmn	appm	xml	ncpdp
<b>Parameters Section</b>									
<b>Routing</b>									
ROUTING_METHOD	X	X					X		X
MBAG_CONSTRUCT	X	X	X	X	X		X	X	X
ADD_MBAG	X	X	X	X	X		X	X	X
MAIL_PREFIX_CONSTRUCT	X	X	X	X	X		X	X	X
MAIL_PREFIX_NAME	X	X	X	X	X		X	X	X
DIST_LIST_CONSTRUCT	X	X	X	X	X		X	X	X
UNIQUE_FILE_NAMES	X	X	X	X	X		X	X	X
NOTIFY	X	X	X	X	X		X	X	X
USE_FN_PRE_ON_ERR	X	X	X	X	X		X	X	X
(Continued on next page)									

(Contd) Parameter	Personality								
	inbd	dnld	file	hcmd	xltr	Inmn	appm	xml	ncpdp
LIFE_CYCLE_DIR	X	X	X	X	X		X	X	X
MULTIPLE_LIFE_CYCLE_FILES	X	X	X	X	X		X	X	X
LIFE_CYCLE_CALL_SWITCH	X	X	X	X	X		X	X	X
LIFE_CYCLE_EXEC_LINE	X	X	X	X	X		X	X	X
REPORT_SET_ERRORS	X								
REPORT_SETID_ERRORS	X								
TRIM_TERMINATORS	X								
BINARY_DATA	X								
<b>Line Manager</b>									
LM_MODEL_DIR						X			
LM_INIT_MODEL						X			
LM_RUN_DIR						X			
LM_TERMINATE						X			
LM_EXEC_LINE						X			
LM_GRACE_SECS						X			
<b>Translator</b>									
XL_MODEL_SCR					X				
XL_RUN_DIR					X				
XL_MODE					X				
XL_LINK_FILES					X				
XL_LINK_DIRS					X				
XL_MAKE_DIRS					X				
XL_ARGS					X				
(Continued on next page)									

(Contd) Parameter	Personality								
	inbd	dnld	file	hcmd	xltr	Inmn	appm	xml	ncpdp
<b>Miscellaneous</b>									
WORK_DIRECTORY	X	X	X	X	X		X	X	X
WORK_TYPE	X	X	X	X	X		X	X	X
DM_LOCK	X	X	X	X	X		X	X	X
TRANS_REGISTER_DIR	X	X	X	X	X		X	X	X
TRADING_PARTNER_DIR	X	X					X	X	X
RESOURCE_GROUP	X	X	X	X	X		X	X	X
VAL_RESOURCE_GRP	X	X	X	X	X		X	X	X
SCAN_DELAY	X	X	X	X	X	X	X	X	X
STAT_CHECK	X	X	X	X	X	X	X	X	X
LOCK_ATTEMPT	X	X	X	X	X	X	X	X	X
PERSONALITY	X	X	X	X	X	X	X	X	X
CMD_STR				X					
CMD_STR_ENV				X					
SWAP_ID_FLAG	X								
USE_RECON_IDS	X	X					X	X	X
SUBSTITUTE_NEWLINE	X								
ACCEPT_NEWLINE	X								
FA_MAIL	X								
FA_HANDLER	X								
ADD_CONFIG	X	X	X	X	X		X	X	X
ADD_TP	X								
SEND_MAIL_TP	X								
CHK_PREFIX			X		X				

(Continued on next page)

(Contd) Parameter	Personality								
	inbd	dnld	file	hcmd	xltr	Inmn	appm	xml	ncpdp
ADD_MBAG_TO_ DOC_REF_NUM			X		X				
REC_ID_MISS							X		
INPUT_FN_APP_TP					X		X		
USE_MOD		X							
SCRIPT_RUN_SWITCH	X	X	X	X	X	X	X	X	X
ENCLOSING_CHARACTER							X		
PROCESS_TA1	X								
FULL_TP_TEST	X								
<b>Archive handling</b>									
ARCHIVE_DIRECTORY	X	X	X	X	X		X	X	X
ARCHIVE_PERIOD	X	X	X	X	X		X	X	X
ARCHIVE_ERROR	X	X	X	X	X		X	X	X
<b>Duplicates checking</b>									
IN_OUT	X	X	X	X	X		X	X	X
ALLOW_DUPLICATES	X	X	X	X	X		X	X	X
<b>Errors Section</b>									
DUPLICATE	X	X	X	X	X		X	X	X
NO_CONFIG	X	X	X	X	X		X	X	X
NO_TP	X	X		X			X	X	X
STRUCT_ERROR	X	X		X			X	X	X
TRANSLATION_FAILED					X				
Q_FILE_MISSING	X	X	X	X	X		X	X	X
ROUTING_ERROR	X	X	X	X	X		X	X	X
(Continued on next page)									

(Contd) Parameter	Personality								
	inbd	dnld	file	hcmd	xltr	Inmn	appm	xml	ncpdp
DEST_Q_ERROR	X	X	X	X	X		X	X	X
SRC_Q_ERROR	X	X	X	X	X		X	X	X
APP_ERROR							X		
EXEC_ERROR	X	X	X	X	X		X	X	X
INVALID_FN	X	X	X	X	X		X	X	X
FA_ERROR	X								
E_ISAM	X	X	X	X	X		X	X	X
E_FILE_ACCESS	X	X	X	X	X		X	X	X
E_RW_SAME_QUEUE	X	X	X	X	X		X	X	X
E_WRITE_ERROR	X	X	X	X	X		X	X	X
E_MEM_ERROR	X	X	X	X	X		X	X	X
E_LOCK_FAIL	X	X	X	X	X		X	X	X
EXEC_FAIL_ERROR	X	X	X	X	X		X	X	X
E_LINE_MANAGER						X			
E_INIT_XLTR					X				
NO_XML_SPLIT_TABLE								X	
<b>Handlers Section</b>									
dup	X	X	X	X	X		X	X	X
no_config	X	X	X	X	X		X	X	X
notp	X	X	X	X	X		X	X	X
struct_e	X	X		X			X	X	X
routing_e	X	X	X	X	X		X	X	X
dest_q_err	X	X	X	X	X		X	X	X
q_err	X	X	X	X	X		X	X	X

(Continued on next page)

(Contd) Parameter	Personality								
	inbd	dnld	file	hcmd	xltr	lnmn	appm	xml	ncpdp
q_file_missing	X	X	X	X	X		X	X	X
default	X	X	X	X	X		X	X	X
beep_a	X	X	X	X	X		X	X	X
exec_fail_error	X	X	X	X	X		X	X	X
exec_error	X	X	X	X	X		X	X	X
ignore	X	X	X	X	X		X	X	X
init_xltr					X				
trans_failed					X				
lm						X			
e_file	X	X	X	X	X		X	X	X
no_split_table								X	

**CAUTION**

**The handler names listed are examples listed in the base initialization file. Your organization determines the actual handler names that you will use.**

---



## Routing Parameters

This table describes the routing parameter functions. The parameters are in the same order as they are in the base initialization file

Parameter	Function
ROUTING_METHOD	<p>Determines whether data is divided and routed by interchange, group, or set. The default is 2.</p> <ul style="list-style-type: none"> <li>▶ 0 = Interchange</li> <li>▶ 1 = Group</li> <li>▶ 2 = Set</li> </ul> <p>Only the application (appm), download (dnld), and inbound (inbd) type data managers use this parameter. The other data managers use the routing information in the Trading Partnership record to determine how to route the file.</p> <p><b>NCPDP data manager only</b> Determines whether data is divided and routed by batch header, transaction header service ID and qualifier, or transaction header transaction code. The default is 2.</p> <ul style="list-style-type: none"> <li>▶ 0 = Batch Header</li> <li>▶ 1 = Transaction Header Service ID/Qualifier</li> <li>▶ 2 = Transaction Header Transaction Code</li> </ul>
MBAG_CONSTRUCT	<p>Determines where the mailbag ID comes from. The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = Use the mailbag ID (up to six characters) that is on the input file name. The mailbag ID follows the first period after the file name <i>filename.mbagid.xxxx</i></li> <li>▶ 1 = Generate new mailbag ID</li> </ul> <p><b>Note</b> Gentran: Server ignores this parameter if the ADD_MBAG parameter is set to 0.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Function</b>
ADD_MBAG	<p>Determines whether the mailbag ID is used in the destination file name during routing. The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul>
MAIL_PREFIX_CONSTRUCT	<p>Determines how the mail prefix that is added to outbound mailbox file names is constructed. The default is 3.</p> <ul style="list-style-type: none"> <li>▶ 0 = Use the MAIL_PREFIX_NAME</li> <li>▶ 1 = Use the prefix that is on the input file name (for example, edi)</li> <li>▶ 2 = Use the destination file name</li> <li>▶ 3 = Use the name of the data manager personality type (for example, hcnd)</li> </ul> <p><b>Note</b> Use this parameter only with the Advanced Data Distribution product.</p>
MAIL_PREFIX_NAME	<p>Defines the mail prefix name to be used (if MAIL_PREFIX_CONSTRUCT is 0). The default is udf.</p> <p><b>Note</b> Use this parameter only with the Advanced Data Distribution product.</p>
DIST_LIST_CONSTRUCT	<p>Determines whether a unique extension is appended to the mailbox file name when multiple copies of the same file are added to several Advanced Data Distribution mailboxes. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul> <p><b>Note</b> Use this parameter only with the Advanced Data Distribution product.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Function</b>
UNIQUE_FILE_NAMES	<p>Determines whether a unique extension is appended to the destination file names. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul>
NOTIFY	<p>Turns real-time data processing (IPC Trigger) on or off. Works with the SCAN_DELAY and STAT_CHECK parameters.</p> <p>If this parameter is on, the upstream data manager writes the file to the destination directory and attempts to open the FIFO. If the downstream data manager is polling the file, the upstream data manager opens the file and writes to the FIFO. This notifies the downstream data manager that a file is ready to be processed.</p> <p>Both the upstream and downstream data managers must have NOTIFY set to 1 for the IPC Trigger to work. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 1 = On</li> <li>▶ 0 = Off</li> </ul> <p><b>References</b> See the <a href="#">Configuring for Real-time Processing</a> section in this chapter.</p> <p>See the descriptions for SCAN_DELAY and STAT_CHECK.</p>
USE_FN_PRE_ON_ERR	<p>Determines whether the input file name prefix is used when an error file is routed. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 1 = Yes</li> <li>▶ 0 = No</li> </ul> <p><b>Note</b> If set to 0 (No), the HANDLER determines the prefix of the routed error file.</p>
LIFE_CYCLE_DIR	<p>Defines the name of the directory in which the Life Cycle event files are placed for processing. The default is <b>lclld</b>.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Function</b>
MULTIPLE_LIFE_CYCLE_FILES	<p>Determines whether all event records are written to one event file or whether event records are sorted according to the routing method and each is written to its own Life Cycle event file. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul> <p><b>Comments</b></p> <ul style="list-style-type: none"> <li>▶ If set to 0 (No), all event records are written to one event file. The file is named &lt;xxxx&gt;.v, where xxxx is the name of the data manager.</li> <li>▶ If set to 1 (Yes) Gentran:Server writes the event records to multiple, separate event files, based on the routing method in the ROUTING_METHOD parameter. To identify the files, the system appends a unique ID to each file name. The file name is &lt;xxxx&gt;.v.&lt;uniqid&gt;, where xxxx is the name of the data manager and uniqid is the system-generated identifier.</li> </ul> <p><b>CAUTION</b></p> <p><b>If you set the MULTIPLE_LIFE_CYCLE_FILES parameter to 1, you must create a separate shell script that calls the Life Cycle load program (lcl) multiple times for each Life Cycle file.</b></p>
LIFE_CYCLE_CALL_SWITCH	<p>Determines the point at which the Life Cycle command is run. The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = Execute after each interchange, group, or set is routed, based on the ROUTING_METHOD</li> <li>▶ 1 = Execute at the end of each input file</li> <li>▶ 2 = Execute at the end of each process cycle (after all the files in the queue or work directory are processed and before the data manager sleeps)</li> </ul> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Function</b>
LIFE_CYCLE_EXEC_LINE	<p>Defines the Life Cycle command you want executed. Type the command on the line that follows the LIFE-CYCLE_EXEC_LINE label.</p> <p><b>Comment</b> The command can update the Life Cycle table only once with one event file.</p> <p><b>CAUTION</b> <b>If you set the MULTIPLE_LIFE_CYCLE_FILES parameter to 1, you must create a separate shell script that calls the Life Cycle load program (lcl) multiple times for each Life Cycle file.</b></p>
REPORT_SET_ERRORS	<p>Applies when routing is by group or interchange. When this parameter is set to 1, the data manager validates the segment that follows an SE segment and validates the segment count within the set.</p> <p>The data manager reports a structural error if the segment that follows SE is not ST, GE, or UNZ (for EDIFACT documents). The data manager reports a warning if it finds an incorrect set count. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul>
REPORT_SETID_ERRORS	<p>Applies when routing is by group or interchange. When this parameter is set to 1, the data manager reports a structural error if a group contains more than one type of transaction sets.</p> <p><b>Example</b> If this parameter is set to 1, the system reports a structural error if a group contains both an 810 transaction set and an 858 transaction set.</p> <p style="text-align: right;"><i>(Continued on next page)</i></p>

<b>(Contd) Parameter</b>	<b>Function</b>
TRIM_TERMINATORS	<p>Removes extra segment terminators from data, and does not count the segment terminator as an additional segment.</p> <p>When this parameter is set to 1, the data manager removes extra segment terminators. If this parameter is set to 0, the data manager writes a newline character for each segment terminator and increments the segment count for the set. The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul>
BINARY_DATA	<p>Determines how an inbound data manager in a process flow handles a binary segment. The default value is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = Treat the binary segment just like other segments. Both the input and output files will have the name of the file that contains the binary data. Used for an inbound data manager in an inbound flow in which you call the edifmat command to preprocess the data. Edifmat strips the binary data and replaces it with a filename for the binary data.</li> <li>▶ 1 = Strip the binary data in the input file and replace it with the filename that contains the binary data. The output file will contain the filename. Used for the inbound data manager in an inbound flow in which edifmat is NOT used to preprocess the data.</li> <li>▶ 2 = Read the inbound binary data and put it in the output file. Used for inbound data managers in an outbound flow. Both the input and output files will contain binary data.</li> </ul>

# Line Manager Parameters

**Introduction** A line manager is a specialized data manager used with the optional Advanced Data Distribution system.

**Line manager parameters** This table describes the line manager parameter functions.

This parameter	Defines the
LM_MODEL_DIR	Name of the directory from which scripts that are necessary to run line manager processes are copied. The scripts are copied into the run directory. The default is abcnet.
LM_INIT_MODEL	Name of the script that initializes the line manager directories for operations. The default is model_ainit.
LM_RUN_DIR	Name of the directory in which the line manager runs. The default is tty2.
LM_TERMINATE	Name of the termination file. The system creates the termination file when it receives a signal to stop the script named in the LM_EXEC_LINE parameter. The presence of the termination file in the run directory triggers the script to stop processing. The default is terminate_file.
LM_EXEC_LINE	Line manager execution command. The command includes the name of the script to be run and the name of the termination file. The default is bin/smgr -stestlm -eLM_RUNDIR=tty2 -eLM_TERMINATE=terminate_file.
LM_GRACE_SECS	Time (in seconds) between these two events: <ul style="list-style-type: none"> <li>▶ The line manager receives a signal from the Foreground Manager to stop the child process.</li> <li>▶ The system stops the child process.</li> </ul> The default is 30.

## Translator Parameters

This table describes the translator parameter functions.

Parameter	Function
XL_MODEL_SCR	Defines the name of the translation script. The default is <i>model_xltr</i> .
XL_RUN_DIR	<p>Defines the subdirectory of \$EDI_ROOT in which translation occurs. The default is <i>xltr</i>. The subdirectory is usually named for the data manager (&lt;data_manager_name&gt;_run_dir). Each translation data manager should have its own run directory.</p> <p><b>Examples</b>  <i>xltr_run_dir</i>  <i>xlto_run_dir</i></p>
XL_MODE	<p>Defines the type of files this translation data manager processes. The default is 2.</p> <ul style="list-style-type: none"> <li>▶ 1 = Application</li> <li>▶ 2 = Record file layout, EDI, or XML (if you have the XML option)</li> </ul> <p><b>Note</b>            If set to 2, Gentran:Server does not pass the application file name or the Trading Partnership code to the <b>lftran</b> command line in the translation script. See the XL_ARGS parameter.</p>
XL_LINK_FILES	<p>Defines the files to link into the XL_RUN_DIR directory.</p> <p><b>Note</b>            You must enter the value on the line that follows the label_name.</p>
XL_LINK_DIRS	<p>Defines the directories to link into the XL_RUN_DIR directory.</p> <p><b>Note</b>            You must enter the value on the line that follows the label_name.</p> <p style="text-align: right;">(Continued on next page)</p>



<b>(Contd) Parameter</b>	<b>Function</b>
XL_MAKE_DIRS	<p>Defines the directories to be created in the XL_RUN_DIR directory. There are no default values.</p> <p><b>Note</b> You must enter the value on the line that follows the label_name.</p>
XL_ARGS	<p>Determines whether the data manager passes the Trading Partnership code or the application file name to the application translator.</p> <p>If the Trading Partnership code is passed, it is used for all sets in the input file. If the application file name is passed, the Trading Partnership code is defined in the application file. The default is 0.</p> <ul style="list-style-type: none"><li>▶ 0 = Pass application file name</li><li>▶ 1 = Pass Trading Partnership code</li></ul>

## Miscellaneous Parameters

This table describes the miscellaneous parameters.

Parameter	Description
WORK_DIRECTORY	<p>Defines the name of the directory that contains the files or queued file names that the data manager processes. Usually, you name this directory for the data manager. The default value is base.</p> <p>If the directory that you specify does not exist, the system prompts you to create it when you exit the editor.</p>
WORK_TYPE	<p>Defines the type of source the data manager looks at for files to process. If the work type is q, you may want to set the RESOURCE_GROUP parameter. The default is d.</p> <ul style="list-style-type: none"> <li>▶ q = Queue</li> <li>▶ d = Directory</li> </ul>
DM_LOCK	<p>Defines the file name that the system locks when the data manager is running. Locking the file name prevents the data manager from starting up again while it is running. If the file name does not exist; Gentran:Server creates it.</p> <p>You should have a unique DM_LOCK file name for each data manager. Use the name of the data manager in the file name.</p> <p><b>Example</b> LOCKS/xltr_lock</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
TRANS_REGISTER_DIR	<p>Defines the name of the directory in which the transaction register file resides.</p> <p>The default is EDI_ROOT. If you want the data manager to use a different transaction register, you must create the directory with the <b>TrnReg</b> option under the <b>Util</b> menu and then type the name here.</p> <p><b>Reference</b> See the <a href="#">Using Transaction Registers to Track Documents</a> chapter in this guide for instructions.</p>
TRADING_PARTNER_DIR	<p>Defines the name of the directory in which the trading partnership file (<i>tp.idx</i>, <i>tp.dat</i>) resides.</p> <p>If you do not specify a directory name, Gentran:Server extracts the directory from <i>envprim.cfg</i>.</p>
RESOURCE_GROUP	<p>If WORK_TYPE is <b>q</b>, defines the name of the resource group for the files that the data manager will process. Used to group files in a queue. The system sets the file's group name when a process makes an entry to the queue.</p> <p>The default value is <b>default</b>, but you must specify this value. If you use the default value for this parameter, the data manager processes all files in the queue that are assigned to the default RESOURCE_GROUP.</p>
VAL_RESOURCE_GRP	<p>Used with an upstream data manager to determine if the system validates the resource group before it writes to its destination queue. If this parameter is set to 1, the upstream data manager writes a processed file to the destination queue only if a downstream data manager's initialization file is configured to pick up files in the resource group named in the upstream data manager's configuration record.</p> <p>The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = No, do not validate the resource group</li> <li>▶ 1 = Yes, validate the resource group</li> </ul> <p style="text-align: right;"><b>(Continued on next page)</b></p>

<b>(Contd) Parameter</b>	<b>Description</b>
SCAN_DELAY	<p>Works in conjunction with the NOTIFY parameter:</p> <ul style="list-style-type: none"> <li>▶ If NOTIFY is 0, SCAN_DELAY defines the number of seconds the data manager sleeps after it finishes processing files in its queue or directory.</li> <li>▶ If NOTIFY is 1, SCAN_DELAY defines the number of seconds the downstream data manager waits to be notified that a data file has arrived in its work directory or queue.</li> </ul> <p>The default is 0. If you set SCAN_DELAY to 0, the data manager runs once. It starts, processes the files in its work directory or queue, and then stops.</p> <p><b>Real-time processing</b> For real-time processing, set SCAN_DELAY to either -1 or a value greater than 0 for the downstream data manager. This makes the downstream data manager poll the FIFO, process files when signaled that the upstream data manager has written to the FIFO, and then start polling again.</p> <p><b>Reference</b> See the <a href="#">Configuring for Real-time Processing</a> section in this chapter.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
STAT_CHECK	<p>Defines the frequency (in number of seconds) with which the data manager checks the input file's modification time.</p> <p>Viewed as the minimum time difference (in seconds) between the current system time and the file's modification or access time. Determines when a data manager will try to access the file.</p> <ul style="list-style-type: none"> <li>▶ If STAT_CHECK is 0, the data manager begins processing the file as soon as the file is written.</li> <li>▶ If STAT_CHECK is 1, the file must be at least 1 second old before the data manager begins to process it.</li> </ul> <p>The default is the value set in SCAN_DELAY. If NOTIFY = 1, STAT_CHECK must = 0 for the downstream data manager.</p> <p><b>Reference</b> See the <a href="#">How to Configure Data Managers for Real-Time Processing</a> topic in this chapter.</p>
LOCK_ATTEMPT	<p>Sets the number of times an input file can pass the STAT_CHECK test, but fail the file locking attempt, before the data manager terminates and the system performs the action specified in the E_LOCK_FAIL handler. The default is 3. A pass to the STAT_CHECK test means the modification time has not changed.</p> <p>If LOCK_ATTEMPT is set to 0, the data manager does not attempt to lock the input file. This allows processing of read-only files.</p> <p><b>CAUTION</b> <b>A value of 0 can cause files to be moved that are currently in use by FTP or by other data managers. If you set LOCK_ATTEMPT to 0 and you expect intermittent file update delays, you should increase the value for STAT_CHECK to avoid concurrency problems.</b></p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
PERSONALITY	<p>Defines the data manager's personality type.</p> <ul style="list-style-type: none"> <li>▶ 1 = file</li> <li>▶ 2 = inbd</li> <li>▶ 3 = dnld</li> <li>▶ 4 = hcmd</li> <li>▶ 5 = xltr</li> <li>▶ 6 = line manager</li> <li>▶ 7 = appm</li> <li>▶ 8 = xml</li> <li>▶ 9 = ncpdp</li> </ul>
CMD_STR	<p>Defines the character that delimits command cards within a file. You can use a control character such as the dollar sign (\$) value.</p> <p>Used with host command card (<b>hcmd</b>) personalities only.</p>
CMD_STR_ENV	<p>Defines the environment variable used to represent the command string in a host command card file.</p> <p>Used with host command card (<b>hcmd</b>) personalities only.</p>
SWAP_ID_FLAG	<p>Determines whether the inbound data manager swaps the sender and receiver identifiers. It is used when the data manager is inbound (type 2 personality), but the data is outbound. (For example, to send ANSI data outbound.)</p> <p>The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = Off</li> <li>▶ 1 = On</li> </ul> <p><b>Note</b> If the SWAP_ID_FLAG is set to 1, the inbound EDI data manager first tries to match a Trading Partnership record to the sender and receiver IDs in the order that they appear in the data. If no match is found, the data manager searches the Trading Partnership records for a match to the sender and receiver IDs in reverse order.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
USE_RECON_IDS	<p>Determines whether reconciliation IDs set in the Trading Partnership records are used in the Life Cycle event record instead of the extracted values.</p> <p>Use reconciliation IDs to handle reconciliation when the IDs on the inbound functional acknowledgments that your trading partner sends you are different from the IDs on the outbound document you send.</p> <p>Applies to inbd, dnld and appm personalities only.</p> <p>The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul>
SUBSTITUTE_NEWLINE	<p>Determines whether the system replaces the segment terminators in the data with the new line character (\n). This causes each segment to appear on a separate line, making them easier to read.</p> <p>The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = Off</li> <li>▶ 1 = On</li> </ul> <p><b>Note</b> The <b>lftran</b> program requires newline-terminated segments in the input file.</p>
ACCEPT_NEWLINE	<p>Determines whether the system accepts EDIFACT data that uses the new line character (\n) as a segment terminator.</p> <p>The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = Accept only the default terminator character</li> <li>▶ 1 = Accept both the new line character and the default terminator character</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
FA_MAIL	<p>Determines whether the system sends mail notice of a bad functional acknowledgment (997 received with a code of "R" or "P" in the AK901 element). The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = Off</li> <li>▶ 1 = On</li> </ul> <p><b>Note</b> FA_HANDLER must be on (active) for FA_MAIL to function.</p>
FA_HANDLER	<p>Determines whether functional acknowledgments (997s only) are reconciled and validated during processing.</p> <p>If set to 1, Gentran:Server checks functional acknowledgments for errors (does AK901 have R or P). Gentran:Server appends functional acknowledgment event records in error to the &lt;dmnm&gt;_fa.v file in \$EDI_ROOT.</p> <p>If parameter is set to 2, Gentran:Server checks functional acknowledgments (the AK9 segment) for errors, but does not create or append to a file.</p> <p>If a functional acknowledgment is in error AND the FA_MAIL parameter is on, Gentran:Server also sends a mail message. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = Off</li> <li>▶ 1 = On. Create or append to &lt;dmnm&gt;_fa.v file</li> <li>▶ 2 = On, but do not create &lt;dmnm&gt;_fa.v file</li> </ul> <p><b>Note</b> If FA_HANDLER and FA_MAIL are set to ON, Gentran:Server does not route functional acknowledgments with errors to the translator. For this reason, the system views the functional acknowledgment as late instead of in error.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Parameter</b>	<b>Description</b>
ADD_CONFIG	<p>Determines whether the data manager creates a configuration record when one does not exist.</p> <p>If the data manager encounters data with a Trading Partnership code that is not in the configuration file and this parameter is set to 1, the system:</p> <ul style="list-style-type: none"> <li>▶ Copies the default configuration record</li> <li>▶ Names it for the Trading Partnership code</li> <li>▶ Routes it accordingly.</li> </ul> <p>This assumes that you have set up a default configuration record (TPID=_default). The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = No, do not create the configuration record</li> <li>▶ 1 = Yes, create configuration record</li> </ul> <p><b>Reference</b> See the <a href="#">How the Inbound Data Manager Creates Records</a> topic in the <a href="#">Working with Configuration Records</a> chapter of this guide.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
ADD_TP	<p>Determines whether the inbound data manager triggers generation of Trading Partnership records. Used with inbound personality only.</p> <p>When this parameter is turned on and the Trading Partnership does not exist, Gentran:Server generates Trading Partnership records (based on a model record) and adds them to the Trading Partner maintenance file. You must have a model Trading Partnership record. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul> <p><b>Note</b> This parameter does not create interchange or group level records.</p> <p><b>References</b> See the <a href="#">Working with Trading Partnerships</a> chapter of the <i>Gentran:Server for UNIX and Workstation Application Integration User's Guide</i> for information about creating Trading Partnership records.</p> <p>See the <a href="#">How the Inbound Data Manager Creates Records</a> topic in the <a href="#">Working with Configuration Records</a> chapter of this guide for information about a model Trading Partnership record.</p>
SEND_MAIL_TP	<p>Determines whether Gentran:Server sends a mail notification when it creates Trading Partnership records through the ADD_TP parameter. Use with inbound personalities only. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
CHK_PREFIX	<p>Determines whether the data manager routes a file based on the file name prefix. This prefix must match the file name set up in the <b>Filename</b> field of the <b>Data Manager Configuration</b> screen. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = No, route by entire file name</li> <li>▶ 1 = Yes, route by file name prefix</li> </ul>
ADD_MBAG_TO_DOC_REF_NUM	<p>Determines how the document reference number is created for <i>file</i> or <i>xltr</i> type data managers. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = Document reference number is the date, without century, and time that the file was processed (&lt;datetime&gt;).</li> <li>▶ 1 = Document reference number is the mailbag ID followed by the date and time the file was processed (&lt;mbagid&gt;.&lt;datetime&gt;)</li> <li>▶ 2 = Document reference number is the date, with century, and time that the file was processed (&lt;datecenturytime&gt;)</li> <li>▶ 3 = Document reference number is the mailbag ID followed by the date, with century, and time the file was processed (&lt;mbagid&gt;.&lt;datecenturytime&gt;)</li> </ul>
REC_ID_MISS	<p>Determines how the data manager handles non-header records with undefined record IDs. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = Ignore</li> <li>▶ 1 = Remove record (delete it)</li> <li>▶ 2 = Handle as a structure error</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
INPUT_FN_APP_TP	<p>Determines what the input file name represents. The default is 0.</p> <ul style="list-style-type: none"> <li>▶ 0 = Input file name is the application description or file definition file name</li> <li>▶ 1 = Input file name is the Trading Partnership code</li> </ul> <p><b>Note</b> If this parameter is set to 0, the data manager first attempts to identify the Trading Partner Code by looking for a (1) TPCODE field in the first record of a file definition or for (2) the field marked as the Trading Partner Code in the application description.</p> <p>If the data manager cannot identify a field that contains the Trading Partner Code, it uses the Trading Partnership rules from the Application/Trading Partnership Rules table (apptptbl) to determine the Trading Partner code.</p>
USE_MOD	<p>For download (dnld) type data managers only. Determines whether the 15th character in the record is used as a Trading Partner Code modifier or as part of the Trading Partnership Code. The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = The 15th character is not a modifier; the first 15 characters in the record are used as the Trading Partnership code</li> <li>▶ 1 = The 15th character is a modifier, and is stripped from the output file</li> </ul>
SCRIPT_RUN_SWITCH	<p>Determines when the data manager runs the scripts specified in the configuration records. The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = Execute script for each file routed using ROUTING_METHOD</li> <li>▶ 1 = Execute script for all data in the input file</li> <li>▶ 2 = Execute script at the end of a process cycle (after all the files in the queue or work directory are processed and before the data manager sleeps)</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
ENCLOSING_CHARACTER	<p>Defines the character used to enclose fields in an input file; for example, quotation marks (“”).</p> <p>Used with appm data managers only.</p> <p><b>Note</b> Gentran:Server does not view the enclosing character as part of the Trading Partnership code and Record ID fields.</p>
PROCESS_TA1	<p>Determines the routing method of an interchange that contains a TA1 segment. The default is 1.</p> <ul style="list-style-type: none"> <li>▶ 0 = Pass the TA1 segment with the rest of the data</li> <li>▶ 1 = Route the TA1 segment enveloped only by an interchange header and trailer</li> </ul> <p><b>Note</b> If set to 1, the stripped-out TA1 segment is routed to the directory specified for the STRUCT_ERROR.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
FULL_TP_TEST	<p>Applies to documents you receive from a trading partner. Used with inbound personalities only. Determines how Gentran:Server searches for a matching Trading Partnership record.</p> <ul style="list-style-type: none"> <li>▶ 0 = Wildcard usage is not enabled</li> <li>▶ 1 = Sequentially substitute the \$ wildcard in key fields and test for a matching Trading Partnership record after each substitution.</li> </ul> <p><b>Comment</b> Set this parameter to 1 when you have these three conditions:</p> <ul style="list-style-type: none"> <li>▶ You want to use a single Trading Partnership record to process inbound documents from multiple trading partners</li> <li>▶ The documents that you receive from the trading partner have different values in one or more of the key fields</li> <li>▶ You want Gentran:Server to ignore the value in one or more key fields when it searches for the Trading Partnership record.</li> </ul> <p><b>Important</b> In the Trading Partnership record, you must enter the wildcard symbol (\$) into one or more key fields that will have different values. These are the key fields that accept the wildcard symbol:</p> <ul style="list-style-type: none"> <li>▶ Your Interchange ID</li> <li>▶ Your Group or Application ID</li> <li>▶ Trading partner's Interchange ID</li> <li>▶ Trading partner's Group or Application ID</li> <li>▶ Standard Version ID</li> <li>▶ Set ID.</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Description</b>
FULL_TP_TEST (contd)	<p><b>Example</b></p> <p>You receive invoices from ten different divisions of trading partner XYZ. Each division has a different Group ID. You want to process all invoices from XYZ under the same Trading Partnership record. To make this happen, you:</p> <ul style="list-style-type: none"><li>▶ Create the Trading Partnership record and enter the wildcard symbol (\$) into the Group ID field</li><li>▶ Set the FULL_TP_TEST parameter in the inbound data manager initialization file to 1.</li></ul>

---

## Archive Handling Parameters

This table describes the archive handling parameters and their functions.

Parameter	Function
ARCHIVE_DIRECTORY	Defines the name of the archive directory. The default is arch.
ARCHIVE_PERIOD	Defines the number of days that archived data is retained. The program <b>cl_arch</b> deletes archived entries older than this. The default is 90.
ARCHIVE_ERROR	Determines whether error files are archived. The default is 0. <ul style="list-style-type: none"><li>▶ 0 = No</li><li>▶ 1 = Yes</li></ul>

---



## Duplicates Checking Parameters

This table describes the duplicates checking parameters.

Parameter	Description
IN_OUT	<p>Determines the direction of the data that the data manager processes. Gentrans:Server passes the value to the transaction register and to the Life Cycle table. The default is i.</p> <ul style="list-style-type: none"><li>▶ i = In</li><li>▶ o = Out</li></ul>
ALLOW_DUPLICATES	<p>Determines whether the data manager treats duplicate data as an error. If you allow duplicates, the system overwrites the previously received data's date/time field in the transaction register with the duplicate data's date/time. The default is 0.</p> <ul style="list-style-type: none"><li>▶ 0 = Do not allow duplicates</li><li>▶ 1 = Allow duplicates; update transaction register with new time</li><li>▶ 2 = Allow duplicates; do not write to or update transaction register records</li></ul>

---

# Maintaining Data Manager Initialization Files

## Overview

---

**In this section** This section contains procedures for accessing, editing, copying, and deleting a data manager's initialization file.

---

**Initialization file rules** These are the general rules you must follow when working with an initialization file:

- Use each section name only once.
- Do not change the standard order of the sections.
- Leave at least one space or one tab insertion between a label name and parameter value.

---

**Default values** Many parameters have a **default value**, which Gentran:Server uses if you do not specify another value. The comment lines that precede the parameter label usually list the possible values and the default value.

---

# How to Access and Edit an Initialization File

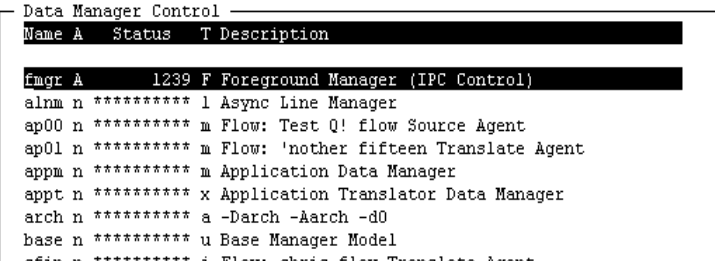
**Introduction** This topic explains how to view and modify the parameters in a data manager's initialization file.

**CAUTION**

**This procedure includes a data validation function. For this reason, you should use only this procedure to access and edit initialization files. If you use another editing method to edit an initialization file, the system will not validate your data.**

**Changes to parameters** Gentran:Server sets any parameters in the initialization file at initialization time only. If you edit the initialization file in a text editor outside of Gentran:Server, changes you make to the data manager while it is running go into effect the next time the data manager starts. If you edit the file inside Gentran:Server, the system requires you to stop the data manager before you edit its initialization file.

**Accessing and editing an initialization file** Use this procedure to view or edit an initialization file.

Step	Action
1	Select <b>DataMgr</b> from the Gentran:Server host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.
2	Select <b>fmgr</b> , the Foreground Manager.   <pre> Data Manager Control Name A Status T Description ----- fmgr A 1239 F Foreground Manager (IPC Control) alnm n ***** l Async Line Manager ap00 n ***** m Flow: Test Q! flow Source Agent ap01 n ***** m Flow: 'nother fifteen Translate Agent appm n ***** m Application Data Manager appt n ***** x Application Translator Data Manager arch n ***** a -Darch -Aarch -d0 base n ***** u Base Manager Model </pre> <p style="text-align: right;">(Continued on next page)</p>



(Contd) Step	Action
3	<p>Press F5 to configure the Foreground Manager data managers.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.</p> <pre data-bbox="662 556 1393 735"> dmc Data Manager Configuration Name A      Status  T Description ----- alnm n      0 1 Async Line Manager ap00 n 0      m Flow: Test Q! flow Source Agent ap01 n 0      m Flow: 'nother fifteen Translate Agent appm n      0 m Application Data Manager appt n      0 x Application Translator Data Manager </pre>
4	<p>Select the data manager you want to edit.</p> <p><b>WARNING</b> <b>You cannot edit an active data manager. Before you continue, check the Status field to make sure the data manager is not running. Inactivate the data manager by pressing F3 and then edit it.</b></p>
5	<p>Press F5 to edit the data manager.</p> <p><b>System Response</b> Gentran:Server displays the Edit screen for the selected data manager.</p> <pre data-bbox="641 1207 1401 1312"> Edit Name A      Status  T Description ----- alnm <u>n</u>      0 1 Async Line Manager F5:EditInit F9:Quit F10:Save </pre>
6	<p>Press F5 to edit the initialization file.</p> <p><b>System Response</b> Gentran:Server calls the vi editor or the editor set in the \$EDITOR variable and displays the initialization file. If an initialization file does not exist for the data manager, Gentran:Server displays the base initialization file.</p>
7	<p>Use the standard editor keys to edit the file.</p>
8	<p>Save your changes.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
9	<p>Exit the editor.</p> <p><b>System Response</b> Gentran:Server redisplay the Data Manager Control screen.</p> <p><b>CAUTION</b> <b>When you change a data manager's initialization file using the procedure described in this guide, Gentran:Server validates the new parameters and records the results in the data manager's log file. The change is logged under the server process ID number. The log entry contains the initialization file parameters.</b></p>

---

## How to Copy or Delete an Initialization File

---

### Copying an initialization file

To copy an initialization file, you must copy a data manager. When you copy a data manager, Gentran:Server also creates a copy of the initialization file. When you name the copy of the data manager, Gentran:Server applies the same name to the initialization file.

#### Reference

See the [Working with Data Managers](#) chapter in this guide.

---

### Deleting an initialization file

To delete an initialization file, delete the data manager. When you delete a data manager, Gentran:Server deletes the data manager's initialization file and mail\_proc file.

#### Reference

See the [How to Delete a Data Manager](#) topic in the [Working with Data Managers](#) chapter.

---

# Configuring Initialization Files for Specific Functions

## Overview

---

**In this section**

This section contains several topics that address setting initialization file parameters to obtain specific results:

- How to Set the Destination File Name
  - How to Set Error Notification
  - How to Use a UNIX Mail Script to Send Messages
  - How to Use an Error Class
  - How to Control Concurrent Processing with File Locking
-

## How to Set the Destination File Name

### Introduction

This topic describes how the destination file name is constructed. It contains procedures for setting parameters in the initialization file that control the:

- ▶ Destination file name
- ▶ Outbound mail file name
- ▶ Error file name.

### Parts of the destination file name

The destination file name can have up to three parts:

```
<file>.<mailbagid>.<uniqueid>
```

This table describes where the parts of the destination file name originate.

Part	Origin
<file>	The Destination File Name field in the configuration record.  <b>Reference</b> See the <a href="#">Working with Configuration Records</a> chapter in this guide for more information.
<mailbagid>	Optional. May come from the input file name or it can be generated, depending on how the initialization file is set. The ADD_MBAG and MBAG_CONSTRUCT parameters control whether or not the mailbagid is used in the destination file name.
<uniqueid>	Optional. The UNIQUE_FILE_NAMES parameter in the initialization file determines whether or not a unique ID is part of the destination file name.

(Continued on next page)



**Setting the destination file name**

Use this procedure to set the initialization file parameters that control the destination file name.

Step	Action
1	Open the data manager's initialization file and locate the Routing parameters subsection.
2	Do you want to use the mailbag ID in the destination file name? <ul style="list-style-type: none"> <li>▶ If YES, type 1 for the ADD_MBAG parameter and continue with Step 3. The default value is 1.</li> <li>▶ If NO, type 0 for the ADD_MBAG parameter and go to Step 4.</li> </ul>
3	Do you want to use the mailbag ID from the input file for the mailbagid portion of the destination file name? <ul style="list-style-type: none"> <li>▶ If YES, type 0 for the MBAG_CONSTRUCT parameter.</li> <li>▶ If NO, type 1 for the MBAG_CONSTRUCT parameter. This causes Gentrans:Server to generate a new mailbag ID for the destination file name. This is the default value.</li> </ul>
4	Do you want Gentrans:Server to append a unique ID to the destination file names? <ul style="list-style-type: none"> <li>▶ If YES, type 1 for the UNIQUE_FILE_NAMES parameter.</li> <li>▶ If NO, type 0 for the UNIQUE_FILE_NAMES parameter. This is the default value.</li> </ul>

(Continued on next page)

### Setting the outbound mail file name

If you have Gentrans:Server with Advanced Data Distribution, you can configure the way in which the outbound mail file name is determined.

Use this procedure to set the initialization file parameters that control the outbound mail file name.

Step	Action
1	Open the data manager's initialization file and locate the Routing parameters subsection.
2	Type a setting for MAIL_PREFIX_CONSTRUCT parameter to specify how you want the prefix for outbound mailbox file names constructed: <ul style="list-style-type: none"> <li>▶ 0 = Use the MAIL_PREFIX_NAME</li> <li>▶ 1 = Use the prefix that is on the input file name (for example, edi)</li> <li>▶ 2 = Use the destination file name</li> <li>▶ 3 = Use the name of the data manager personality type (for example, hcnd). This is the default value.</li> </ul>
3	Did you type 0 in Step 2? <ul style="list-style-type: none"> <li>▶ If YES, type the mail prefix name you want to use into the MAIL_PREFIX_NAME parameter. The default is udf.</li> <li>▶ If NO, skip this step.</li> </ul>

### Setting error file prefix

If you want a data manager to use the input file name prefix when routing a file with an error, set the USE\_FN\_PRE\_ON\_ERR parameter (Routing subsection) to 1.

The default for the USE\_FN\_PRE\_ON\_ERR parameter is 0. If the value is 0, the data manager will not use the input file name prefix for files in error. Instead, the HANDLER determines the prefix of the file routed in error.

# How to Set Error Notification

## Introduction

You can configure the Handlers section in the data manager’s initialization file to notify you or someone else in your organization whenever the data manager encounters a specific type of error. There are two types of notification:

- ▶ Dial a beeper
- ▶ Send a mail message.

## Beeper notification

When you set error notification to call a beeper, the data manager calls a Sterling Commerce script named **beeper**. The beeper script calls the script that you specify for the error handler.

## Dialing a beeper when an error occurs

Use this procedure to have the data manager call a beeper when an error is encountered.

Step	Action
1	Open the data manager’s initialization file and locate the Handlers section.
2	Type the name of the error handler in the handler_name position.  <b>Note</b> This is the format of the Handlers section. <pre>&lt;handler_name&gt; &lt;action&gt; &lt;file_name or script_name&gt;</pre> <b>Example</b> <pre>:q_err &lt;action&gt; &lt;file_name or script_name&gt;</pre>
3	Type <b>beep</b> in the action position.  <b>Example</b> <pre>:q_err <b>beep</b> &lt;file_name or script_name&gt;</pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>	
4	Type the command to invoke the beeper.	
	<b>IF you want to...</b>	<b>THEN...</b>
	Activate the beeper with a custom shell script	<p>Type the name of the file you want handled by the beeper script in the file_name position.</p> <p><b>Example</b> :q_err beep error/q_err</p> <p>On the line that follows the handler line, type the name of the shell script that contains the command to invoke the beeper system and the text message that you want appended to the file in the file_name position.</p> <p><b>Example</b> :q_err beep error/q_err beep joeuser "Queue error"</p> <p>In this example, "beep" is a customer shell script that activates the beeper system.</p>
	Have a communication package dial the beeper	<p>Type the name of the file you want the beeper script to handle in the file_name position.</p> <p>On the line that follows the handler line, type the communication line command.</p> <p><b>Example</b> :q_err beep script/beeper.num au9-555-3748,,,,,3214, 15 r1</p>
	<p><b>Comment</b> If you do not have a beeper script, use the model beeper script that Sterling Commerce provides. The model beeper script is in the <i>script</i> directory.</p> <p><b>Reference</b> See the <a href="#">Working with Scripts</a> chapter for information about modifying a script.</p> <p style="text-align: right;">(Continued on next page)</p>	

(Contd) Step	Action
5	<p>Exit the editor.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p> <p><b>CAUTION</b> <b>When you change a data manager’s initialization file, Gentran:Server validates the new parameters and records the results in the data manager’s log file. The change is logged under a new process ID number. The log entry contains the initialization file parameters.</b></p>
6	<p>Modify the beeper step in the beeper script to either invoke the shell script file or submit the file to a communication package to dial the beeper.</p> <p><b>Examples</b> To invoke the shell script:</p> <pre>sh tmp.num</pre> <p>To submit the file to a communication package:</p> <pre>3780Plus -b 9600 -d /dev/tty001 -c abcnet/abcnet.cfg -j script/tmp.num</pre> <p><b>Reference</b> See the <a href="#">Working with Scripts</a> chapter in this guide for information about modifying scripts.</p>

**Sending a mail message when an error occurs**

Use this procedure to have the data manager send a mail message when an error is encountered.

Step	Action
1	Locate the Handlers section in the data manager’s initialization file.
2	<p>Type the name of the error handler in the handler_name position.</p> <p><b>Note</b> This is the format of the Handlers section.</p> <pre>&lt;handler_name&gt; &lt;action&gt; &lt;file_name or script_name&gt;</pre> <p style="text-align: right;"><b>(Continued on next page)</b></p>

<b>(Contd) Step</b>	<b>Action</b>
3	Type either mm or move_e_mm in the action position.  <b>Reference</b> For definitions of these actions, see the <a href="#">Initialization File Structure</a> topic in this chapter.
4	Type the message you want mailed on the line that follows the action.  <b>Example</b> This is an example of configuring an error handler to mail a message when the data manager encounters a duplicate error.  <pre>:dupl_a mm Duplicate transmission from customer class_a.</pre>
5	Save your changes.
6	Exit the editor.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.
7	Configure the data manager's UNIX mail script for the message.  <b>Reference</b> See the <a href="#">How to Use a UNIX Mail Script to Send Messages</a> topic in this chapter.

---

# How to Use a UNIX Mail Script to Send Messages

---

**Introduction**    You can use UNIX mail scripts with data managers to send messages based on the consequences of the data manager operations.

**Example**

Your organization may want a certain individual or group to be notified if a data manager fails. You can use UNIX mail scripts to do this.

---

**Mail script location**

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

---

**The default UNIX mail script**

Each data manager has its own UNIX mail script, named for the data manager.

If you follow the procedures for editing a data manager's mail script and one does not exist for the data manager, Gentran:Server copies the **default** UNIX mail script and names it for the data manager.

---

(Continued on next page)

### Using environment variables in the mail script

You can use environment variables in the mail script to include detailed information in the mail message. The mail message will contain the actual value in place of the variable.

This table describes the mail message environment variables you can use in constructing the mail message in the mail script.

Environment Variable	Description
TP_CODE	Trading Partnership code
SRC_FN	Source file name
DEST_FN	Destination file name (set to <dir>/<filename>)
DATE	Date that the file was processed
TIME	Time that the file was processed
DM_NAME	Data manager's name
PERSONALITY	Data manager's personality type
ERROR_TYPE	Type of error

### CAUTION

**The availability of an environment variable depends upon the type of data manager and the type of error. You can not use all environment variables in all situations.**

### Example

This example shows the use of environment variables in a UNIX mail script.

```
echo "The data manager" $DM_NAME "produced the error" $ERROR_TYPE >
xltr.tmp

echo "at" $TIME "on" $DATE>>xltr.tmp

echo "Source file was" $SRC_FN>>xltr.tmp

echo "Dest file is" $DEST_FN>>xltr.tmp

cat xltr.tmp lrmmail $LOGNAME

rm -f xltr.tmp
```

(Continued on next page)



**Setting environment variables in the translation script**

If you call the UNIX mail script from a translation script, set the environment variables in the translation script, before calling the UNIX mail script. You can set the environment variables in the Steps section or in the Environment section.

**Example**

This is an example of using the Environment section to set environment variables.

```
#These values may be used by error handling scripts

SRC_FN=$XL_INFNAME

DEST_FN=$XL_OUTFNAME

TP_CODE='echo $XL_INFNAME|cut -d -f2`

DATE='date +%Y%m%d`

TIME='date +%H%M%S`

ERROR_TYPE="xltr_script"
```

**Creating or editing a UNIX mail script**

Use this procedure to create or edit a UNIX mail script for a data manager.

Step	Action
1	Configure the data manager's initialization file for sending a mail message when the error occurs.  <b>Reference</b> See <a href="#">How to Set Error Notification</a> .
2	Select <b>DataMgr</b> from the host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.  <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>Data Manager Control Name A Status T Description Emgr A 1239 F Foreground Manager (IPC Control) alnm n ***** l Async Line Manager ap00 n ***** m Flow: Test Q! flow Source Agent ap01 n ***** m Flow: 'nother fifteen Translate Agent appm n ***** m Application Data Manager appt n ***** x Application Translator Data Manager arch n ***** a -Darch -Aarch -d0 base n ***** u Base Manager Model -----</pre> </div> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	Select the data manager that you want to associate with a UNIX mail script.
4	<p>Press F4 to edit the mail script.</p> <p><b>System Response</b>            Gentran:Server calls the editor set in the \$EDITOR variable. If an editor is not set in this variable, the system uses the vi editor. The editor displays the UNIX mail script. If the data manager does not already have its own UNIX mail script, the system copies the default UNIX mail script and renames it for the data manager. From that point on, the data manager uses its own UNIX mail script.</p> <p><b>Reference</b>            For details about how to use the vi editor, see your UNIX reference books.</p>
5	<p>Make the changes.</p> <p><b>Example</b></p> <pre>echo "The data manager" \$DM_NAME "produced the error" \$ERROR_TYPE &gt; xltr.tmp  echo "at" \$TIME "on" \$DATE&gt;&gt;xltr.tmp  echo "Source file was" \$SRC_FN&gt;&gt;xltr.tmp  echo "Dest file is" \$DEST_FN&gt;&gt;xltr.tmp  cat xltr.tmp lrmail \$LOGNAME  rm -f xltr.tmp</pre> <p><b>Comment</b>            In the example, <i>xltr</i> is the data manager.</p>
6	Save the script and then exit the editor.
7	<p>Type the message into the data manager's initialization file.</p> <p><b>Reference</b>            For information and instructions, see the <a href="#">Maintaining Data Manager Initialization Files</a> section in this chapter.</p>

## How to Use an Error Class

**Introduction** An **error class** enables you to have a data manager handle the same type of error differently for different trading partners or file names. The error class is linked to a configuration record.

**When to use** Because the error class is linked to a configuration record, the data manager must be able to locate the correct configuration record to determine how to handle the error.

A data manager detects some types of errors, such as structural errors, before it checks the configuration record. For this reason, you cannot use error classes for certain types of errors.

You can use an error class for any type of error that can occur after the data manager has found the configuration record that tells it how to handle the data.

**Task table** This table describes the tasks you perform to use an error class.

Task	Action
1	Set up an error handler for each way that you want a particular error handled.
2	Assign a different error class to each handler combination for the error.
3	Assign the same class to all other error type and handler combinations that apply to the Trading Partnership code or file name.
4	Type the class into the Error Handling Class field of the Data Manager Configuration screen when you create a configuration record. The class defines how you want Gentran:Server to handle errors for the Trading Partnership code or file name.

(Continued on next page)

**Example** For Trading Partner A10, you want to receive a mail message when the data manager encounters a duplicate error (DUPLICATE). For all other trading partners, routing the data in error to the dupes directory is sufficient.

Task	Action
1	In the Parameters section of the data manager's initialization file, set ALLOW_DUPLICATES to 0.
2	<p>In the Errors section, type CLA10 as the class for the handler for trading partner A10.</p> <p>Type default as the class for the handler that you want to apply to all other trading partners.</p> <p><b>Example</b></p> <pre>:DUPLICATE dup_CLA10 CLA10 :DUPLICATE dup_def default</pre>
3	<p>Set up an error handler for a duplicate error, using move_e_mm as the action; then set up another error handler for a duplicate error, using move_e as the action.</p> <p><b>Example</b></p> <pre># Duplicate errors :dup_def move_e dupes/dup :dup_CLA10 move_e_mm dupes/dup  Duplicate error from customer A10</pre>
4	In the configuration record for Trading Partner A10, type CLA10 in the Error Handling Class field. For all other trading partners, type default into this field.

# How to Control Concurrent Processing With File Locking

---

**Introduction** You can configure data managers' initialization files to lock access to files in their scan directories until external processing is complete. This ensures that data managers do not process a file while another process is writing to the file.

---

**When to use** Use this procedure when you want to ensure that a data manager has exclusive access to a file it is processing.

If you experience long delays in transferring files between computers, you can use this procedure to set the STAT\_CHECK and LOCK\_ATTEMPTS parameters so that the data manager tolerates these longer delays.

By tuning the SCAN\_DELAY, STAT\_CHECK, and LOCK\_ATTEMPT parameters, you can set a data manager to have a long sleep cycle, but still react quickly to files when the data manager is awake.

---

## Configuring a data manager to lock files

Use this procedure to configure a data manager to lock files.

Step	Action
1	Open the data manager's initialization file.  <b>Reference</b> See <a href="#">How to Access and Edit an Initialization File</a> for instructions.  (Continued on next page)

<b>(Contd) Step</b>	<b>Action</b>
2	<p>Set these parameters:</p> <ul style="list-style-type: none"> <li>▶ STAT_CHECK to the frequency (in number of seconds) with which you want the data manager to check the input file's modification time</li> <li>▶ LOCK_ATTEMPT to the number of times an input file can pass the STAT_CHECK test but fail the file locking attempt. Must be greater than 0.</li> </ul> <p>When the LOCK_ATTEMPT limit is reached, the data manager terminates and moves the input file based on instructions for the E_LOCK_FAIL handler. Passing the STAT_CHECK test means the modification time has not changed.</p> <ul style="list-style-type: none"> <li>▶ SCAN_DELAY to the frequency (in number of seconds) with which the data manager checks the queue or directory for files to process or waits to be notified that a file has arrived</li> </ul> <p><b>Comments</b></p> <ul style="list-style-type: none"> <li>▶ STAT_CHECK sets the frequency with which the data manager checks a file's modification time. Setting this parameter to a higher value increases the data manager's toleration time. If you do not set STAT_CHECK, the data manager uses the value set in the SCAN_DELAY parameter for STAT_CHECK.</li> <li>▶ If LOCK_ATTEMPT is set to 0, the data manager does not try to lock the input file.</li> <li>▶ LOCK_ATTEMPT is the maximum number of attempts the data manager makes to lock a file before the data manager terminates. Again, a higher value increases the data manager's ability to tolerate delays in file transfers.</li> </ul>
3	Save your changes.
4	<p>Exit the editor.</p> <p><b>System Response</b> Gentran:Server redisplayes the Data Manager Control screen.</p>

# The File Locking Process

**Introduction** File locking is an important way to ensure that files are handled by only one data managers or other process at a time.

**Process** This table describes the stages in the file locking process.

Stage	Description	
1	Depending on the value in the data manager's NOTIFY parameter, the data manager: <ul style="list-style-type: none"> <li>▶ Is notified that a file has arrived in its work directory or queue or</li> <li>▶ Detects a file in its work directory or queue during a periodic scan.</li> </ul>	
2	If the data manager's LOCK_ATTEMPT parameter is greater than 0, the data manager checks the file's modification time to see whether the file is being updated by another process. (Updating includes transferring the file into the data manager's work directory or queue.)	
	<b>IF the modification time has...</b>	<b>THEN...</b>
	Changed since the last time the data manager checked	The data manager leaves the file alone until the next check and resets the number of lock attempts to zero.  Repeat Stages 1 and 2.
Not changed since the last time the data manager checked	Continue with Stage 3. (Continued on next page)	

<b>(Contd) Stage</b>	<b>Description</b>	
3	The data manager attempts to lock and process the file.	
	<b>IF the file...</b>	<b>THEN the data manager...</b>
	Is not locked by another process and has read/write permissions	Locks and processes the file.
	Is already locked by another process	Increments the number of lock attempts by 1 because the lock attempt has failed.
Does not have read/write permissions	Continue with Stage 4.	
<b>Note</b> The process that puts the file into the work directory sets the file permissions.		
4	The data manager compares the number of lock attempts on the file to the value specified in the LOCK_ATTEMPT parameter of its initialization file.	
	<b>IF the number of lock attempts...</b>	<b>THEN the data manager...</b>
	Exceeds the value in LOCK_ATTEMPTS	Terminates and follows the instructions in the E_LOCK_FAIL handler, which may be to send a mail message or start a script.
	Does not exceed the value in LOCK_ATTEMPTS	Waits for the number of seconds specified in the STAT_CHECK parameter to recheck the file's modification time.  Begins again with Stage 2.



# Configuring for Real-time Processing

## Overview

---

**Introduction** You can configure data managers to run only when data to be processed arrives in their scan directories or queues. This real-time processing mechanism is referred to as the **Inter Process Control (IPC) Trigger**.

---

**Purpose** The purpose of the IPC Trigger is to speed processing time.

---

### Upstream and downstream data managers

To enable a data manager to run only when data arrives in its work directory or queue, you must set parameters in the configuration files of both the upstream and the downstream data managers.

The **upstream data manager** is the starting agent. It routes the data to the downstream data manager's work directory or queue and then notifies the downstream data manager that data has arrived. Any data manager personality type except a line manager can be an upstream data manager. (Line managers do not have the ability to notify other data managers.)

The **downstream data manager** is the agent that runs only when the upstream data manager notifies it that data is in its work directory or queue.

---

### Using the FTP daemon to notify a data manager

Another way to speed processing time is to configure your FTP daemon to notify a data manager that it has moved data into the data manager's work directory or queue.

---

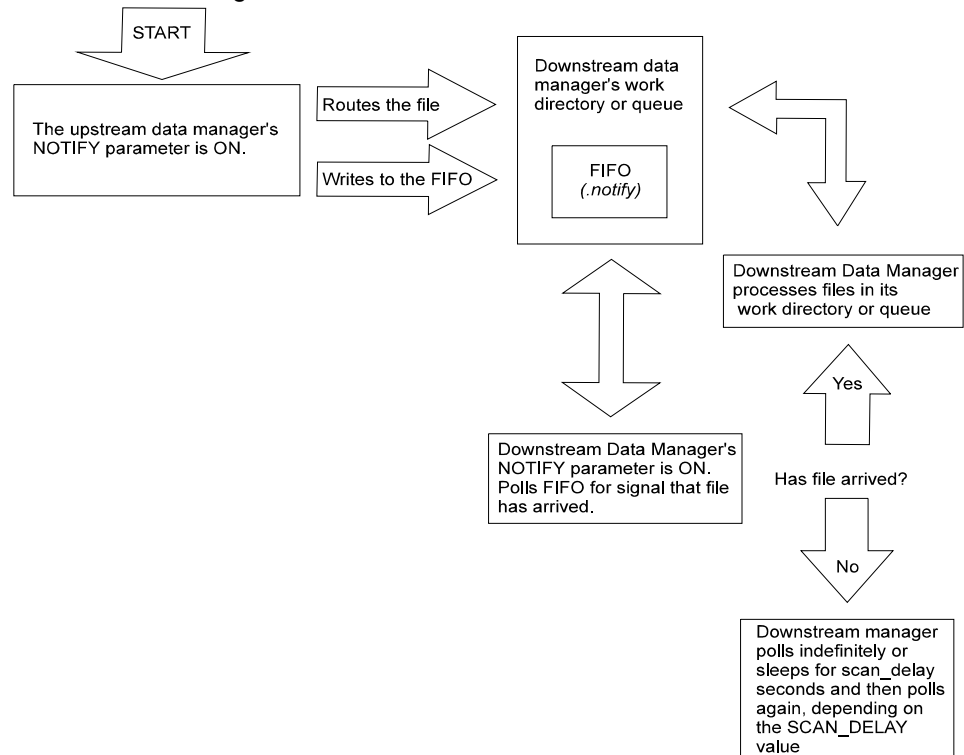
## The Stages of Real-Time Processing

**Introduction**    When you turn on the NOTIFY parameters in the upstream and downstream data managers' initialization files, the data managers communicate through a named pipe referred to as a FIFO.

**Definition of FIFO**    A **FIFO** is a first-in-first-out file that can be read only once before the data in it is removed.

**Processing priority**    More than one data manager can read from or write to the work directory or queue at the same time. The file's processing priority (set in the Data Manager Configuration screen) affects the order in which the file is processed.

**Process illustration**    This illustration shows the how the upstream and downstream data managers communicate through a FIFO.



(Continued on next page)

**Process stages**

This table describes the stages in real-time processing.

Stage	Description	
1	The upstream data manager checks the NOTIFY parameter in its initialization file. If NOTIFY is set to on (1), the upstream data manager routes the file to the destination directory or queue.  <b>Comment</b> The destination directory is the downstream data manager’s work queue.	
2	The upstream data manager attempts to open the FIFO.	
	<b>IF...</b>	<b>THEN...</b>
	The downstream data manager is polling the FIFO	The upstream data manager succeeds in opening the file and writes a NULL character to the FIFO.  <b>Comment</b> This process lets the downstream data manager know that a file is ready to be processed.
The FIFO does not exist or a downstream data manager is not polling the FIFO	The upstream data manager does not open or write to the FIFO.	
3	The downstream data manager attempts to poll the FIFO, which means that it waits for an event to happen.  (Continued on next page)	

<b>(Contd) Stage</b>	<b>Description</b>	
4	<p>If the FIFO does not exist, the downstream data manager creates it with prw-rw-r permissions. (The p refers to the file pipe attribute.)</p> <p><b>Comment</b> The FIFO file name and directory depend on whether the data manager is reading from a directory or queue.</p>	
	<b>IF the downstream data manager is reading from a...</b>	<b>THEN it creates the FIFO...</b>
	Queue	And names it <i>.notify_RESOURCE_GRP</i> , where RESOURCE_GRP is the resource group defined in the downstream data manager's initialization file.
	Directory	In the work directory and names it <i>.notify</i> .
5	<p>Once the FIFO exists, the downstream data manager polls the FIFO for the number of seconds specified in the SCAN_DELAY parameter.</p>	
	<b>IF...</b>	<b>THEN the downstream data manager...</b>
	SCAN_DELAY is -1	Polls the FIFO indefinitely, until signaled that the upstream data manager has written to the FIFO.
	An upstream data manager opens and writes to the FIFO	Stops polling and starts processing the file that the upstream data manager has written to the downstream data manager's work directory or queue.
	An upstream data manager has not opened and written to the FIFO	Wakes up after the number of seconds specified in its SCAN_DELAY parameter, starts processing files in its work directory or queue, and then returns to polling.

(Continued on next page)

**Files moved by FTP processes**

This table describes the process when an FTP daemon moves a file into a data manager's work directory.

Stage	Description	
1	An FTP process moves the file into a data manager's work directory.	
2	<p>After successful completion of the <b>put</b> command, the FTP daemon issues an <b>append</b> command to the FIFO to trigger processing.</p> <p><b>Command format</b> The <b>append</b> command format is <i>local_file.notify_file</i>.</p> <p><b>Comment</b> The local file should be between 1 and 1000 bytes in size and should contain only alphanumeric characters. If you want to set this manually, you can use a third-party FTP utility.</p>	
3	The next action depends on whether or not a data manager is polling the FIFO when the FTP daemon sends the <b>append</b> command.	
	<b>IF the data manager...</b>	<b>THEN...</b>
	Is polling the FIFO when the <b>append</b> command is sent	<p>The data manager opens the <i>.notify</i> file and returns control to the FTP.</p> <p><b>Comment</b> The <b>append</b> command returns a success code. The return of the FTP prompt signals return of control to the FTP process.</p> <p><b>Note</b> If the <b>append</b> command returns success, but there is not an FTP prompt, the FIFO may be blocked.</p>
Is not polling the FIFO when the <b>append</b> command is sent	The FIFO blocks, hanging the FTP daemon until the data manager opens the <i>&lt;data manager name&gt;.notify</i> file.	

# How to Configure Data Managers for Real-Time Processing

---

**Introduction**    This topic explains how to use the IPC Trigger mechanism to configure data managers for real-time processing.

---

**When to use**    Use this configuration procedure when you want faster processing time. The function enables your system to move critical documents through the processing cycle as quickly as possible.

---

**When not to use**    Do not use the IPC Trigger:

- With a line manager
- When you are using batch processing
- If you are not using a scan delay in processing.

---

**Procedure**    Use this procedure to enable real-time processing by turning on the IPC Trigger for the upstream and downstream data managers.

Step	Action
1	Open the upstream data manager's initialization file.  <b>Reference</b> See the <a href="#">How to Access and Edit an Initialization File</a> topic in this chapter for instructions.
2	Locate the Parameters section and set the NOTIFY field to 1.
3	Exit the editor.  <b>System Response</b> Gentran:Server redisplay the Data Manager Control screen.
4	Open the downstream data manager's initialization file.  <span style="color: red;">(Continued on next page)</span>

<b>(Contd) Step</b>	<b>Action</b>
5	<p>Set these parameters:</p> <ul style="list-style-type: none"> <li>▶ NOTIFY to 1</li> <li>▶ STAT_CHECK to 0</li> <li>▶ SCAN_DELAY to either -1 or a value greater than 0.</li> </ul> <p><b>Comments</b></p> <p>Setting STAT_CHECK to a value higher than 0 delays processing because the file has to be at least STAT_CHECK seconds old before the downstream data manager will process it.</p> <p>Setting SCAN_DELAY to -1 causes the downstream data manager to poll indefinitely until an upstream data manager opens the FIFO. When this happens, the data manager processes the files in its work directory or queue, and then starts polling again.</p> <p>Setting SCAN_DELAY to greater than 0 causes the downstream data manager to poll for the number of seconds set in SCAN_DELAY. When the number of seconds is up, the data manager processes any files in its scan directory or queue, and then starts polling again.</p>
6	<p>Exit the editor.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p>

# How to Configure the FTP Daemon to Notify a Data Manager

---

**Introduction** You can reduce processing time by having your FTP daemon notify a data manager that it has moved data into the data manager's work directory or queue.

---

**Procedure** Use this procedure to configure the FTP daemon to notify a data manager.

Step	Action
1	Open the ftpaccess file in an editor.
2	Locate the mbx_qprefix line and set the <b>ipc_flag</b> field to <b>y</b> .
3	Save the file.
4	Exit the editor.

**WARNING**

To avoid permission conflicts, the FTP daemon must be part of the Server group. The FTP daemon account in /etc/passwd must be the same group as the owner of the script manager (smgr) process. See your UNIX system administrator for assistance.

---



---

# Working with Configuration Records

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	3
	▶ Configuration Records .....	5
	▶ Ways to Create Configuration Records .....	6
	<b>Creating Records Manually</b>	
	▶ The Data Manager Configuration Screen .....	8
	▶ How to Manually Create a Configuration Record .....	13
	▶ How to Open a Configuration Record .....	15
	▶ How to Copy a Configuration Record .....	17
	<b>Using Patterns</b>	
	▶ Introduction .....	19
	▶ Token List .....	22
	▶ Categories in Patterns .....	24
	▶ Data Manager Pattern Configuration Screen .....	25
	▶ The Pattern and Meld Tasks .....	30
	▶ How to Create a New Pattern .....	31
	▶ How to Copy a Pattern .....	37
	▶ How to Add Another Data Manager to a Pattern .....	45
	▶ How to Add a New Pattern Configuration Record .....	47
	▶ How to Copy a Pattern Configuration Record In a Pattern .....	51
	▶ How to Edit a Pattern Configuration Record .....	54
	▶ How to Delete a Pattern .....	56
	▶ How to Use the Meld Function .....	57
	<b>Using the Inbound Data Manager</b>	
	▶ How the Inbound Data Manager Creates Records .....	61
	▶ How to Have the Inbound Data Manager Create Records .....	63
	<b>Maintaining Configuration Records</b>	
	▶ How to Verify Creation of Configuration Records .....	66
	▶ How to Edit a Configuration Record .....	67

- ▶ How to Delete a Single Configuration Record ..... 69
  - ▶ How to Delete a Range of Configuration Records ..... 70
-

# Overview

## Introduction

**In this chapter** This chapter explains how to create and maintain configuration records.

**Key terms** This table lists the key terms used in this chapter.

Term	Description
category	A class or grouping of Trading Partnership records.
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: <ul style="list-style-type: none"> <li>▶ Routing data</li> <li>▶ Invoking scripts</li> <li>▶ Archiving data</li> <li>▶ Handling data errors.</li> </ul>
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: <ul style="list-style-type: none"> <li>▶ Specifies the Trading Partnership code or file name that the data manager is to use to identify data</li> <li>▶ Tells the data manager what to do with the data it has identified.</li> </ul>
intelligent agent	An event-driven computer program that can operate without interaction from a person at a computer terminal.
meld	The process of merging a pattern with Trading Partnership records to produce configuration records.
pattern	One or a series of generic configuration records that describe the flow of data in a process.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
pattern configuration record	A generic configuration record used in the meld process to generate actual configuration records from selected Trading Partnership codes or file names.
token	A generic place holder in a pattern file. During the meld process, Gentran:Server replaces the tokens in the pattern with specific values from the Trading Partnership records and other Gentran:Server sources. This creates a configuration record for each Trading Partnership you meld with the pattern.
Trading Partner record	One of the records maintained in trading partner files: Trading Partnership record, Interchange Organization record, Group Organization record, and Contact record.
Trading Partnership code	A user-defined code that uniquely identifies a Trading Partnership record.
Trading Partnership	An arrangement with a specific trading partner to exchange information in a specific document type, using a particular standard version.

---

# Configuration Records

- 
- Definition** A configuration record describes how the data manager handles the data 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
  - Names the destination locations for the data
  - Specifies the name of the script (if any) the data manager runs
  - Indicates whether the data manager should archive the data.

---

**Why you need these records** When a data manager locates a file in its work directory or queue, it bases the way it handles the file on the Trading Partnership code or file name.

---

**How many records do you need?** You create a separate record for each file name or Trading Partnership code and modifier combination or file name that you expect the data manager to handle. If you have several trading partners or file names, you must create a configuration record for each one, even if the data flow is the same for each.

**Note**

For a file data manager, you can create a configuration record that routes files based only on the suffix of the file name. You do this by entering an asterisk (\*) as a wildcard character in the prefix position of the **File Name** field of the configuration record. The wildcard directs the file data manager to accept any value in the file name prefix. Thus, \*.xcl in the **File Name** field routes *nvcs.xcl*, *pske.xcl*, and *mbayh.xcl* and all other files with names that end in .xcl to the specified destination.

---

## Ways to Create Configuration Records

### Introduction

There are four basic ways to create configuration records:

- ▶ Use the Process Control Manager wizard
- ▶ Manually create the configuration records
- ▶ Enable an inbound data manager to create configuration records
- ▶ Use the pattern and meld feature to generate the records.

### CAUTION

**You cannot use EC Workbench to modify flow components you created with the PCM Wizard.**

### Selecting a method

The method you use to create configuration records depends on your circumstances. Use this decision table to determine how to create your configuration records.

WHEN you...	THEN...
Use the Process Control Manager wizard to create a process flow between two data managers	The Process Control Manager generates the configuration records.  <b>Reference</b> See the <a href="#">Creating a Flow with the PCM Wizard</a> chapter in this guide.
Need to create a few new records	Use the Data Manager Configuration screen to manually enter the configuration parameters for each Trading Partnership code and modifier or file name.  <b>Reference</b> See the <a href="#">How to Manually Create a Configuration Record</a> topic in this chapter.

(Continued on next page)

<b>(Contd)</b> <b>WHEN you...</b>	<b>THEN...</b>
Are not able to determine the identification for a Trading Partnership record before receiving data	Enable an inbound data manager to automatically create the records.  <b>Reference</b> See the <a href="#">How the Inbound Data Manager Creates Records</a> in this chapter.
Have a more complex flow that has more than two data managers in it	Use Gentran:Server's pattern and meld feature to generate the records.  <b>Reference</b> See the <a href="#">How to Use the Meld Function</a> topic in this chapter.

---

# Creating Records Manually

## The Data Manager Configuration Screen

**Introduction** One way to create a configuration record is to enter the record information on the Data Manager Configuration screen.

**Illustration** This illustration shows the Data Manager Configuration screen with a record displayed.

```

Data Manager Configuration
-----
Data Manager:
Name      edii
Type      i                Inbound Data Manager
Pattern   Inbound_Test

Trading Partner:
Code      INBND1911
Modifier

Description Inbound Sample Flow Test

Destination:
Directory xlti/
File Name INBND1911

Script Name
Desc

Archive Data (y/n) ? y
Error Handling Class default

Mailboxes
-----
InBox
OutBox-1
OutBox-2
OutBox-3

Queue Info
-----
Queue Name      xlti
Resource Group  xlti
Priority

F3:Delete F4:Copy F7:Next F8:Prev F9:Quit F10:Save

```

(Continued on next page)



## Fields and functions

This table describes the fields of the Data Manager Configuration screen and their functions.

Field	Function
Trading Partner Code	Specifies the code that uniquely identifies a Trading Partnership.
Modifier	<p>Defines the Trading Partnership code modifier, if any. The modifier is used to direct a record file layout for special processing or routing. You can enter any value that has meaning for your organization.</p> <p><b>Examples</b></p> <ul style="list-style-type: none"> <li>▶ N for normal</li> <li>▶ R for rush</li> </ul> <p><b>WARNING</b></p> <p><b>Use the modifier only with outbound (d type) data managers. If you define a modifier here, you must set the USE_MOD parameter to 1 in the data manager's initialization file. This makes the dnld data manager use the first 14 characters of the record as the Trading Partnership code and the 15th character as the modifier.</b></p>
File Name	<p>Defines the name of the input file. Used only with the file (type f) and translator (xltr) data managers. Can be used to separate data by type, application, trading partner, VAN, and so on.</p> <p><b>CAUTION</b></p> <p><b>For file (type f) or xltr (type x) data managers, the field File Name appears in place of the Trading Partner Code and Modifier fields.</b></p> <p><b>File data managers accept an asterisk (*) as a wildcard indicator in the prefix of the filename, (for example, *.mif). This enables you route all files with the same suffix to the same destination.</b></p>
Description	<p>Describes the data.</p> <p><b>Example</b> outbound X12</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Field</b>	<b>Function</b>
Destination Directory	Defines the path name of the destination directory for files processed under this configuration record.
Destination File Name	Defines the destination file name for the data.
Script Name	<p>Defines the name of the script to run.</p> <p><b>CAUTION</b> The <b>SCRIPT_RUN_SWITCH</b> in the data manager's initialization file controls when the script is run.</p> <p><b>Reference</b> See the <a href="#">Miscellaneous Parameters</a> topic in the <a href="#">Maintaining Initialization Files</a> chapter of this guide.</p>
Archive Data	<p>Controls data archiving.</p> <ul style="list-style-type: none"> <li>▶ Y enables archiving</li> <li>▶ N disables archiving</li> </ul>
Error Handling Class	<p>Specifies the error class, which is defined in the data manager's initialization file.</p> <p>Leave field blank for the default class.</p> <p><b>Reference</b> See the <a href="#">Maintaining Initialization Files</a> chapter of this guide for information about error handling classes.</p>
Mailboxes	<p>Defines the Mailbox identifiers for the inbox and up to three destinations (outboxes). There are two fields available for each destination.</p> <p><b>CAUTION</b> These fields are available only if you have the <b>Gentran:Server Advanced Data Distribution System</b>.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Field	Function
Type	<p>Indicates whether the Mailbox identifier is an individual mailbox or a mailbox distribution list:</p> <ul style="list-style-type: none"> <li>▶ M = Mailbox</li> <li>▶ L = Distribution List</li> </ul> <p><b>CAUTION</b> <b>This field is available only if you have the Gentran:Server Advanced Data Distribution System.</b></p>
Name	Specifies the name of the mailbox or distribution list.
Queue Name	Defines the name of the destination queue for files processed under this configuration record.
Resource Group	<p>Defines the name of the group to which the file belongs.</p> <p>Another data manager (a “read” data manager) uses this group name to identify the files it is to process in its queue. You set the group name that the read data manager looks for in the RESOURCE_GROUP parameter of its initialization file.</p>
Priority	Defines the processing priority for the file. Type a value from 0 through 9, where 0=highest priority; 9=lowest priority.

### Function keys

This table describes the function keys on the Data Manager Configuration screen.

Function Key	Function
F2	Displays a list of elements available for the selected field. Used for TP Code, Modifier, Script Name, Mailboxes, and Queue Name fields.
F5	<p>Displays a list of those elements that are already configured for the data manager. Used for the TP Code or File Name fields.</p> <p>When used in a Mailboxes field, displays a list of mailboxes or distribution list names.</p>

(Continued on next page)

<b>(Contd) Function Key</b>	<b>Function</b>
F6	<p>Displays a screen that contains fields you can use to enter search criteria for the Trading Partnership record. After you enter the search criteria, Gentran:Server displays a list of Trading Partnership codes that match the criteria.</p> <p><b>Reference</b> See the <a href="#">How to Search For a Trading Partnership Record</a> topic in the <a href="#">Understanding the Basics</a> chapter of this guide.</p> <p>When used in a Mailboxes field, displays the Mailbox search screen.</p> <p><b>Reference</b> If you have the optional Gentran:Server Advanced Data Distribution System, see the documentation for instructions on conducting a search for a mailbox.</p>
F7	Displays the next configuration record for the data manager.
F8	Displays the previous configuration record for the data manager.
F9	Leaves the Data Manager Configuration screen without saving the record.
F10	Saves the configuration record.

---

# How to Manually Create a Configuration Record

**Introduction** This topic explains how to create a configuration record by entering the record information on the Data Manager Configuration screen.

**Procedure** Use this procedure to create a configuration record.

Step	Action
1	<p>Select <b>DataMgr</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p> <pre data-bbox="649 898 1399 1108"> Data Manager Control ----- Name A      Status  T Description ----- fmgr: A      1239  F Foreground Manager (IPC Control) alnm n ***** l Async Line Manager ap00 n ***** m Flow: Test Q! flow Source Agent ap01 n ***** m Flow: 'nother fifteen Translate Agent appm n ***** m Application Data Manager amnt n ***** v Application Translator Data Manager           </pre>
2	<p>Select the data manager you want to configure; then press F5 to create a configuration record for the data manager.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.</p> <pre data-bbox="649 1318 1399 1791"> ----- Data Manager Configuration ----- Data Manager: Name      edii Type      i           Inbound Data Manager Pattern  Trading Partner: Code      [REDACTED] Modifier Description  Destination: Directory File Name  Script Name Desc  Archive Data (y/n) ? n Error Handling Class  Mailboxes ----- InBox OutBox-1 OutBox-2 OutBox-3  Queue Info ----- Queue Name Resource Group Priority  F2:Select F5:Find F6:TP Search F7:Next F8:Prev F9:Quit           </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>											
3	<p>Complete the Trading Partner Code field and Modifier field (if any). If this configuration record is for a file or translator data manager, the File Name field appears in place of the Trading Partner Code and Modifier fields.</p> <div data-bbox="630 527 1382 604" style="border: 1px solid black; padding: 5px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Trading Partner:</td> <td style="width: 40%;"></td> </tr> <tr> <td>Code</td> <td>INBMD1911</td> </tr> <tr> <td>Modifier</td> <td></td> </tr> </table> <table style="width: 40%; border-collapse: collapse; margin-left: 20px;"> <tr> <td style="text-align: right;">Mailboxes</td> <td style="border: 1px solid black; padding: 2px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">InBox</td> </tr> <tr> <td style="padding: 2px;">OutBox-1</td> </tr> <tr> <td style="padding: 2px;">OutBox-2</td> </tr> </table> </td> </tr> </table> </div> <p><b>Note</b> For a file data manager, the File Name field accepts an asterisk (*) as a wildcard indicator in the prefix position. When you use a wildcard, the file data manager accepts any values in the file prefix. This enables you to route files based on only the suffix.</p> <p><b>Example</b> File Name *.rpt routes all files with an .rpt suffix to the specified destination.</p>	Trading Partner:		Code	INBMD1911	Modifier		Mailboxes	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">InBox</td> </tr> <tr> <td style="padding: 2px;">OutBox-1</td> </tr> <tr> <td style="padding: 2px;">OutBox-2</td> </tr> </table>	InBox	OutBox-1	OutBox-2
Trading Partner:												
Code	INBMD1911											
Modifier												
Mailboxes	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">InBox</td> </tr> <tr> <td style="padding: 2px;">OutBox-1</td> </tr> <tr> <td style="padding: 2px;">OutBox-2</td> </tr> </table>	InBox	OutBox-1	OutBox-2								
InBox												
OutBox-1												
OutBox-2												
4	<p>Press the TAB key.</p> <p><b>System Response</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th data-bbox="613 1031 1019 1087">IF...</th> <th data-bbox="1019 1031 1417 1087">THEN...</th> </tr> </thead> <tbody> <tr> <td data-bbox="613 1087 1019 1255">A configuration record with the Trading Partnership code and modifier or file name already exists</td> <td data-bbox="1019 1087 1417 1255">Gentran:Server displays the configuration record. Press F9 to exit the screen and start over.</td> </tr> <tr> <td data-bbox="613 1255 1019 1413">A record with the Trading Partnership code and modifier or file name does not exist</td> <td data-bbox="1019 1255 1417 1413">Gentran:Server displays a confirmation prompt for a new configuration record. Continue with Step 5.</td> </tr> </tbody> </table>	IF...	THEN...	A configuration record with the Trading Partnership code and modifier or file name already exists	Gentran:Server displays the configuration record. Press F9 to exit the screen and start over.	A record with the Trading Partnership code and modifier or file name does not exist	Gentran:Server displays a confirmation prompt for a new configuration record. Continue with Step 5.					
IF...	THEN...											
A configuration record with the Trading Partnership code and modifier or file name already exists	Gentran:Server displays the configuration record. Press F9 to exit the screen and start over.											
A record with the Trading Partnership code and modifier or file name does not exist	Gentran:Server displays a confirmation prompt for a new configuration record. Continue with Step 5.											
5	<p>Type <b>y</b> at the confirmation prompt to confirm the addition.</p> <div data-bbox="743 1486 1333 1570" style="border: 1px solid black; padding: 5px; text-align: center;"> <p>Confirmation</p> <p>Configuration Record Not Found</p> <p>Do You Wish To Add The Config Record (y/n) ? _</p> </div>											
6	<p>Complete the remaining fields.</p> <p><b>Reference</b> See <a href="#">The Data Manager Configuration Screen</a> topic in this chapter for field descriptions and a sample of a completed screen.</p>											
7	<p>Press F10 to save the record.</p>											

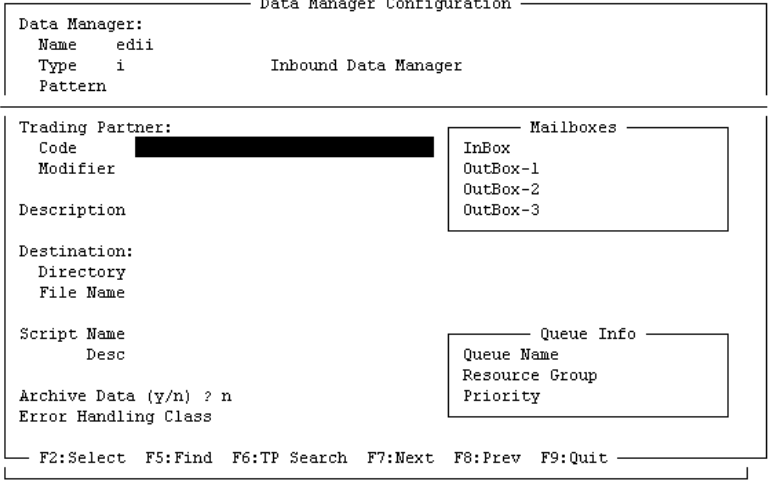
## How to Open a Configuration Record

**Introduction** To open a configuration record, you need to know one of these:

- ▶ The Trading Partnership code and modifier
- ▶ The file name for file or translator data managers.

**Opening a record** Use this procedure to open a configuration record.

Step	Action
1	<p>Select <b>DataMgr</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p> <pre data-bbox="649 966 1396 1155"> Data Manager Control Name A      Status  T Description ----- fmgr A      1239 F Foreground Manager (IPC Control) alrm n      ***** l Async Line Manager ap00 n      ***** m Flow: Test Q! flow Source Agent ap01 n      ***** m Flow: 'nother fifteen Translate Agent appm n      ***** m Application Data Manager           </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

(Contd) Step	Action
2	<p>Select the data manager of the entry you want to open; then press F5 to configure the data manager.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.</p> 
3	Select the Trading Partner Code or File Name field.
4	Press F5 to display a list of the records that are configured for the data manager; then select from this list.
5	<p>Does the configuration record you want to open have a modifier?</p> <ul style="list-style-type: none"> <li>▶ If YES, select the Modifier field, press F2 to display the available modifiers, and then select the modifier from this list. Continue with Step 5.</li> <li>▶ If NO, continue to Step 5.</li> </ul>
6	Press ENTER to open the configuration record.



# How to Copy a Configuration Record

**Introduction** The copy function duplicates the detail of a configuration record to create a new record.

**Copying a record** Use this procedure to copy a configuration record.

Step	Action
1	<p>Select <b>DataMgr</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p> <pre> Data Manager Control Name A      Status  T Description ----- Emgr A      1239 F Foreground Manager (IPC Control) alnm n      ***** l Async Line Manager ap00 n      ***** m Flow: Test Q! flow Source Agent ap01 n      ***** m Flow: 'nother fifteen Translate Agent appm n      ***** m Application Data Manager appt n      ***** x Application Translator Data Manager           </pre>
2	<p>Select the data manager of the entry you want to copy; then press F5 to configure the data manager.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.</p>
3	<p>Open the configuration record you want to copy.</p> <p><b>Reference</b> See <a href="#">How to Open a Configuration Record</a> for instructions.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
4	<p>Press F4 to copy the record.</p> <p><b>System Response</b> The system displays the <b>Copy</b> screen.</p> <div data-bbox="753 520 1297 684" style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre>Copy ----- New Trading Partner: Code      ████████████████████ Modifier ----- F9:Quit  F10:Save</pre> </div> <p><b>CAUTION</b> <b>For application configuration records, the screen shows File Name in place of the Trading Partner Code and Modifier fields.</b></p>
5	Type the new Trading Partnership code and modifier (if any) or the file name.
6	Press F10 to save the copy.

**Reference**

To edit the new record, see the [How to Edit a Configuration Record](#) topic in this chapter.

# Using Patterns

## Introduction

---

**Definition of pattern**

A pattern consists of one or more configuration record templates that describe the flow of data through a data manager in general terms. You can think of a pattern as a set of data flow models that you use when you generate the actual configuration records.

A pattern name is linked to:

- One or more data managers
- One or more Trading Partnership code or file name values.

---

**Definition of pattern configuration records**

We refer to the generic configuration record that links a pattern name, a data manager, and a Trading Partnership code or file name as a pattern configuration record.

---

**Tokens are used to create pattern configuration records**

To create a pattern configuration record, you name the pattern, add a data manager to it, and use a screen that looks almost exactly like the Data Manager Configuration screen to enter flow information.

When you create the pattern configuration record, instead of fixed values, you enter generic value place holders called tokens in the screen's fields. A token can represent:

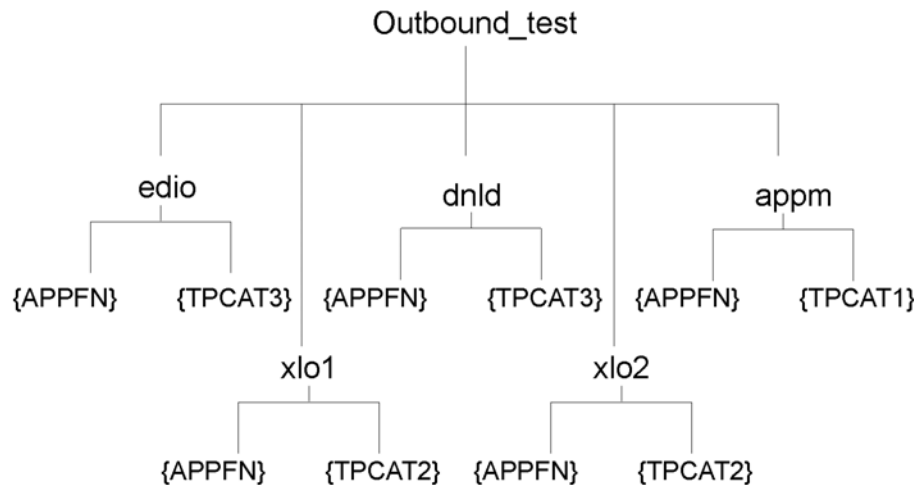
- The Trading Partnership code
- A category
- An environment variable
- A Trading Partnership description
- Other value.

---

(Continued on next page)

### Example of relationships in a pattern

This illustration shows the relationships in a pattern named `Outbound_test`. The data managers `edio`, `dnld`, `appm`, `xlo1`, and `xlo2` are added to the pattern. The file name tokens `{APPFN}`, `{TPCAT1}`, `{TPCAT2}`, and `{TPCAT3}` represent the generic file name and category configuration records for the data managers. The file name values are enclosed in braces to indicate that they symbolize a value, such as a Trading Partnership category.



### Example pattern configuration records

In the previous illustration, there are ten pattern configuration records:

- ▶ `Outbound_test+edio+{APPFN}`
- ▶ `Outbound_test+edio+{TPCAT3}`
- ▶ `Outbound_test+dnld+{APPFN}`
- ▶ `Outbound_test+dnld+{TPCAT3}`
- ▶ `Outbound_test+appm+{APPFN}`
- ▶ `Outbound_test+appm+{TPCAT1}`
- ▶ `Outbound_test+xlo1+{APPFN}`
- ▶ `Outbound_test+xlo1+{TPCAT2}`
- ▶ `Outbound_test+xlo2+{APPFN}`
- ▶ `Outbound_test+xlo2+{TPCAT2}`

(Continued on next page)

**Example**

This example shows tokens in the **Data Manager Pattern Configuration** screen for the ap00 data manager, which belong to the pattern named flow.

```

Data Manager Pattern Configuration
-----
Data Manager:
  Name      ap00                Flow: Test Q! flow Source Agent
  Type      m                  Application Data Manager
  Pattern   flow                Inbound Flow QA Test 1
-----
Trading Partner:
  Code      {TPCODE}
  Modifier
  Description {UIC_ID}
  Destination:
  Directory {TPCAT1}/
  File Name outvan
  Script Name {DOCTYP}
  Desc Token(s) inserted via the meld
  Archive Data (y/n) ? n
  Error Handling Class default
-----
Mailboxes
-----
InBox
OutBox-1
OutBox-2
OutBox-3
-----
Queue Info
-----
Queue Name
Resource Group
Priority
-----
F1:Token F2:Select F5:Name F6:Search F7:Next F8:Prev F9:Quit

```

**Reference**

See the [Token List](#) topic in this chapter for information about tokens.

### Melding generates configuration records

A pattern helps you create multiple configuration records quickly. After you create a pattern, you merge it with selected Trading Partnership records in a process called melding.

During the meld process, Gentran:Server replaces the tokens in the pattern configuration records with values from the Trading Partnership records and other Gentran:Server sources. This creates a configuration record for each Trading Partnership code or file name you meld with the pattern.

### Pattern or melding limitations

You can add a data manager to as many patterns as you want.

You can meld a Trading Partnership code or file name with one or more patterns.

Each data manager can have only one configuration record for a particular file name or Trading Partnership code and modifier combination.

## Token List

**Introduction** This topic lists and describes several common tokens.

**Reference**

See the readme files in the \$EDI\_ROOT/readme directory for descriptions of any new tokens that are not described here.

**Braces enclose tokens**

To enable Gentran:Server to recognize a token, the token must be enclosed in braces.

**Example**

This is an example of the Trading Partnership token enclosed in braces.

```
{TPCODE}
```

**Token list**

This table describes common tokens you can enter in the fields of a Data Manager Configuration Pattern screen.

Token	Description
{TPCODE}	Trading Partnership code.
{APPFN}	Application file name.
{UIC_ID}	Your interchange ID.
{PIC_ID}	Trading partner's interchange ID.
{UAP_ID}	Your group or application ID.
{PAP_ID}	Trading partner's group or application ID.
{STDVER}	Standard version.
{DOCTYP}	Transaction set or message ID.
{DESCR}	Trading partnership description.
{TPCAT1}	Category 1 value.
{TPCAT2}	Category 2 value.
{TPCAT3}	Category 3 value.

(Continued on next page)

(Contd) Token	Description
{TPCAT4}	Category 4 value.
{TPCAT5}	Category 5 value.
{ENV_VARIABLE}	Any environmental variable you have set up and globally exported. Omit the dollar sign (\$) when using an environment variable token.  <b>Note</b> This token is not available in the selection list.

### Using Trading Partnership categories

---

#### **WARNING**

**You must establish the Trading Partnership categories and their values before using the [TPCAT#] tokens.**

#### **Reference**

See the [How to Create Categories](#) topic in the [Working with Trading Partnerships](#) chapter of the *Gentran:Server for UNIX and Workstation Application Integration User's Guide*.

---

## Categories in Patterns

---

**Definition** A category is a way to group Trading Partnership records by type.

**Example**

You create a category called INDUSTRY so that you can group trading partners by industry. You then create values for different types of industries, such as retail, manufacturing or warehousing.

---

**Reason to use categories in patterns**

You can use categories to establish rules within patterns. For example, use the {TPCAT#} token that represents the INDUSTRY category as the destination directory. This causes the data manager to route the data based on the industry of the trading partner.

When you meld a pattern with a Trading Partnership record, Gentran:Server replaces the category (INDUSTRY) with the category value (retail, manufacturing, or warehousing) for the Trading Partnership.

**Example 1**

Create a category called APPLICATION so that you can group Trading Partnerships by the application that data is coming from or going to. You set values for different types of applications and then use these values to locate relevant records and to set rules within patterns. If APPLICATION is the second category, use the {TPCAT2} token in the pattern, such as in the Destination File Name field.

**Example 2**

Create a category called PATTERN NAME to represent the name of the pattern, such as inbound\_van or outbound\_mail. If PATTERN NAME is the fourth category, use the {TPCAT4} token in the pattern, such as in the Description field.

**Example 3**

Create a category called TRADING PARTNER'S VAN to group Trading Partnerships by the name of the trading partner's VAN, such as GEIS, Advantis, or Commerce:Network.

**Example 4**

Create a category called COMMUNICATIONS to group Trading Partnerships by their communications path (VAN, FTP, and so on).

---

**Reference**

See the [How to Create Categories](#) topic in the [Working with Trading Partnerships](#) chapter of the *Gentran:Server for UNIX and Workstation Application Integration User's Guide* for instructions.

---



# Data Manager Pattern Configuration Screen

**Introduction** When you create a pattern, you enter values and tokens on the Data Manager Pattern Configuration screen. This is the screen that creates the pattern configuration record. This record links the pattern name, the data manager, and the Trading Partnership code or file name to the generic flow information.

**Illustration** This illustration shows the Data Manager Pattern Configuration screen with a record displayed.

## CAUTION

**if you selected a type f (file) or xltr data manager, the Trading Partner Code and Modifier fields are replaced with File Name on this screen.**

```

Data Manager Pattern Configuration
-----
Data Manager:
Name      sp00          Flow: Test Q! flow Source Agent
Type      m            Application Data Manager
Pattern   flow         Indound Flow QA Test 1
-----
Trading Partner:
Code      {TPCODE}
Modifier
Description {UIC_ID}
-----
Destination:
Directory {TPCAT1}/
File Name outvan
-----
Script Name {DOCTYP}
Desc Token(s) inserted via the meld
-----
Archive Data (y/n) ? n
Error Handling Class default
-----
Mailboxes
-----
InBox
OutBox-1
OutBox-2
OutBox-3
-----
Queue Info
-----
Queue Name
Resource Group
Priority
-----
F1:Token F2:Select F5:Name F6:Search F7:Next F8:Prev F9:Quit

```

## Combining tokens with other values

In most fields, you can use tokens in combination with fixed values to gain a specific end result in your configuration records.

### Example 1

If your destination file name for inbound invoices is INV and the Trading Partnership code. Enter INV and the TPCODE token in the Destination File Name field.

```

Destination:
Directory: invoices
File Name INV{TPCODE}

```

(Continued on next page)

**Example 2**

If your script name for FTP routing is FTP followed by the Trading Partnership category for the FTP type, type **ftp** and the appropriate category token in the Script Name field.

```
Script Name: ftp{TPCAT2}
```

**Fields and functions**

This table describes the fields of the Data Manager Pattern Configuration screen and their functions.

Field	Function
Name	Displays the name and description of the data manager linked to the pattern.
Type	Displays the data manager's Type code. The Type code determines the field names on the configuration screen.
Pattern	Displays the name of the pattern and its description.
Trading Partner Code	For inbound flows, defines the Trading Partnership code. You can use a token.  <b>Example</b> {TPCODE}
Modifier	Defines the Trading Partnership code modifier, if any. The modifier is used to direct a record file layout for special processing or routing. You can enter any value that has meaning for your organization.  <b>Examples</b> <ul style="list-style-type: none"> <li>▶ N for normal</li> <li>▶ R for rush</li> </ul> <b>WARNING</b> <b>The modifier is used only with outbound (d type) data managers.</b>
File Name	Defines the file name for file and translator type data managers. You can use a token.  <b>Example</b> {APPFN}

(Continued on next page)

<b>(Contd) Field</b>	<b>Function</b>
Description	Describes the data. You can enter a token.  <b>Example</b> {DESCR}
Destination Directory	Defines the path name of the destination directory. You can use tokens.  <b>Example</b> move_edi/{TPCAT1}
Destination File Name	Defines the resulting file name for the data. You can use tokens.  <b>Example</b> EDI_{TPCODE}
Script Name	Defines the name of the script to be run. You can use tokens.  <b>Example</b> {TPCAT1}
Archive Data	Controls data archiving: <ul style="list-style-type: none"> <li>▶ Y enables archiving</li> <li>▶ N disables archiving</li> </ul>
Error Handling class	Specifies the error class, which is defined in the data manager's initialization file.
Mailboxes	If your company has the optional Advanced Data Distribution System, defines the Mailbox identifiers for the inbox and up to three destinations (outboxes). There are two fields for each destination.  <b>CAUTION</b> <b>This field is available only if you have the Gentran:Server Advanced Data Distribution System.</b>  <div style="text-align: right;">(Continued on next page)</div>

<b>(Contd) Field</b>	<b>Function</b>
Type	<p>Indicates whether the Mailbox identifier is an individual mailbox or a mailbox distribution list:</p> <ul style="list-style-type: none"> <li>▶ M = Mailbox</li> <li>▶ L = Distribution List</li> </ul> <p><b>CAUTION</b> This field is available only if you have the Gentran:Server Advanced Data Distribution System.</p>
Name	<p>Defines the name of the mailbox or list.</p> <p><b>CAUTION</b> This field is available only if you have the Gentran:Server Advanced Data Distribution System.</p>
Queue Name	<p>Defines the name of the destination queue.</p> <p><b>Reference</b> See the <a href="#">Using Queues</a> chapter in this guide for information about queues, resource groups, and processing priority.</p>
Resource Group	<p>Defines the name of the group to which the files this data manager processes belong.</p> <p>A downstream data manager that looks for its files in a queue uses this group name to identify the files it is to process. You set the group name that the downstream data manager looks for in the RESOURCE_GROUP parameter of its initialization file.</p>
Priority	<p>Defines the processing priority for the file. Type a value from 1 through 9, where 1=highest priority; 9=lowest priority.</p>

---

(Continued on next page)

**Function keys**

This table describes the function keys on the Data Manager Pattern Configuration screen.

Function Key	Function
F1	Displays a list of tokens for the field.
F2	Displays a list of elements available for the selected field. Used for the TP Code, Modifier, Script Name, Mailboxes, Queue Name, Resource Group, and Priority fields.
F3	Deletes the pattern configuration record.
F4	Copies the pattern configuration record.
F5	Displays a list of those elements that are already configured for the data manager and pattern. Used for the TP Code and File Name fields.
F5	In the Mailboxes fields, displays a list of mailboxes or distribution list names already configured for the data manager and pattern.
F6	Displays a search screen with fields you can use to enter search criteria for the Trading Partnership code. After you enter the search criteria, Gentran:Server displays a list of Trading Partnership codes that match the criteria.
F6	In the Mailboxes fields, displays a Mailbox search screen with fields you can use to enter search criteria for a mailbox identifier.
F7	Displays the next pattern configuration record configured for the data manager and pattern.
F8	Displays the previous pattern configuration record configured for the data manager and pattern.
F9	Leaves the Data Manager Configuration screen without saving the pattern configuration record.
F10	Saves the pattern configuration record.

## The Pattern and Meld Tasks

### Task summary

This table summarizes the tasks you must complete to generate configuration records with the pattern and meld operations.

Task	Description
1	Create the data managers that you want to use in the pattern. <b>Reference</b> See the <a href="#">Working with Data Managers</a> chapter in this guide.
2	Create the pattern. <b>Reference</b> See these topics in this chapter: <ul style="list-style-type: none"> <li>▶ <a href="#">How to Create a New Pattern</a></li> <li>▶ <a href="#">How to Copy a Pattern</a></li> </ul>
3	Add to the pattern the data managers you want to use. <b>Reference</b> See <a href="#">How to Add Another Data Manager to a Pattern</a> .
4	Add to the pattern any pattern configuration records you want to use. <b>Reference</b> See these topics in this chapter: <ul style="list-style-type: none"> <li>▶ <a href="#">How to Add a New Pattern Configuration Record</a></li> <li>▶ <a href="#">How to Copy a Pattern Configuration Record In a Pattern</a></li> </ul>
5	Meld the pattern with selected Trading Partnership codes or file names to generate the configuration records. <b>Reference</b> See <a href="#">How to Use the Meld Function</a> .
6	Verify that the process created the configuration records. <b>Reference</b> See the <a href="#">How to Verify Creation of Configuration Records</a> topic in this chapter.

# How to Create a New Pattern

---

**Introduction**

This topic describes how to:

- ▶ Name and describe a new pattern
- ▶ Add a data manager to the pattern
- ▶ Enter the tokens and other values into the pattern configuration record.

---

**Before you begin**

Before you begin this procedure, create the data managers that you want to associate with the pattern.

**Reference**

See the [Working with Data Managers](#) chapter in this guide for instructions.

---

**Pattern rules**

When you create patterns, follow these rules:

- ▶ You must add at least one data manager to the pattern
- ▶ You can add any number of data managers to a pattern
- ▶ You can add a specific data manager to more than one pattern.

---

**When to use**

Use this procedure when you want to create a pattern that you can use to generate configuration records.

---

**Log file entries**

When you save a pattern configuration record, Gentran:Server validates the new parameters and records the results in the data manager's log file. The system logs the event under the server process ID number. The log entry contains the data manager's initialization file parameters.

---

(Continued on next page)

**Procedure** Use this procedure to create a new pattern and a pattern configuration record.

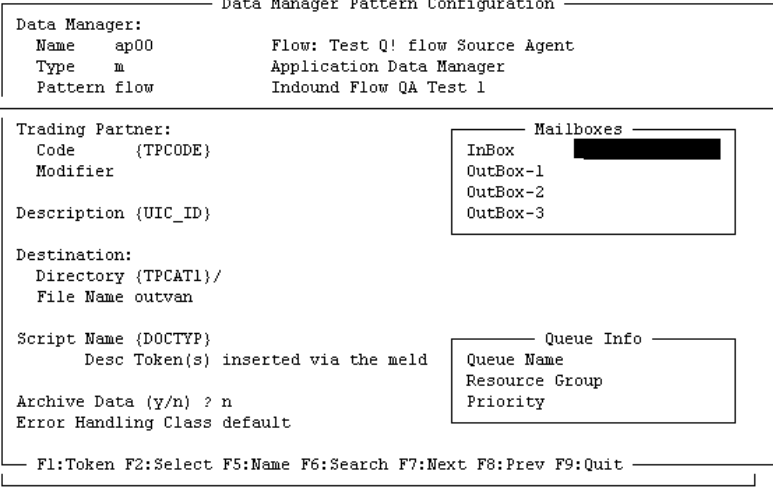
Step	Action
1	<p>Select <b>Patrn</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Patterns screen.</p> <pre data-bbox="651 638 1382 814"> Data Manager Patterns ----- Pattern Name: ██████████ Desc: F2:Select F9:Quit </pre>
2	Type the name of the new pattern in the Pattern Name field.
3	<p>Press ENTER.</p> <p><b>System Response</b> Gentran:Server displays a message to let you know the pattern name is not on file and asks if you want to add it.</p>
4	<p>Type <b>y</b> at the prompt.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Patterns screen with additional function keys.</p>
5	<p>Type a description of the new pattern in the Desc field.</p> <pre data-bbox="651 1310 1390 1499"> Data Manager Patterns ----- Pattern Name: flow Desc: Inbound Flow QA Test 1 F3:Del F4:Copy F5:Config F6:Meld F9:Quit F10:Save </pre> <p><b>Caution</b> If you press F10 at this point, Gentran:Server saves the new pattern name and description, but exits the screen.</p> <p style="text-align: right;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
6	<p>Press F5 to configure the pattern.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Selection screen. This screen is used to choose the data manager that you want to associate with the pattern. The cursor is in the Data Manager Name field.</p> <pre data-bbox="662 604 1382 835"> Data Manager Selection _____ Data Manager: Name ████████ Type - Desc F2:Select F5:Find F9:Quit F10:Save _____ </pre>
7	<p>Press F2 to display a list of all data managers; then select the name of the data manager you want to add to the pattern and press ENTER.</p> <p><b>Comment</b> To display a list of data managers that are already associated with this pattern, press F5.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Selection screen with values in the fields.</p> <pre data-bbox="675 1251 1369 1472"> Data Manager Selection _____ Data Manager: Name ap00 Type m-Application Data Manager Desc Flow: Test Q! flow Source Agent F2:Select F5:Find F9:Quit F10:Save _____ </pre> <p><b>WARNING</b> <b>You cannot change the Type or Desc fields on this screen.</b></p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
8	<p>Is this the data manager you want to add to the pattern?</p> <ul style="list-style-type: none"> <li>▶ If YES, press F10 to continue to the Data Manager Pattern Configuration screen.</li> <li>▶ If NO, press F9 to quit and return to the Data Manager Selection screen to select a different data manager.</li> </ul> <p><b>CAUTION</b></p> <p><b>On most platforms, you can select a different data manager at any time by pressing CTRL+D. This shortcut key displays the Data Manager Selection screen, enabling you to choose another data manager.</b></p> <pre> Data Manager Pattern Configuration ----- Data Manager: Name      ap00          Flow: Test Q! flow Source Agent Type      m             Application Data Manager Pattern   flow          Inbound Flow QA Test 1 -----  Trading Partner: Code      ██████████ Modifier Description  Destination: Directory File Name  Script Name Desc  Archive Data (y/n) ? n Error Handling Class  Mailboxes ----- InBox OutBox-1 OutBox-2 OutBox-3  Queue Info ----- Queue Name Resource Group Priority  F1:Token F2:Select F5:Find F6:TP Search F7:Next F8:Prev F9:Quit </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
9	<p>On the Data Manager Pattern Configuration screen, type a value in the Trading Partner Code or File Name field and then press TAB.</p> <p><b>CAUTION</b>  <b>To display the token list, press F1. Gentran:Server appends any token you select to any data you have entered into the field.</b></p> <p><b>Comment</b>  For inbound flows, you usually use the {TPCODE} token in the Trading Partner Code field. For outbound flows where the first data manager in the flow is an application data manager, you usually use the {TPCODE} token in the Trading Partner Code field. For application flows, you usually use the {APPFN} token (application file name) in the File Name field.</p> <p><b>System Response</b>  Gentran:Server displays a message to let you know that the pattern configuration does not exist and asks if you want to add it.</p> <div data-bbox="740 1003 1308 1087" style="border: 1px solid black; padding: 5px; text-align: center;"> <pre> Confirmation Configuration Record Not Found Do You Wish To Add The Config Record (y/n) ? _ </pre> </div>
10	<p>Type <b>y</b> at the prompt to add the configuration record.</p> <p><b>System Response</b>  Gentran:Server displays the Data Manager Pattern Configuration screen and selects the Description field.</p> <p style="text-align: right;"><b>(Continued on next page)</b></p>

(Contd) Step	Action
11	<p>Complete the remaining fields, inserting appropriate tokens and values.</p> <p><b>CAUTION</b></p> <p><b>Gentran:Server checks the Directory field to determine if the directory you entered exists. If it does not, Gentran:Server asks you if you want to add it. You can either add it now or during the meld process.</b></p> <p><b>Gentran:Server compares the value in the Script field to the script library list. If the script is not in the script list, Gentran:Server displays a warning message.</b></p> 
12	<p>Press F10 to save the pattern configuration record.</p> <p><b>System Response</b></p> <p>Gentran:Server saves the pattern configuration record and clears the fields on the Data Manager Pattern Configuration screen.</p>
13	<p>Do you want to add another pattern configuration for the data manager?</p> <ul style="list-style-type: none"> <li>▶ If YES, repeat Steps 9 through 12.</li> <li>▶ If NO, continue with Step 14.</li> </ul>
14	<p>Press F9 to exit the Data Manager Pattern Configuration screen.</p>
15	<p>Press F10 to save the pattern.</p>

## How to Copy a Pattern

**Introduction** Another way to create a new pattern is to copy an existing pattern and give it a new name. When you copy a pattern, Gentran:Server copies all the data managers and pattern configuration records of the copied pattern to the new pattern. You can then modify the pattern configurations and add any new data managers to the new pattern.

**Pattern rules** When you create patterns, follow these rules:

- You must add at least one data manager to the pattern
- You can add any number of data managers to a pattern
- You can add a specific data manager to more than one pattern.

**Procedure** Use this procedure to copy a pattern.

Step	Action
1	<p>Select <b>Pattn</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Patterns screen.</p> <pre> Data Manager Patterns ----- Pattern Name: ██████████ Desc: ----- F2:Select  F9:Quit           </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
2	<p>Press F2 to list the pattern names; then select the pattern name you want to copy and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the screen with additional function keys.</p> <pre data-bbox="651 611 1395 793"> Data Manager Patterns ----- Pattern Name: flow Desc: Inbound Flow QA Test 1 ----- F3:Del F4:Copy F5:Config F6:MeId F9:Quit F10:Save </pre>
3	<p>Press F4 to copy the pattern.</p> <p><b>System Response</b> Gentran:Server displays the Copy Data Manager Pattern screen.</p> <pre data-bbox="639 1062 1391 1205"> Copy Data Manager Pattern ----- New Pattern Name: Desc: ----- F9:Quit F10:Save </pre>
4	<p>Type the name of the new pattern and then press F10 to save the new pattern.</p> <p><b>System Response</b> The system briefly displays a message that shows the number of pattern configuration records copied.</p>
5	<p>Type the pattern description in the Desc field.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>	
6	Use this table to determine your next action.	
	<b>IF you want to...</b>	<b>THEN...</b>
	Save the pattern and its pattern configuration records without making any changes	Press F10 to save the pattern and then press F9 to exit the screen.
	Modify a pattern configuration	Press F5 to display the Data Manager Selection screen and then follow the procedure for Modifying a pattern configuration record in this topic.
	Add a new data manager and pattern configuration to the pattern	Press F5 to display the Data Manager Selection screen and then follow the procedure for Adding a new data manager and pattern configuration in this topic.

---

(Continued on next page)

### Modifying a pattern configuration record

Use this procedure to modify a pattern configuration record in the new pattern.

Step	Action
1	<p>On the Data Manager Selection screen, press F5 to display a list of data managers that were added to this pattern; then select the name of the data manager.</p> <p><b>Comment</b> If you want to select a data manager that has not been added to the pattern, press F2.</p> <p><b>System Response</b> Gentran:Server displays the screen with values in the fields.</p> <pre data-bbox="678 810 1370 1024"> Data Manager Selection _____ Data Manager: Name ap00 Type m-Application Data Manager Desc Flow: Test Q! flow Source Agent F2:Select F5:Find F9:Quit F10:Save _____ </pre>
2	Press F10 to continue to the Data Manager Pattern Configuration screen.
3	<p>Select the Trading Partner Code or File Name field; then press F5 to display a list of Trading Partnership code or file name values configured for the data manager.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
4	<p>From the list, select the pattern configuration you want to modify and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Pattern Configuration record.</p> <pre> Data Manager: Name      ap00          Flow: Test Q! flow Source Agent Type      m             Application Data Manager Pattern   flow          Indound Flow QA Test 1  Trading Partner: Code      {TPCODE} Modifier  Description {UIC_ID}  Destination: Directory {TPCAT1}/ File Name outvan  Script Name {DOCTYP} Desc      Token(s) inserted via the meld  Archive Data (y/n) ? n Error Handling Class default  Fl:Token F2:Select F5:Name F6:Search F7:Next F8:Prev F9:Quit </pre>
5	<p>Does the record you want to change have a Trading Partner Code modifier?</p> <ul style="list-style-type: none"> <li>▶ If YES, select the Modifier field; press F2 to display the available modifiers; then select the modifier from this list. Continue with Step 6.</li> <li>▶ If NO, continue with Step 6.</li> </ul>
6	<p>Modify the field values.</p> <p><b>Reference</b> See the <a href="#">Data Manager Pattern Configuration Screen</a> topic in this chapter for field descriptions.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
7	<p>Press F10 to save your changes.</p> <p><b>System Response</b> Gentran:Server clears the fields on the Data Manager Pattern Configuration record.</p>
8	<p>Do you want to modify another pattern configuration record for this data manager?</p> <ul style="list-style-type: none"> <li>▶ If YES, repeat Steps 3 through 6.</li> <li>▶ If NO, press F9 to exit the Data Manager Pattern Configuration screen and return to the Data Manager Patterns screen.</li> </ul>

### Adding a new data manager and pattern configuration record

Use this procedure to add a new data manager and pattern configuration record to the new pattern.

Step	Action
1	Press F2 to display a list of available data managers on the Data Manager Selection screen.
2	<p>Select the name of the data manager you want to add to the pattern and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the screen with values in the fields.</p> <pre style="border: 1px solid black; padding: 10px; margin: 10px 0;"> Data Manager Selection _____  Data Manager: Name ap00 Type m-Application Data Manager Desc Flow: Test Q! flow Source Agent  F2:Select F5:Find F9:Quit F10:Save _____ </pre>
3	Press F10 to continue to the Data Manager Pattern Configuration screen.

(Continued on next page)

(Contd) Step	Action
4	<p>Type a value in the Trading Partner Code or File Name field and then press the TAB key.</p> <p><b>CAUTION</b> To display the token list, press F1. Any token you select is appended to any data you have entered into the field.</p> <p><b>Comment</b> For inbound flows, you usually use the {TPCODE} token in the <b>Trading Partner Code</b> field. For outbound flows where the first data manager in the flow is an application data manager, you usually use the {TPCODE} token in the Trading Partner Code field. For other outbound flows, you usually use the {APPFN} token in the <b>File Name</b> field.</p> <p><b>System Response</b> Gentran:Server displays a message to let you know that the pattern configuration does not exist and asks if you want to add it.</p> <div data-bbox="717 1010 1330 1100" style="border: 1px solid black; padding: 5px; text-align: center;"> <pre> Confirmation Configuration Record Not Found Do You Wish To Add The Config Record (y/n) ? _ </pre> </div>
5	<p>Type <b>y</b> at the prompt.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Pattern Configuration screen and selects the Description field.</p>
6	<p>Complete the remaining fields, inserting appropriate tokens and values.</p> <p><b>CAUTION</b> Gentran:Server checks the <b>Directory</b> field to determine if the directory you entered exists. If it does not, Gentran:Server asks you if you want to add it. You can either add it now or during the meld process.</p> <p>Gentran:Server checks the <b>Script</b> field to determine if the script exists. If it does not exist, Gentran:Server displays a warning message.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
7	Press F10 to save the pattern configuration record.  <b>System Response</b> Gentran:Server saves the pattern configuration record and clears the fields on the Data Manager Pattern Configuration screen.
8	Do you want to add another pattern configuration record? <ul style="list-style-type: none"><li>▶ If YES, repeat Steps 3 through 6.</li><li>▶ If NO, press F9 to exit the Data Manager Pattern Configuration screen.</li></ul>

---

## How to Add Another Data Manager to a Pattern

**Introduction** Once you have created a pattern that describes a data flow, you can add to it other data managers that have the same flow.

**When to use** Add another data manager to the pattern when:

- ▶ The basic data manager flow is the same as those already associated with the pattern
- ▶ You want to use the pattern and meld features to generate configuration records for the data manager you added.

**Procedure** Use this procedure to add another data manager to a pattern.

Step	Action
1	Select <b>Patrn</b> from the host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Patterns screen.
2	Press F2 to list the pattern names; then select the pattern you want to use and press ENTER.  <b>System Response</b> Gentran:Server displays the screen with additional function keys.  <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> Data Manager Patterns _____ Pattern Name: flow Desc: Inbound Flow QA Test 1 _____ F3:Del F4:Copy F5:Config F6:Meld F9:Quit F10:Save _____           </pre> </div> <p style="text-align: right; color: red;">(Continued on next page)</p>

(Contd) Step	Action
3	<p>Press F5 to configure the pattern.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Selection screen, which enables you to choose the data manager you want to add to the pattern.</p> <pre data-bbox="673 611 1365 827"> Data Manager Selection _____ Data Manager: Name ████████ Type - Desc F2:Select F5:Find F9:Quit F10:Save _____ </pre>
4	<p>Press F2 to display a list of data managers; then select the name of the data manager you want to add to the pattern and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the screen with values in the fields.</p> <p><b>WARNING</b> <b>You cannot change the field values on this screen.</b></p>
5	<p>Press F10 to save the data manager addition and continue to the Data Manager Pattern Configuration screen.</p>
6	<p>Add a pattern configuration for this data manager by following Steps 9 through 12 in <a href="#">How to Create a New Pattern</a>.</p>

## How to Add a New Pattern Configuration Record

**Introduction** You can add a new pattern configuration record to a data manager that is linked to another pattern.

**When to use** Use this procedure when you want to add a new pattern configuration record to a pattern.

**Procedure** Use this procedure to add a pattern configuration record.

Step	Action
1	Select <b>Patrn</b> from the host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Patterns screen.
2	Press F2 to list the pattern names; then select the pattern you want to edit and press ENTER.  <b>System Response</b> Gentran:Server displays the Data Manager Patterns screen with additional function keys.  <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> Data Manager Patterns _____ Pattern Name: flow Desc: Inbound Flow QA Test 1 _____ F3:Del F4:Copy F5:Config F6:Meld F9:Quit F10:Save _____ </pre> </div> <p style="text-align: right; color: red;">(Continued on next page)</p>

(Contd) Step	Action
3	<p>Press F5 to configure the pattern.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Selection screen.</p> <pre> Data Manager Selection ----- Data Manager: Name ████████ Type - Desc ----- F2:Select F5:Find F9:Quit F10:Save </pre>
4	<p>Press F5 to display a list of data managers added to this pattern; then select the name of the data manager and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the screen with values in the fields.</p>
5	<p>Press F10 to continue to the Data Manager Pattern Configuration screen.</p> <pre> ----- Data Manager Pattern Configuration ----- Data Manager: Name ap00          Flow: Test Q! flow Source Agent Type m            Application Data Manager Pattern flow      Inbound Flow QA Test 1 -----  Trading Partner: Code ██████████ Modifier ██████████ Description -----  Destination: Directory File Name -----  Script Name Desc -----  Archive Data (y/n) ? n Error Handling Class -----  Mailboxes ----- InBox OutBox-1 OutBox-2 OutBox-3 -----  Queue Info ----- Queue Name Resource Group Priority -----  F1:Token F2:Select F5:Find F6:TP Search F7:Next F8:Prev F9:Quit </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
6	<p>On the Data Manager Pattern Configuration screen, type a value in the Trading Partner Code or File Name field and then press TAB.</p> <p><b>CAUTION</b> To display the token list, press F1. Any token you select is appended to any data you have entered into the field.</p> <p><b>Comment</b> For inbound flows, you usually use the {TPCODE} token in the Trading Partner Code field. For outbound flows where the first data manager in the flow is an application data manager, you usually use the {TPCODE} token in the Trading Partner Code field. For other outbound flows, you usually use the {APPFN} token in the File Name field.</p> <p><b>System Response</b> Gentran:Server displays a message to let you know that the pattern configuration does not exist and asks if you want to add it.</p> <div data-bbox="732 982 1313 1066" style="border: 1px solid black; padding: 5px; text-align: center;"> <p>Confirmation</p> <p>Configuration Record Not Found</p> <p>Do You Wish To Add The Config Record (y/n) ? _</p> </div>
7	<p>Type <b>y</b> at the prompt to add the configuration record.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Pattern Configuration screen and selects the Description field.</p>
8	<p>Complete the remaining fields, inserting appropriate tokens and values.</p> <p><b>CAUTION</b> Gentran:Server checks the Directory field to determine if the directory you entered exists. If it does not, Gentran:Server asks you if you want to add it. You can either add it now or during the meld process.</p> <p>Gentran:Server checks the Script field to determine if the script exists. If it does not exist, Gentran:Server displays a warning message.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
9	<p>Press F10 to save the pattern configuration record.</p> <p><b>System Response</b> Gentran:Server saves the pattern configuration record and clears the fields on the Data Manager Pattern Configuration screen.</p> <p><b>CAUTION</b> <b>When you save a pattern configuration record, Gentran:Server validates the new parameters and records the results in the data manager's log file. The change is logged under the server process ID number. The log entry contains the initialization file parameters.</b></p>
10	<p>Do you want to add another pattern configuration for the data manager?</p> <ul style="list-style-type: none"><li>▶ If YES, repeat Steps 6 through 9.</li><li>▶ If NO, Press F9 to exit the Data Manager Pattern Configuration screen.</li></ul>

# How to Copy a Pattern Configuration Record In a Pattern

**Introduction** You can copy a pattern configuration record and apply it to a new file name or Trading Partnership code and modifier. This creates a new pattern configuration record within the pattern.

**When to use** Use this procedure when you want to copy the configuration information of an existing pattern configuration record to create a new pattern configuration record in the same pattern.

**Procedure** Use this procedure to copy a pattern configuration record in a pattern.

Step	Action
1	<p>Select <b>Pattern</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Patterns screen.</p> <pre data-bbox="651 1192 1382 1373"> Data Manager Patterns _____ Pattern Name: ██████████ Desc: F2:Select F9:Quit _____ </pre>
2	<p>Press F2 to list the pattern names; then select the pattern name and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the screen with additional function keys.</p> <pre data-bbox="651 1629 1390 1810"> Data Manager Patterns _____ Pattern Name: flow Desc: Inbound Flow QA Test 1 F3:Del F4:Copy F5:Config F6:Mod F9:Quit F10:Save _____ </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Press F5 to configure the pattern.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Selection screen.</p> <pre data-bbox="678 548 1370 764"> Data Manager Selection _____ Data Manager: Name ████████ Type - Desc F2:Select F5:Find F9:Quit F10:Save _____ </pre>
4	<p>Press F5 to display a list of data managers that were added to this pattern; then select the name of the data manager and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the screen with values in the fields.</p> <pre data-bbox="678 1010 1370 1226"> Data Manager Selection _____ Data Manager: Name ap00 Type m-Application Data Manager Desc Flow: Test Q! flow Source Agent F2:Select F5:Find F9:Quit F10:Save _____ </pre>
5	<p>Is this the data manager with the pattern configuration record you want to copy?</p> <ul style="list-style-type: none"> <li>▶ If YES, press F10 to continue to the Data Manager Pattern Configuration screen.</li> <li>▶ If NO, press F9 to quit and return to the Data Manager Selection screen to select a different data manager.</li> </ul>
6	<p>On the Data Manager Pattern Configuration screen, select the Trading Partner Code or File Name field and then press F5 to display a list of configured Trading Partnership code or file name values.</p>
7	<p>Select the Trading Partnership code or file name of the pattern configuration you want to copy and press ENTER.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
8	<p>Does the record you want to copy have a Trading Partnership Code modifier?</p> <ul style="list-style-type: none"><li>▶ If YES, select the Modifier field; press F2 to display the available modifiers; then select the modifier from this list.</li></ul> <p><b>System Response</b> Gentran:Server displays the pattern configuration record.</p> <ul style="list-style-type: none"><li>▶ If NO, continue with Step 9.</li></ul>
9	<p>Press F4 to copy the pattern configuration record.</p> <p><b>System Response</b> Gentran:Server displays the Copy screen. The Code field contains the Trading Partnership code or file name of the pattern you copied.</p>
10	<p>Type the new file name or Trading Partnership code and modifier (if any) into the appropriate fields.</p>
11	<p>Press F10 to save the new record.</p> <p><b>Comment</b> Once you save the configuration record, you can make any modifications to it. Be sure to press F10 to save your changes.</p>


## How to Edit a Pattern Configuration Record

**Introduction** You can modify the field values of a pattern configuration record and then use it to generate new configuration records.

**Cautions** **WARNING**

**When you modify a data manager pattern configuration record and then do a meld operation, the system generates new configuration records. This process can overwrite existing records. (The meld operation gives you an option to overwrite configuration records.) For this reason, we recommend that you back up your files before you perform the meld operation. Note that any old records that cannot be overwritten remain in the file.**

**Procedure** Use this procedure to edit a pattern.

Step	Action
1	Select <b>Patrn</b> from the host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Patterns screen.  
2	Press F2 to list the pattern names; then select the pattern you want to edit and press ENTER.  <b>System Response</b> Gentran:Server displays the Data Manager Patterns screen with additional function keys.  <div style="text-align: right; color: red;">(Continued on next page)</div>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Press F5 to configure the pattern.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Selection screen.</p> <pre data-bbox="673 546 1367 766"> Data Manager Selection _____ Data Manager: Name ████████ Type - Desc F2:Select F5:Find F9:Quit F10:Save _____ </pre>
4	<p>Press F5 to display a list of data managers added to this pattern; then select the name of the data manager and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the screen with values in the fields.</p>
5	<p>Is this the data manager you want?</p> <ul style="list-style-type: none"> <li>▶ If YES, press F10 to continue to the Data Manager Pattern Configuration screen.</li> <li>▶ If NO, press F9 to quit and return to the Data Manager Selection screen to select a different data manager.</li> </ul>
6	<p>Select the Trading Partner Code or File Name field; then press F5 to display a list of the Trading Partnership codes or file names configured for the data manager in this pattern.</p>
7	<p>Select the pattern configuration you want to edit and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays pattern configuration record.</p>
8	<p>Modify the field values.</p>
9	<p>Press F10 to save your changes.</p> <p><b>System Response</b> The system clears the fields on the Data Manager Pattern Configuration screen.</p>
10	<p>Press F9 to exit the Data Manager Pattern Configuration screen.</p>


## How to Delete a Pattern

**Introduction** If you no longer need a pattern, you can delete it.

**Comment**

Deleting a pattern does not affect configuration records that were created when the pattern was melded.

**Procedure** Use this procedure to delete a pattern.

Step	Action
1	<p>Select <b>Patrn</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Patterns screen.</p> 
2	<p>Press F2 to list the pattern names; then select the pattern name you want to delete and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the screen with additional function keys.</p>
3	<p>Press F3 to delete the pattern.</p> <p><b>System Response</b> Gentran:Server displays a confirmation prompt.</p>
4	<p>Type <b>y</b> to confirm the deletion.</p>



## How to Use the Meld Function

**Introduction** The meld process merges all the pattern configuration records in a pattern with selected Trading Partnership records to create configuration records.

**Procedure** Use this procedure to generate configuration records from a pattern.

Step	Action
1	<p>Select <b>Patrn</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Patterns screen.</p> <pre data-bbox="657 934 1385 1115"> Data Manager Patterns ----- Pattern Name: ██████████ Desc: ----- F2:Select  F9:Quit           </pre>
2	<p>Press F2 to list the pattern names, select the pattern name you want to meld, and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the screen with additional function keys.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Press F6 to start the meld process.</p> <p><b>System Response</b> Gentran:Server displays the Trading Partnership Search screen.</p> <div data-bbox="789 548 1256 877" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <pre> Trading Partnership Search ----- Do you wish to enter a range of ...  Trading Partnership Codes ?      N Interchange and/or Group Ids ?   N Organization Codes ?            N Standard Version and/or Set Ids ? N User Defined Categories ?        N  Target an Inbound Mapping Table ? N Target an Outbound Mapping Table ? N Search by Trading Partnership Name ? N </pre> </div>
4	<p>Type the search criteria and then press F10 to continue.</p> <p><b>Reference</b> For more information about using the Trading Partnership search function, see the <a href="#">How to Search For a Trading Partnership Record</a> topic in the <a href="#">Understanding the Basics</a> chapter of this guide.</p> <p><b>System Response</b> Gentran:Server displays a list of all the records matching your search criteria.</p>
5	<p>Type <b>y</b> in the Proc Record (Process Record) field of each record you want to meld with the pattern.</p> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>▶ To select every Trading Partnership code, press F2 (Mark All).</li> <li>▶ To remove all the y's from the Proc Record field, press F3 (Unmark All).</li> </ul>
6	<p>Press F10 to continue.</p> <p><b>System Response</b> Gentran:Server begins the meld process and asks if you would like the meld process to overwrite duplicates with new entries.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
7	<p>Do you want Gentran:Server to overwrite duplicates with new entries?</p> <ul style="list-style-type: none"> <li>▶ If YES, type <b>y</b> at the prompt.</li> <li>▶ If NO, type <b>n</b> at the prompt.</li> </ul> <p><b>System Response</b> Gentran:Server asks if you would like the process to create the destination directories.</p>
8	<p>Do you want Gentran:Server to create the destination directories if they do not exist?</p> <ul style="list-style-type: none"> <li>▶ If YES, type <b>y</b> at the prompt.</li> <li>▶ If NO, type <b>n</b> at the prompt.</li> </ul> <p><b>System Response</b> Gentran:Server asks if you would like the process to run a compliance check for blank or empty tokens.</p>
9	<p>Do you want Gentran:Server to run a compliance check for tokens?</p> <ul style="list-style-type: none"> <li>▶ If YES, type <b>y</b> at the prompt.</li> <li>▶ If NO, type <b>n</b> at the prompt.</li> </ul> <p><b>System Response</b> Gentran:Server melds the pattern with the selected Trading Partnership codes or file names. When the processing is finished, Gentran:Server displays a <b>meld log</b> with messages that apply to the operations.</p> <p><b>Sample Meld Log</b> This is an example of a meld log.</p> <pre style="margin-left: 40px;"> Notice: Begin Meld Operations for Pattern: Outbound_Test  35 Total record(s) processed   0 record(s) with errors  35 record(s) sent to load  Notice: End Meld Operations for Pattern: Outbound_Test  35 Total record(s) read  27 new record(s)   8 duplicate record(s) rewritten   0 bad record(s)  35 Total record(s) written to: dm </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
10	Use the arrow keys to scroll the log. Use <b>R</b> or <b>r</b> to resize the viewing screen.  <b>Comment</b> The end of the log tells you how many new records were created and the number of records that were rewritten.
11	Press ESC to exit the log.  <b>WARNING</b> <b>The meld log is a temporary file. Once you leave this screen, Gentran:Server deletes the file.</b>  <b>Reference</b> To verify that the configuration records were created, see the <a href="#">How to Verify Creation of Configuration Records</a> topic in this chapter.

---

# Using the Inbound Data Manager

## How the Inbound Data Manager Creates Records

---

**Introduction** You can configure Gentrans:Server to enable an inbound (personality 2) data manager to automatically generate Trading Partnership and configuration records. The inbound data manager generates the records when the data manager encounters a file with a particular standard version and set ID.

---

**How this feature works**

This table describes how the records are created.

Stage	Description
1	The inbound data manager reads the file in its work directory and examines the values in the EDI envelope.
2	The inbound data manager checks the Trading Partner file for a Trading Partnership code record that matches the values in the EDI envelope, but cannot find a match.

(Continued on next page)

<b>(Contd) Stage</b>	<b>Description</b>	
3	The inbound data manager checks its initialization file to determine the action to take.	
	<b>IF...</b>	<b>THEN...</b>
	The ADD_TP flag is set and a model interchange, group, and Trading Partnership record is set up for the version and set ID	<p>The inbound data manager starts creation of a Trading Partnership record, using information in the inbound document to complete the record.</p> <p>Continue with Stage 4.</p> <p><b>Comment</b> The Trading Partnership code for the new Trading Partnership record is &lt;setid&gt;&lt;unique_id&gt;. If the set ID is 850, the Trading Partnership code is 850 plus a unique 6-digit numeric ID that Gentran:Server generates: 850359274</p>
	The ADD_TP flag is not set	The NO_TP error handler routes the file. The data manager does not start generation of the configuration records.
4	The data manager invokes the meld process for the pattern name specified in the <b>Your ID</b> field of the model Interchange record. The meld process creates the configuration records.	

# How to Have the Inbound Data Manager Create Records

**Introduction** This topic explains how to have the inbound data manager generate Trading Partnership and configuration records.

**When to use** Use this procedure when you:

- ▶ Receive files for which you cannot identify the trading partner in advance
- ▶ Expect to receive a number of files from so many sources that creating configuration records for each source is impractical.

**Procedure** Use this procedure to have an inbound data manager generate Trading Partnership and configuration records.

Step	Action
1	<p>Set the ADD_TP parameter in the inbound data manager's initialization file to 1 (on).</p> <p><b>Reference</b> See the <a href="#">Maintaining Initialization Files</a> chapter in this guide.</p>
2	<p>Identify the data managers that are downstream of the inbound data manager. For each downstream data manager, set the ADD_CONFIG parameter in the data manager's initialization file to 1 (on).</p> <p><b>Reference</b> See the <a href="#">Maintaining Initialization Files</a> chapter in this guide.</p>
3	<p>Use the pattern feature to:</p> <ul style="list-style-type: none"> <li>▶ Create a pattern for the inbound flow and name it</li> <li>▶ Add the inbound data manager to the pattern</li> <li>▶ Create a pattern configuration record.</li> </ul> <p><b>Reference</b> See the <a href="#">How to Create a New Pattern</a> topic in this chapter for instructions.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
4	<p>Create a default configuration record for each data manager that is downstream from the inbound data manager. Use <b>_default</b> as the File Name in each record.</p> <p><b>Reference</b> See the <a href="#">How to Manually Create a Configuration Record</a> topic in this chapter.</p>
5	<p>Use the Gentran:Server Trading Partner Administration functions to set up a model interchange record.</p> <ul style="list-style-type: none"> <li>▶ Type the name of the pattern you created in Step 3 into the Interchange: Your ID field.</li> <li>▶ Type <b>_default</b> into your partner's interchange ID field.</li> </ul> <p><b>Reference</b> See the Gentran:Server <i>for UNIX and Workstation Application Integration User's Guide</i> for information about Trading Partnership administration.</p>
6	<p>Use the Gentran:Server mapping and translation functions to set up a model group record.</p> <p>Type <b>_default</b> into the fields that hold your group ID and your partner's group ID.</p> <p><b>Reference</b> See the Gentran:Server <i>for UNIX and Workstation Application Integration User's Guide</i> for information about Trading Partnership administration.</p>
7	<p>Use the Gentran:Server Trading Partner Administration subsystem to set up a model Trading Partnership record for every standard version and set ID combination you expect the data manager to encounter.</p> <ul style="list-style-type: none"> <li>▶ On the Trading Partnership tab, use the standard version number and the set ID as the Trading Partnership code.</li> <li>▶ On the Inbound EDI Information tab, type the same standard version number in the Standard version field. Type the set ID into the Document ID field.</li> </ul> <p><b>Reference</b> See the <a href="#">Working with Trading Partnerships</a> chapter in the Gentran:Server <i>for UNIX and Workstation Application Integration User's Guide</i> for information about creating a Trading Partnership record.</p>

(Continued on next page)



**Tip**

If you want Gentran:Server to send a mail message that contains information about the records created through the inbound data manager, you must turn on the SEND\_MAIL\_TP flag in the inbound data manager's initialization file.

**Reference**

See the [Maintaining Initialization Files](#) chapter in this guide.

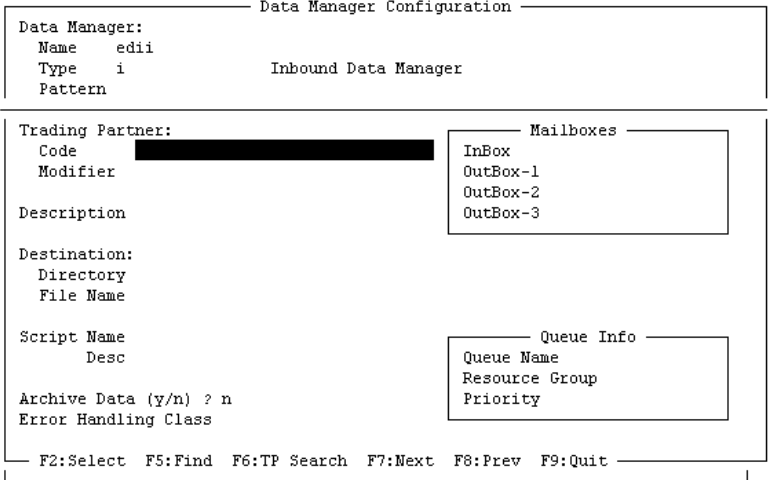
---

# Maintaining Configuration Records

## How to Verify Creation of Configuration Records

**Introduction** If you use the pattern and meld procedure or an inbound data manager to generate configuration records, you should verify that the records were created.

**Procedure** Use this procedure to verify that the configuration records were created.

Step	Action
1	Open the Data Manager Control screen and select the data manager of the records you want to verify.
2	<p>Press F5 to display the Data Manager Configuration screen.</p> 
3	<p>Highlight the Trading Partner Code or File Name field and press F5 to view a list of the records that were created.</p> <p><b>Comment</b> The list includes manually created configuration records.</p>

# How to Edit a Configuration Record

**Introduction** You can edit any configuration record created in EC Workbench.

## CAUTION

**You cannot use EC Workbench to modify flow components created with the PCM Wizard.**

**When to use** Use this procedure when:

- ▶ The configuration record is not directing files correctly
- ▶ Changes to your system require you to change how a data manager handles files for a specific Trading Partnership or file name.

**Procedure** Use this procedure to edit a configuration record.

Step	Action
1	<p>Select <b>DataMgr</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p> <pre> Data Manager Control Name A      Status  T Description ----- fmgr A      1239  F Foreground Manager (IPC Control) alnm n      ***** l Async Line Manager ap00 n      ***** m Flow: Test Q! flow Source Agent ap01 n      ***** m Flow: 'nother fifteen Translate Agent appm n      ***** m Application Data Manager appt n      ***** x Application Translator Data Manager arch n      ***** a -Darch -Aarch -d0 base n      ***** u Base Manager Model cfin n      ***** i Flow: chris_flow Translate Agent dhld n      ***** d UDF Data Manager edii n      ***** i Inbound Data Manager edio n      ***** i Outbound Data Manager file n      ***** f File Data Manager hcmd n      ***** h Host Command Card Data Manager in00 n      ***** i Flow: flow Source Agent in01 n      ***** i Flow: fifteen digits. Translate Agent in02 n      ***** i Flow: test0505 Translate Agent F2:Arch F3:Stop F4:EditMail F5:Config F6:Stat F7:Log F8:Start F9:Quit           </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

(Contd) Step	Action
2	<p>Select the data manager of the record you want to edit and then press F5 to configure the data manager.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.</p> <pre> Data Manager Configuration ----- Data Manager: Name      edii Type      i           Inbound Data Manager Pattern  Trading Partner: Code      ██████████ Modifier  ██████████ Description  Destination: Directory File Name  Script Name Desc  Archive Data (y/n) ? n Error Handling Class  Mailboxes ----- InBox OutBox-1 OutBox-2 OutBox-3  Queue Info ----- Queue Name Resource Group Priority  F2:Select F5:Find F6:TP Search F7:Next F8:Prev F9:Quit </pre>
3	<p>Select the Trading Partner Code or File Name field; then press F5 to display the records that are configured for the data manager and select from this list.</p> <p>If the record you want has a Trading Partnership Code modifier, select the Modifier field; press F2 to display the available modifiers; then select from this list.</p> <p><b>System Response</b> Gentran:Server displays the configuration record.</p>
4	<p>Modify the field values.</p> <p><b>Reference</b> See the <a href="#">The Data Manager Configuration Screen</a> topic in this chapter for field descriptions.</p>
5	<p>Press F10 to save your changes.</p>

# How to Delete a Single Configuration Record

**Introduction** This topic covers how to delete a single configuration record.

**Reference**

To delete a range of records, see [How to Delete a Range of Configuration Records](#).

**Deleting a single record**

Use this procedure to delete a configuration record.

Step	Action
1	<p>Select <b>DataMgr</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p> <pre> Data Manager Control Name A      Status  T Description ----- Emgr A      1239 F Foreground Manager (IPC Control) alnm n ***** l Async Line Manager ap00 n ***** m Flow: Test Q! flow Source Agent ap01 n ***** m Flow: 'nother fifteen Translate Agent appm n ***** m Application Data Manager appt n ***** x Application Translator Data Manager </pre>
2	<p>Select the data manager associated with the entry you want to delete; then press F5 to configure the data manager.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.</p>
3	<p>Display the configuration record you want to delete.</p> <p><b>Reference</b> See the <a href="#">How to Open a Configuration Record</a> topic in this chapter.</p>
4	<p>Press F3 to delete the entry.</p> <p><b>System Response</b> The system displays a confirmation prompt.</p>
5	<p>Type <b>y</b> to confirm the deletion.</p>

## How to Delete a Range of Configuration Records

### Introduction

The Remove Data Manager TP Configurations screen enables you to delete a range of configuration records at the same time. To delete a single record, see [How to Delete a Single Configuration Record](#).

### Note

If you delete a data manager, the system deletes the configuration records associated with the data manager.

### Reference

See the topic [How to Delete a Data Manager](#) in the Working with Data Managers chapter of this guide.

### Remove Data Manager TP Configurations screen

This illustration shows the Remove Data Manager TP Configurations screen.

```

Remove Data Manager TP Configurations
-----
Pattern Name ██████████
Data Manager

Starting TP Code
Ending TP Code

F2:Select  F9:Quit  F10:Save
  
```

### Fields and functions

This table lists the fields of the Remove Data Manager TP Configurations screen and their functions.

Field	Function
Pattern Name	<p>The name of the pattern, if any. Press F2 to display a list of acceptable values.</p> <p><b>Comment</b> To select all data manager patterns, select <b>ALL</b> from the list. If the configuration records were created manually rather than with a pattern, select <b>NONE</b> from the list. Otherwise, select a pattern name.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Field</b>	<b>Function</b>
Data Manager	<p>The name of the data manager.</p> <p><b>Comment</b> Press F2 to display a list; then select a data manager name or select <b>ALL</b> to choose all data managers.</p>
Starting TP Code	<p>The first Trading Partnership code you want to remove.</p> <p><b>Comment</b> If you want to remove all the configuration records for the data manager, type <b>ALL</b> in this field. If you leave this field blank it defaults to <b>ALL</b>.</p>
Ending TP Code	<p>The last Trading Partnership code of the records you want to delete.</p> <p><b>Comment</b> If you want to remove only the Trading Partnership code specified in the Starting TP Code field, press F3 to enter the starting Trading Partnership code in this field.</p> <p>If you are entering a range of codes, you can press F4 to enter "To End of File" in this field. The "To End of File" option removes all configuration records to the end of file.</p>

---

(Continued on next page)

## Deleting a range of records

Use this procedure to delete a range of configuration records at the same time.

Step	Action
1	<p>Press CTRL+E from any Gentran:Server Data Flow Administration screen.</p> <p><b>System Response</b> Gentran:Server displays the Remove Data Manager TP Configurations screen.</p>
2	<p>Complete the fields to identify the records you want to remove.</p> <div data-bbox="695 751 1349 989" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> Remove Data Manager TP Configurations ----- Pattern Name Inbound_Test Data Manager edii  Starting TP Code ██████████ Ending TP Code  F9:Quit  F10:Save  ----- </pre> </div> <p><b>Comment</b> If you want to remove all the records and then rebuild them with the meld process, type ALL in the Pattern Name, Data Manager, and Starting TP Code fields.</p>
3	<p>Press F10 to start the removal process.</p> <p><b>System Response</b> Gentran:Server displays a confirmation prompt.</p> <div data-bbox="654 1352 1393 1434" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> ----- Confirmation ----- All Trading Partner Data Manager Configurations that match your selected criteria will be removed.  Are You Sure ? </pre> </div>
4	<p>Type <b>y</b> to confirm that you want the records removed.</p> <p><b>System Response</b> Gentran:Server displays a message that shows the number of records that were removed.</p>



---

# Using Queues

<b>Contents</b>	<b>Overview</b>	
	<ul style="list-style-type: none"> <li>▶ Introduction ..... 2</li> <li>▶ Queues ..... 4</li> <li>▶ The Queue Process ..... 6</li> <li>▶ Queue Select Screen ..... 7</li> <li>▶ Queue File Screen ..... 8</li> </ul>	
	<b>Configuring Gentran:Server to Use a Queue</b>	
	<ul style="list-style-type: none"> <li>▶ The Flow of Work ..... 10</li> </ul>	
	<b>How to Create a Queue</b>	
	<ul style="list-style-type: none"> <li>▶ How to Configure the Downstream Data Manager ..... 13</li> <li>▶ How to Enter Queue Data Into a Configuration Record ..... 16</li> <li>▶ How to Configure the Upstream Data Manager ..... 18</li> <li>▶ Maintaining Queue Entries ..... 21</li> <li>▶ Overview ..... 21</li> <li>▶ How to View Entries in a Queue ..... 22</li> <li>▶ Add Queue Entry Screen ..... 24</li> <li>▶ How to Add an Entry to a Queue ..... 26</li> <li>▶ How to Delete an Entry From a Queue ..... 28</li> </ul>	
	<b>Maintaining Queues</b>	
	<ul style="list-style-type: none"> <li>▶ Overview ..... 30</li> <li>▶ How to Remove a Queue from the Select List ..... 31</li> <li>▶ How to Delete a Queue ..... 32</li> </ul>	

---

# Overview

## Introduction

---

**In this chapter** This chapter describes queues and how to use them in Gentran:Server.

---

**Key terms** This table lists the key terms used in this chapter.

Term	Description
configuration record	<p>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:</p> <ul style="list-style-type: none"> <li>▶ Specifies the Trading Partnership code or file name that the data manager is to use to identify data</li> <li>▶ Tells the data manager what to do with the data it has identified.</li> </ul>
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	<p>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.</p> <p><b>Comment</b> An ISAM file has the file extensions <i>.dat</i> and <i>.idx</i>.</p>
priority	<p>A number from 0 through 9 that designates the order in which a downstream data manager is to process files.</p> <p>A 0 represents the highest priority and 9 represents the lowest priority.</p>
queue	<p>An indexed list of files to be processed.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

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

Gentran:Server uses the *<queue name>/q.idx* part of the ISAM file to access the records in the *<queue name>/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 Gentran:Server assigned.

---

**Reading from and writing to queues**

Any data manager can use a queue. You can configure a data manager to read the information in the queue to locate files to process. You can also configure a data manager to write to a queue after it processes files so that the next data manager in the flow can locate and process the files.

---

**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**.

---

(Continued on next page)

---

**Queue names**

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

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

**Example**

You create a queue and name it *inque1*.

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
    - With a particular Trading Partnership modifier 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.  <b>Comment</b> 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.

```

Select
inbd
xlti
xlto

```

```

<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.

```

QueueFile: xlt0/.q
P Uniq Directory/File PID Group Num
000027 xlt0/OUTBND02856.000000143 xlo1 0
000028 xlt0/OUTBOUND856.000000145 xlo1 0
000029 xlt0/OUTBND02856.000000146 xlo1 0
000030 xlt0/OUTBND03856.000000147 xlo1 0
000031 xlt0/OUTBND02856.000000149 xlo1 0
000032 xlt0/OUTBOUND856.000000154 xlo1 0
000033 xlt0/OUTBOUND856.000000156 xlo1 0
F1:Zoom F2:Add F3:Del F4:First F5>Last F6:Srch F7:Next F8:Prev F9:Quit

```

### Queue File screen fields and functions

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

Field	Function
P	<p>Defines the processing priority assigned to the file, where 0 represents the highest priority and 9 represents the lowest priority.</p> <p><b>Comment</b> If the processing priority field is blank (as in the above example), the system uses 9 as the default priority value.</p>
Uniq	<p>Displays the unique number Gentran:Server assigned to this queue entry.</p>
Directory/File	<p>Specifies the name of the directory and the name of the file.</p> <p><b>Comment</b> The screen field size is limited. To see the full path, press F1.</p>

(Continued on next page)



<b>(Contd) Field</b>	<b>Function</b>
PID	Not used.
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.

<b>Key</b>	<b>Function</b>
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.

# Configuring Gentran:Server to Use a Queue

## The Flow of Work



**Task summary** This table describes the process for configuring Gentran:Server to use a queue.

Stage	Description
1	Create the queue. <b>Reference</b> See the <a href="#">How to Create a Queue</a> topic in this chapter.
2	Configure the downstream data manager's initialization file. <b>Reference</b> See the <a href="#">How to Configure the Downstream Data Manager</a> topic in this chapter.
3	Enter the queue information into the upstream data manager's configuration records. <b>Reference</b> See the <a href="#">How to Enter Queue Data Into a Configuration Record</a> topic in this chapter
4	Configure the upstream data manager's initialization file. <b>Reference</b> See the <a href="#">How to Configure the Upstream Data Manager</a> topic in this chapter.

# How to Create a Queue

**Introduction** 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	<p>Select <b>Queue</b> from the Util menu.</p>  <p><b>System Response</b> Gentran:Server displays the Select screen. This screen lists the existing queues.</p>
3	<p>Press F2 to add a new queue.</p>  <p>&lt;CR&gt;: Select F2: Add a Queue F3: Del Q Reference F4: Del Entire Q F9: Exit</p> <p><b>System Response</b> Gentran:Server displays the Add screen.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
4	<p>Type the name of the queue.</p> <div data-bbox="878 449 1175 575" style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"><p>Add</p><p>Name: <input type="text" value="ino3"/></p><p>F9:Exit F10:Save</p></div> <p><b>Comment</b> You can use up to 9 characters in the queue name. Use only numbers or letters. Do not use symbols or spaces.</p>
5	<p>Press F10.</p> <p><b>System Response</b> Gentran:Server creates the queue and adds the queue name to the Select list.</p>

---

# How to Configure the Downstream Data Manager

**Introduction** You must configure a downstream data manager to read from the queue before you configure an upstream data manager to write to the queue.

**Procedure** Use this procedure to configure the downstream data manager.

Step	Action
1	Select <b>DataMgr</b> from the Gentran:Server host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.
2	Select <b>fmgr</b> , the Foreground Manager.  <pre> Data Manager Control Name A      Status  T Description ----- fmgr: A      1239  F Foreground Manager (IPC Control) alnm n      *****  l Async Line Manager ap00 n      *****  m Flow: Test Q! flow Source Agent ap01 n      *****  m Flow: 'nother fifteen Translate Agent appm n      *****  m Application Data Manager appt n      *****  x Application Translator Data Manager arch n      *****  a -Darch -Aarch -d0 base n      *****  u Base Manager Model </pre>
3	Press F5 to configure the Foreground Manager data managers.  <b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.  <pre> dmc Data Manager Configuration Name A      Status  T Description ----- alnm n      0  l Async Line Manager ap00 n 0      0  m Flow: Test Q! flow Source Agent ap01 n 0      0  m Flow: 'nother fifteen Translate Agent appm n      0  m Application Data Manager appt n      0  x Application Translator Data Manager </pre> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
4	Select the downstream data manager you want to configure.  <b>WARNING</b> <b>You cannot edit an active data manager. Before you continue, check the Status field to make sure the data manager is not running. To stop the data manager, return to the Data Manager Control screen and inactivate the data manager by pressing F3. Once the data manager stops, you can edit it.</b>
5	Press F5 to edit the data manager.  <b>System Response</b> Gentran:Server displays the edit screen for the selected data manager.
6	Press F5 to edit the initialization file.  <b>System Response</b> Gentran:Server calls the vi editor or the editor set in the \$EDITOR variable and displays the initialization file. If an initialization file does not exist for the data manager, Gentran:Server copies the base initialization file and renames it for the data manager.
7	Set the WORK_DIRECTORY parameter to the queue that this data manager reads.  <b>Example</b>  <pre>:WORK_DIRECTORY          xlti</pre>
8	Set the WORK_TYPE parameter to q.  <b>Example</b>  <pre>:WORK_TYPE                q</pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
9	<p>Set the RESOURCE_GROUP parameter to the name of the resource group assigned to the files that you want this data manager to process.</p> <p><b>Comment</b> If you have more than one data manager read from the queue, use resource groups to have each data manager process only files that are in its resource group.</p> <p><b>WARNING</b> <b>You must supply a value in this field. If you use “default” as the value, you must enter: RESOURCE_GROUP default</b> <b>Do not leave this parameter blank.</b> <b>You will use the RESOURCE_GROUP name in the Resource Group field of the upstream data manager’s configuration record.</b></p>
10	Save your changes.
11	<p>Exit the editor.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p> <p><b>CAUTION</b> <b>When you change a data manager’s initialization file, Gentran:Server validates the new parameters and records the results in the data manager’s log file. The change is logged under the UNIX process ID number of the process named “server.” The log entry contains the initialization file parameters.</b></p>

## How to Enter Queue Data Into a Configuration Record

### Introduction

You must enter queue information into the configuration records of the upstream data manager. The queue information tells the data manager:

- ▶ The name of queue this data manager writes to
- ▶ The name of the resource group to assign to the files this data manager processes
- ▶ The order (priority) in which the downstream data manager should process the files.

### Before you begin

Before you begin this procedure, you must:

- ▶ Create the queue
- ▶ Configure the downstream data manager to read the queue and look for the resource group.

### Procedure

Use this procedure to specify the queue information that an upstream data manager uses for the files it processes.

Step	Action
1	<p>Open an existing configuration record or create a new configuration record that describes how you want the upstream data manager to process files that match the Trading Partnership code or file name.</p> <p><b>Reference</b> See the <a href="#">Working with Configuration Records</a> chapter in this guide for information about creating and editing configuration records.</p>
2	<p>Select the <b>Queue Name</b> field located in the Queue Info box and press F2 to select from the list of queue names.</p>
3	<p>Select the <b>Resource Group</b> field and press F2 to select the resource group from the list of resource groups.</p> <p><b>WARNING</b> <b>Make sure that the name of the Resource Group matches the name in the RESOURCE_GROUP parameter of the downstream data manager's initialization file.</b></p> <p style="text-align: right;">(Continued on next page)</p>



(Contd) Step	Action
4	<p>Type the processing priority you want to assign to the files into the <b>Priority</b> field. Use a number from 1 through 9, where 1 represents the highest priority and 9 represents the lowest priority.</p> <p><b>Comments</b> If you leave the Priority field blank, Gentran:Server uses 9 as the default value. You can use the F2 key to display a select list.</p> <p><b>Example</b> In this example, the queue name is <b>xlti</b> and the resource group name is <b>xlti</b>. The priority is <b>1</b>.</p> <div data-bbox="672 772 1382 1213" style="border: 1px solid black; padding: 5px;"> <pre> Data Manager Configuration ----- Data Manager: Name      edii Type      1 Pattern   Inbound_Test Inbound Data Manager  Trading Partner: Code      inbnd850 Modifier  Description flow for 850  Destination: Directory xlti/ File Name CCDinbd850D  Script Name mv_inbd Desc ** Script Not On File **  Archive Data (y/n) ? Error Handling Class default  Mailboxes ----- InBox OutBox-1 OutBox-2 OutBox-3  Queue Info ----- Queue Name   xlti Resource Group xlti Priority      1  F3:Delete F4:Copy F7:Next F8:Prev F9:Quit F10:Save </pre> </div>
5	Press F10 to save the configuration record.

# How to Configure the Upstream Data Manager

- Introduction** You must configure the upstream data manager's initialization file to ensure that the upstream data manager writes to a queue only if a downstream data manager:
- Reads from the upstream data manager's destination queue
  - Is configured to process files in the resource group that the upstream data manager assigns to the files it processes.

**Procedure** Use this procedure to configure the upstream data manager.

Step	Action
1	<p>Select <b>DataMgr</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p>
2	<p>Select <b>fmgr</b>, the Foreground Manager.</p> <pre> Data Manager Control Name A      Status  T Description ----- fmgr A      1239 F Foreground Manager (IPC Control) alnm n      ***** l Async Line Manager ap00 n      ***** m Flow: Test Q! flow Source Agent ap01 n      ***** m Flow: 'nother fifteen Translate Agent appm n      ***** m Application Data Manager appt n      ***** x Application Translator Data Manager arch n      ***** a -Darch -Aarch -d0 base n      ***** u Base Manager Model </pre>
3	<p>Press F5 to configure the Foreground Manager data managers.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Configuration screen.</p> <pre> dmc Data Manager Configuration Name A      Status  T Description ----- alnm n      0 l Async Line Manager ap00 n 0      m Flow: Test Q! flow Source Agent ap01 n 0      m Flow: 'nother fifteen Translate Agent appm n      0 m Application Data Manager appt n      0 x Application Translator Data Manager </pre> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
4	Select the upstream data manager you want to configure.  <b>WARNING</b> <b>You cannot edit an active data manager. Before you continue, check the Status field to make sure the data manager is not running. To stop the data manager, return to the Data Manager Control screen and inactivate the data manager by pressing F3. Once the data manager stops, you can edit it.</b>
5	Press F5 to edit the data manager.  <b>System Response</b> Gentran:Server displays the edit screen for the selected data manager.
6	Press F5 to edit the initialization file.  <b>System Response</b> Gentran:Server calls the vi editor or the editor set in the \$EDITOR variable and displays the initialization file. If an initialization file does not exist for the data manager, Gentran:Server copies the base initialization file and renames it for the data manager.
7	Set the WORK_DIRECTORY parameter to the directory that this data manager reads.  <b>Comment</b> Typically, a file starts off in a directory, but the source can be a queue.
8	Set the WORK_TYPE parameter to <b>d</b> for directory or <b>q</b> for queue.  <b>Example</b> <pre>WORK_TYPE      d</pre>
9	Set the VAL_RESOURCE_GRP parameter to <b>1</b> .  <b>Comment</b> If the parameter is set to 1, the system checks to see whether another data manager is set to scan for the resource group that is being added to the queue.  <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
10	Save your changes.
11	Exit the editor.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.  <b>Comment</b> When you change a data manager's initialization file, Gentran:Server validates the new parameters and records the results in the data manager's log file. The change is logged under the UNIX process ID number of the process named "server." The log entry contains the initialization file parameters.

---

# 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	<p>Select <b>Queue</b> from the Util menu.</p> <p><b>System Response</b> Gentran:Server displays the queue Select screen.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre> Select inbd xlti xlto </pre> </div> <p>&lt;CR&gt;: Select F2: Add a Queue F3: Del Q Reference F4: Del Entire Q F9: Exit</p>
3	<p>Use the cursor keys to select the name of the queue you want to view and then press ENTER (the carriage return key).</p> <p><b>System Response</b> Gentran:Server displays the Queue File screen for the selected queue.</p>
4	Use the screen's F7 and F8 function keys to navigate through the entries.

(Continued on next page)

## 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 <b>Priority (P)</b> and <b>Unique (Uniq)</b> fields and press F10. <b>System Response</b> 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
Unique Id      000001
Priority       █
Directory
Filename
Process Id
Resource Group
Number of Jobs
F9:Quit  F10:Save
  
```

**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 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.

(Continued on next page)



**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 `svr_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.  <b>Reference</b> See the <a href="#">How to View Entries in a Queue</a> .
2	<b>Press F2.</b>  <b>System Response</b> Gentran:Server displays the Add Queue Entry screen.   <pre> Add Queue Entry ----- Unique Id      000001 Priority Directory Filename Process Id Resource Group Number of Jobs ----- F9:Quit  F10:Save </pre>
3	Complete the fields.
4	Press F10 to save the entry.

### Using the `svr_enq` program to add a record

The `svr_enq` command line program adds entries to a queue. You can use this command in a Gentran:Server script.

The command format is:

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

(Continued on next page)

This table describes the arguments in the **svr\_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.  <b>Note</b> 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 `svr_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.  <b>Reference</b> See the <a href="#">How to View Entries in a Queue</a> topic in this chapter.
2	Use the cursor or function keys to locate and select the entry you want to delete.
3	Press F3.  <b>System Response</b> Gentran:Server displays a confirmation screen.  
4	Type <b>y</b> to confirm the deletion.  <b>System Response</b> Gentran:Server deletes the entry from the queue.

### Using the `svr_deq` program to delete an entry

The `svr_deq` command line program deletes entries from a queue. You can use this command in a Gentran:Server script.

The command format is:

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

(Continued on next page)

This table describes the arguments in the command line format.

<b>Argument</b>	<b>Description</b>
-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	<p>Select <b>Queue</b> from the Util menu.</p> <p><b>System Response</b> Gentran:Server displays the queue Select screen.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre> Select inbd xlti xlto </pre> </div> <p>&lt;CR&gt;: Select F2: Add a Queue F3: Del Q Reference F4: Del Entire Q F9: Exit</p>
3	Use the cursor keys to select the name of the queue you want to remove.
4	<p>Press F3.</p> <p><b>System Response</b> Gentran:Server removes the queue name from the Select screen.</p>

## How to Delete a Queue

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

### CAUTION

**Before you delete a queue, remove all references to the queue in the data manager initialization files and configuration records.**

**Procedure** Use this procedure to delete a queue.

Step	Action
1	Go to the host main menu.
2	<p>Select <b>Queue</b> from the Util menu.</p> <p><b>System Response</b> Gentran:Server displays the queue Select screen.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <pre> Select inbd xlti xlto </pre> </div> <p>&lt;CR&gt;: Select F2: Add a Queue F3: Del Q Reference F4: Del Entire Q F9: Exit</p>
3	Use the cursor keys to select the name of the queue you want to delete.
4	<p>Press F4.</p> <p><b>System Response</b> Gentran:Server deletes the queue and removes the queue name from the Select screen.</p>



---

# Archiving Your Data

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	2
	▶ Archiving .....	4
	▶ The Archive Handler .....	6
	▶ Flow of Data Through Archive Handler .....	8
	▶ The Archiving Process .....	9
	<b>Configuring Your System to Archive Data</b>	
	▶ Overview .....	10
	▶ How to Configure a Data Manager to Archive Data .....	11
	▶ How to Set a Configuration Record to Archive Data .....	12
	▶ How to Configure an Archive Handler .....	13
	<b>Managing Your Archives</b>	
	▶ Overview .....	15
	▶ The cl_arch Program .....	16
	▶ The longterm Script .....	17
	▶ How to Edit the Longterm Script .....	19
	▶ How to Manually Purge Your Archives .....	22
	<b>Retrieving Archived Files</b>	
	▶ Overview .....	25
	▶ The Retrieval Process .....	26
	▶ The Archive Query Screen .....	27
	▶ The Archive List Screen .....	29
	▶ How to Select and Retrieve Archived Files .....	31
	▶ The Retrieval Log .....	34
	▶ How to Display the Retrieval Log .....	35

---

# Overview

## Introduction

**In this chapter** This chapter explains how to:

- ▶ Configure your system to archive data
- ▶ Maintain your archived files
- ▶ Retrieve archived data.

**Key terms** This table lists the key terms used in this chapter.

Term	Description
active archive period	The number of days prior to today for which the system maintains active archive entries in the archive index file.
arch	The default name of the archive handler.
archive	The process of capturing and storing a copy of a document after a data manager has processed it.
archive directory	The directory to which a data manager routes an archive copy of a document it processes. This directory is the archive handler's work directory.
archive data file	The file that contains the data that the archive handler processed and archived. The format of the file name is <i>&lt;dmname&gt;.ccyymmddhh</i> , where <i>ccyymmddhh</i> represents the date and time that the data was archived.
archive handler	The special data manager designed to archive data.
archive index file	An index file that contains a detailed record of the archived files for a specific data manager.
audit file	The file that a data manager produces to pass archiving instructions to the archive handler.
cl_arch	The program that purges your archives.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
document reference number	The number used to identify and track a document throughout its life in Gentran:Server. This number also is used to track the archived documents.
EDI_ARCHIST	The environment variable that sets the default active archive period.
longterm	The script that controls long-term archiving.
rtv_arc log	The log produced when you retrieve archived files. Contains a description of the entries retrieved, their names, and where they were sent.

---

# Archiving

---

**Introduction** Archiving is the process of storing a copy of a document after a data manager handles it.

---

**Why archive data?** Some auditors require an archive copy of the document at each point in the process where the document may change. They also may require a process history of the document. Your organization's auditing requirements should determine how you structure Gentran:Server to archive the files that it processes.

---

**Each data manager can archive data** Each data manager handles archiving separately. You specify archiving at both the data manager level and at the Trading Partnership code or file name level.

To have a data manager archive data, you must:

- ▶ Set the archiving parameters, which include the name of the **archive directory** (the destination directory for archived documents), in the data manager's initialization file
- ▶ Set the **Archive Data** field to **y** (yes) in each configuration record for which you want the data manager to archive data.

---

**Audit files** When a data manager routes an archived copy of a file to its archive directory, it also sends a set of instructions for archiving the file. This set of instructions is called an audit file.

An audit file contains:

- ▶ The names of the files to be archived
- ▶ The name of the data manager that processed the file
- ▶ The time the file was processed
- ▶ Instructions for archiving the data.

---

(Continued on next page)

---

<b>Audit file name</b>	<p>Audit files have an underscore ( <code>_</code> ) as a prefix to identify them as audit files. The file name format is <code>&lt;dmname&gt;ddhhmmss</code>, where <code>ddhhmmss</code> represents the day, hour, minute, and second that the data manager created the audit file.</p> <p><b>Example</b> The data manager named <code>inbd</code> creates an audit file named <code>_inbd21053442</code> on the 21st at 05:34:42 a.m.</p>
<b>Archive file</b>	<p>The file that contains a copy of the data that is to be archived is called the archive file. The file name format is <code>&lt;dmname&gt;&lt;uniqueid&gt;</code>. This is the file that the data manager sends to the archive handler's work directory.</p>
<b>Archived data file</b>	<p>The archived data file contains the data that the archive handler processed and archived. The format of the file name is <code>&lt;dmname&gt;.ccyymmddhh</code>, where <code>ccyymmddhh</code> represents the date and time that the data was archived.</p>
<b>Archive index file</b>	<p>The archive index file contains a record of each document archived for the data manager. The format of the archive index file name is <code>&lt;dmname&gt;.dat</code> and <code>&lt;dmname&gt;.idx</code>.</p>

---

## The Archive Handler

---

**Definition** An archive handler is a special data manager that archives documents. The default name of the archive handler is arch.

---

**Archive handler functions**

An archive handler:

- ▶ Scans its work directory for audit files that other data managers placed in the directory
- ▶ Checks the audit files that the data managers sent to determine how to process the documents
- ▶ Processes the documents
- ▶ Enters record details about the processed documents into the data manager's archive index file
- ▶ Routes the archived data to the data manager's archive data file.

---

**Archive handler configuration**

You configure an archive handler somewhat differently than the way you configure other data managers.

- ▶ When you copy an archive handler on the Data Manager Control screen (to add a new archive handler), you enter a set of three parameters into the Description field to set the:
  - Scan period
  - The name of the work directory
  - The name of the destination directory.
- ▶ Archive handlers do not have an initialization file. The system determines the archive handler's personality from the configurator type (**a** for archive handler) on the Data Manager Configuration screen.

---

**How many archive handlers do you need?**

Normally, you have only one archive handler, but you can use more than one if a single archive handler is not sufficient to meet your organization's archiving needs.

**Example**

You may want one archive handler to handle inbound archiving and another one to handle outbound archiving.

---

(Continued on next page)

---

**Ways to invoke  
an archive  
handler**

You can invoke the archive handler from:

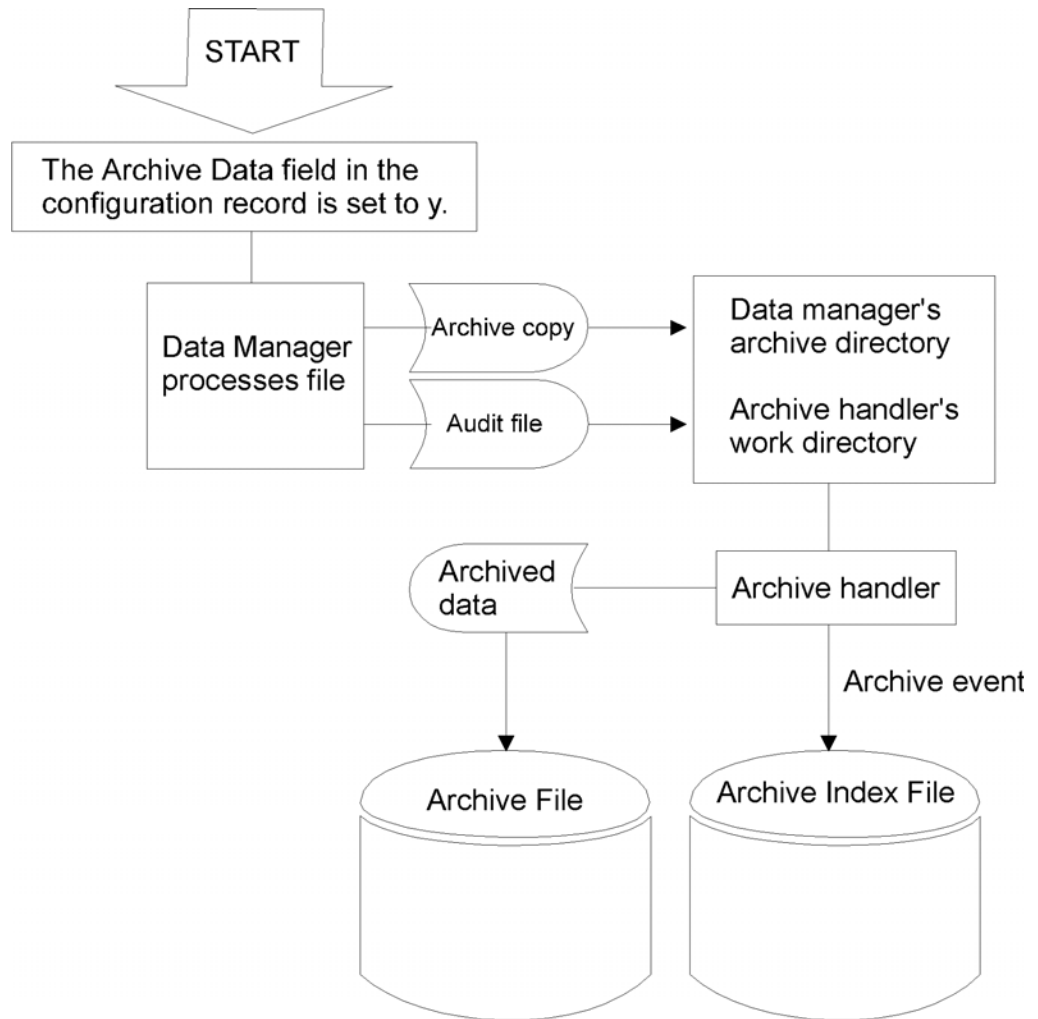
- The command line
- A script
- The Foreground Manager.

In most cases, you should start the archive handler from the Foreground Manager and configure the archive handler so that archiving is a continuous process.

---

## Flow of Data Through Archive Handler

This illustration shows the flow of data through an archive handler.





# The Archiving Process

## Stages in the archiving process

This table describes the archiving process.

Stage	Description
1	As a data manager begins to process a file in its work directory, it checks the configuration record to determine if the file should be archived.
2	If the Archive Data field is set to <b>y</b> , the data manager checks its initialization file to determine where to send the archive files.
3	After processing the file, the data manager moves the following to the archive directory named in its initialization file: <ul style="list-style-type: none"> <li>▶ An archive copy of the document</li> <li>▶ An audit file that contains archiving instructions.</li> </ul>
4	When the number of seconds in its scan period has passed, the archive handler looks in its work directory (the archive directory) for the audit files. Audit files have an underscore ( <b>_</b> ) as a prefix.
5	The archive handler processes the data files in its work directory based on the instructions in the audit files. The audit file directs the archive handler to: <ul style="list-style-type: none"> <li>▶ Load the data files into the archive files</li> <li>▶ Record the detail, including the document reference number, in the archive index.</li> </ul>
6	Once the archive handler has archived a file referred to in the audit file, it deletes the file and makes an entry in the process log to show that the file was archived.
7	After the archive handler has processed all the audit files in the archive directory, it: <ul style="list-style-type: none"> <li>▶ Names the archived data file  <i>&lt;data manager name&gt;.ccyymmddhh</i>            where ccyymmddhh is the date and hour</li> <li>▶ Deletes the audit file and the original archive file.</li> </ul>

# Configuring Your System to Archive Data

## Overview

### Task summary

This table summarizes the tasks you must perform to enable your system to archive data.

Task	Action
1	Configure the data manager's initialization file to archive data. <b>Reference</b> See the <a href="#">How to Configure a Data Manager to Archive Data</a> topic in this chapter.
2	Set the <b>Archive Data</b> field to <b>y</b> in the configuration records of the Trading Partnerships and file names for which you want to archive data. <b>Reference</b> See the <a href="#">How to Set a Configuration Record to Archive Data</a> topic in this chapter.
3	Configure the archive handler. <b>Reference</b> See the <a href="#">How to Configure an Archive Handler</a> topic in this chapter.

## How to Configure a Data Manager to Archive Data

**Introduction** Each data manager handles its own archive data, so you must configure each data manager that you want to archive data.

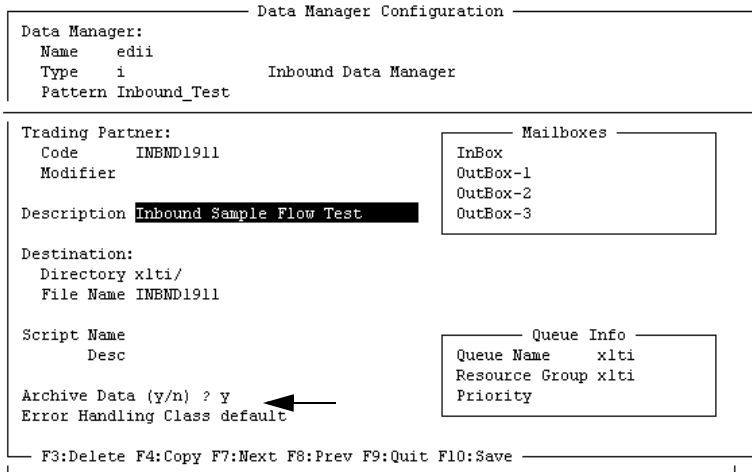
**Procedure** This table describes how to configure a data manager to archive data.

Step	Action	
1	Open the data manager's initialization file.  <b>Reference</b> For instructions, see the <a href="#">How to Access and Edit an Initialization File</a> topic in the <a href="#">Maintaining Initialization Files</a> chapter in this guide.	
2	Locate the Archive Handling subgroup.	
3	Complete the fields.	
	Field	Description
	ARCHIVE_DIRECTORY	The name of the archive directory, which is the archive handler's work directory.  The default is arch.
	ARCHIVE_PERIOD	The number of days that active archived data is to be retained.
	ARCHIVE_ERROR	Do you want error files archived?  <ul style="list-style-type: none"> <li>▶ 0 = No</li> <li>▶ 1 = Yes</li> </ul>
4	Save your changes.	
5	Exit the editor.	

## How to Set a Configuration Record to Archive Data

**Introduction** In addition to setting parameters in the data manager's initialization file, you must also enable archiving in each configuration record for which you want data archived.

**Procedure** Use this procedure to set data archiving in a configuration record.

Step	Action
1	Open the configuration record.  <b>Reference</b> For instructions, see the <a href="#">How to Open a Configuration Record</a> topic in the <a href="#">Working with Configuration Records</a> chapter.
2	Type <b>y</b> into the Archive Data field.   <p>The screenshot shows the 'Data Manager Configuration' screen. It is divided into several sections: 'Data Manager:' with fields for Name (edii), Type (i), and Pattern (Inbound_Test); 'Trading Partner:' with Code (INBND1911), Modifier, and Description (Inbound Sample Flow Test); 'Destination:' with Directory (xlti/) and File Name (INBND1911); 'Script Name' with Desc; 'Mailboxes' with InBox, OutBox-1, OutBox-2, and OutBox-3; and 'Queue Info' with Queue Name (xlti), Resource Group (xlti), and Priority. The 'Archive Data (y/n) ? y' field is highlighted with a black arrow pointing to it.</p>
3	Press F10 to save your changes.

### Setting archiving in multiple records

To set data archiving in several configuration records at once, you can edit a pattern configuration record, using **y** in the Archive Data field, and then meld the modified pattern with Trading Partnership codes or file names.

#### Reference

See the [Using Patterns](#) section in the [Working with Configuration Records](#) chapter of this guide for instructions.

# How to Configure an Archive Handler

**Introduction** Because an archive handler is a data manager, you configure it through the DataMgr option on the host main menu.

**Adding an archive handler** You add the archive handler by configuring the foreground manager, *fmgr* (the parent manager), as you would to add other data managers.

Step	Action
1	Select <b>DataMgr</b> from the host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.
2	Select <b>fmgr</b> , the Foreground Manager.  <pre> Data Manager Control Name A Status T Description ----- Emgr A 1239 F Foreground Manager (IPC Control) alnm n ***** l Async Line Manager ap00 n ***** m Flow: Test Q! flow Source Agent ap01 n ***** m Flow: 'nother fifteen Translate Agent appm n ***** m Application Data Manager appt n ***** x Application Translator Data Manager arch n ***** a -Darch -Aarch -d0 base n ***** u Base Manager Model </pre>
3	Press F5 to configure the Foreground Manager.  <b>System Response</b> Gentran:Server displays the Data Manager Configuration screen. This screen lists the data managers that are child processes of the Foreground Manager.
4	Select an existing archive handler.  <pre> dmc Data Manager Configuration Name A Status T Description ----- alnm n 0 l Async Line Manager appm y 0 m Application Data Manager appt n 0 x Application Translator Data Manager arch n 0 a -Darch -Aarch -d0 base n 0 u Base Manager Model dnld y 0 d UDF Data Manager </pre> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>											
5	Press F4 to copy the archive handler.  <b>System Response</b> Gentran:Server displays the Copy screen.  <pre> Copy _____ Name A   Status   T Description arch n           0 a -Darch -Aarch -d0 _____ F9:Quit  F10:Save _____           </pre>											
6	Type the name of the new archive handler in the Name field.											
7	Complete the remaining fields.											
	<table border="1"> <thead> <tr> <th data-bbox="623 802 834 861"><b>Field</b></th> <th data-bbox="834 802 1424 861"><b>Description</b></th> </tr> </thead> <tbody> <tr> <td data-bbox="623 861 834 951">A</td> <td data-bbox="834 861 1424 951">The flag used to start the archive handler with fmgr: <b>y</b> or <b>n</b>.</td> </tr> <tr> <td data-bbox="623 951 834 1190">T</td> <td data-bbox="834 951 1424 1190">           Personality type. Type the letter <b>a</b> for archive.   <b>Comment</b>            Use the configurator type "a" value exclusively on archive handlers. This value has special meaning to the cl_arch program, which is used to periodically purge the archive files.         </td> </tr> <tr> <td data-bbox="623 1190 834 1518" rowspan="3">Description</td> <td data-bbox="834 1190 1424 1249">List the arguments:</td> </tr> <tr> <td data-bbox="834 1249 1424 1308">-d = The number of seconds in the scan period.</td> </tr> <tr> <td data-bbox="834 1308 1424 1430">-D = The name of the scan directory. The default is <b>arch</b>, but you can specify another directory name if you have more than one archive handler.</td> </tr> <tr> <td data-bbox="834 1430 1424 1518">-A = The destination directory where the archive data files reside. The default is <b>arch</b>.</td> </tr> </tbody> </table>	<b>Field</b>	<b>Description</b>	A	The flag used to start the archive handler with fmgr: <b>y</b> or <b>n</b> .	T	Personality type. Type the letter <b>a</b> for archive.  <b>Comment</b> Use the configurator type "a" value exclusively on archive handlers. This value has special meaning to the cl_arch program, which is used to periodically purge the archive files.	Description	List the arguments:	-d = The number of seconds in the scan period.	-D = The name of the scan directory. The default is <b>arch</b> , but you can specify another directory name if you have more than one archive handler.	-A = The destination directory where the archive data files reside. The default is <b>arch</b> .
	<b>Field</b>	<b>Description</b>										
	A	The flag used to start the archive handler with fmgr: <b>y</b> or <b>n</b> .										
	T	Personality type. Type the letter <b>a</b> for archive.  <b>Comment</b> Use the configurator type "a" value exclusively on archive handlers. This value has special meaning to the cl_arch program, which is used to periodically purge the archive files.										
Description	List the arguments:											
	-d = The number of seconds in the scan period.											
	-D = The name of the scan directory. The default is <b>arch</b> , but you can specify another directory name if you have more than one archive handler.											
-A = The destination directory where the archive data files reside. The default is <b>arch</b> .												
A	The flag used to start the archive handler with fmgr: <b>y</b> or <b>n</b> .											
T	Personality type. Type the letter <b>a</b> for archive.  <b>Comment</b> Use the configurator type "a" value exclusively on archive handlers. This value has special meaning to the cl_arch program, which is used to periodically purge the archive files.											
Description	List the arguments:											
	-d = The number of seconds in the scan period.											
	-D = The name of the scan directory. The default is <b>arch</b> , but you can specify another directory name if you have more than one archive handler.											
-A = The destination directory where the archive data files reside. The default is <b>arch</b> .												
8	Press F10 to save the new archive handler.											

# Managing Your Archives

## Overview

### Introduction

To manage your disk storage space for archived data effectively, you should periodically purge old archived files from your host machine and move them to long term storage media, such as tape. You can purge archived files:

- Manually by running the **cl\_arch** command from the command line
- Automatically by configuring the *longterm* script to run on a schedule.

### Components of archive management

Gentran:Server provides a number of components that you can combine to handle your long-term archives automatically.

This table describes the components of archive management.

Component	Description
<i>longterm</i> script	A Gentran:Server script designed to remove old archived files.
<b>cl_arch</b> program	The Gentran:Server program that removes archived files.
archive period	The number of days that archived data is retained. The program <b>cl_arch</b> deletes archived entries older than this number.
Permanent Schedule	The Gentran:Server feature that enables you to run scripts on a schedule.

### The *longterm* script

Gentran:Server provides a script named *longterm* to help you:

- Invoke a purge program, **cl\_arch**
- Move old archived files to long term storage with the **cpio** command.

To set the number of days to keep archived data, you can use the EDI\_ARCHIST environment variable in the *longterm* script.

(Continued on next page)

## The cl\_arch Program

---

**Function** The `cl_arch` program removes archived files from the archives. You can run `cl_arch` from the command line or from a script that you run on a schedule.

---

**Archive period** The archive period is the number of days that Gentran:Server keeps archived data. You can set this number in these places:

- The `-a <days>` argument if you run **cl\_arch** from the command line
- The `EDI_ARCHIST` environment variable
- The `ARCHIVE_PERIOD` parameter of each data manager's initialization file.

---

**Order of precedence** This is the order Gentran:Server uses to determine the archive period when `cl_arch` is invoked:

- The number of days set in the **cl\_arch** command line `-a <days>` argument
- The number of days set in the `ARCHIVE_PERIOD` parameter in the data manager's initialization file
- The number of days set in the `EDI_ARCHIST` environment variable
- 90 days as the default archive period.

---

**Setting the archive period** This table describes the methods of setting the active archive period.

For this method...	See...
Setting the data manager's <code>ARCHIVE_PERIOD</code> parameter	<a href="#">How to Configure a Data Manager to Archive Data</a> topic in this chapter.
Setting the <code>EDI_ARCHIST</code> environment variable in the environment	The <a href="#">Environment Variables</a> chapter in the <i>Gentran:Server for UNIX and Workstation Technical Reference Guide</i> .
Editing the <i>longterm</i> script to include the <code>EDI_ARCHIST</code> environment variable	The <a href="#">How to Edit the Longterm Script</a> topic.
Setting the <code>-a</code> parameter when you run <b>cl_arch</b> from the command line	The <a href="#">How to Manually Purge Your Archives</a> topic.

---



## The *longterm* Script

### Introduction

Gentran:Server provides a script named *longterm* to help you:

- ▶ Invoke the archive purge program, **cl\_arch**
- ▶ Move old archived files to long-term storage with the **cpio** command.

To set the number of days to keep archived data, you can use the EDI\_ARCHIST environment variable in the *longterm* script.

### Using the Permanent Schedule

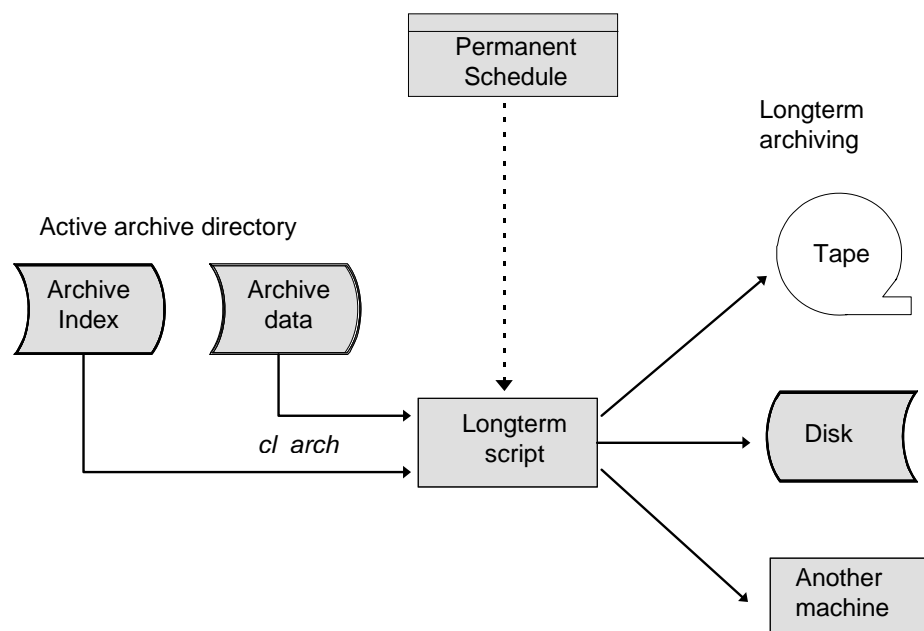
To process the *longterm* script on a schedule, add the script to Gentran:Server's Permanent Schedule. Schedule this process to meet the particular needs of your organization. If your organization does not have specific requirements, run the script once a day.

### Reference

For instructions, see the [Running Scripts on a Schedule](#) section in the [Running Scripts](#) chapter of this guide.

### Model for long-term archiving

This illustration shows one way to combine Gentran:Server components to handle long-term archives.



(Continued on next page)

**Stages in the  
long-term  
archive model**

---

This table describes the stages in the long-term archive model.

Stage	Description
1	The Permanent Schedule starts the <i>longterm</i> script.
2	The <i>longterm</i> script moves the old archived data to a long-term storage device, which can be a: <ul style="list-style-type: none"><li>▶ Tape</li><li>▶ Disk</li><li>▶ Machine.</li></ul>
3	The <i>longterm</i> script starts the <b>cl_arch</b> program.
4	The <b>cl_arch</b> program removes the old archived data.

---

# How to Edit the Longterm Script

---

## Introduction

You can edit the *longterm* script to specify the destination for your long-term archives. You can also specify devices, change paths, and add processing options.

---

## What to modify

Here are some items that you may need to modify when you edit the *longterm* script:

- ▶ The EDI\_ROOT and EDIARCHDIR variables
- ▶ The **cpio** command line options. This command contains the path to the long-term storage destination.
- ▶ Device names. You will need to:
  - Create a lock file that contains a device name

### Example

```
echo "/dev/devicename" > $EDI_ROOT/LOCKS/DEV1
```

- Create a type 2 lock in the *longterm* script.

- ▶ The **cl\_arch** command line parameters. This command removes archived files from the archives.

---

## Example cpio command

This is an example of using the cpio command to copy all the files in the archive directory to the device.

```
find $EDIARCHDIR -print | cpio -oc > $DEVICE
```

---

## Example device name

This is an example of using a type 2 lock for a device in the *longterm* script.

```
:type2lock
```

```
2 ./LOCKS/DEV1 DEVICE
```

### Reference

See the [Working with Scripts](#) chapter in this guide for information about using LOCKS.

---

(Continued on next page)

**Example cl\_arch command**

This is an example of using the `cl_arch` command to delete archived files older than 60 days.

```
cl_arch -a 60
```

**Reference**

See the [How to Manually Purge Your Archives](#) topic in this chapter for a list of `cl_arch` parameters.

**Editing the longterm script**

Use this procedure to edit the *longterm* script.

Step	Action
1	<p>Select <b>Script</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays a list of the scripts in the script library and a list of function choices.</p> <pre data-bbox="646 1003 1404 1354"> 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           </pre>
2	<p>Select <b>longterm</b> and then press F5.</p> <p><b>System Response</b> Gentran:Server displays a Modify screen with additional function keys.</p> <pre data-bbox="662 1591 1383 1690"> Modify ----- Longterm      Longterm Archive Script ----- F4:EditMail F5:EditScr F9:Quit F10:SaveDesc           </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	Press F5 to edit the script.  <b>System Response</b> Gentran:Server opens <i>longterm</i> script. The file is opened in the editor that your organization set in the \$EDITOR variable. The editor defaults to vi if the EDITOR variable is not set.
4	Use the standard editor keys to move the cursor and edit the script.  <b>Reference</b> See the documentation for the specific editor you are using.
5	Save the script.
6	Exit the editor.

---

# How to Manually Purge Your Archives

---

## Introduction

If you prefer to control your archives manually instead of configuring your system to purge old archived files automatically, you can run the `cl_arch` purge program from the command line. To run `cl_arch` from the command line, you must specify the criteria for the files you want removed from the active archives.

When you run `cl_arch` from the command line, you must specify the archive period for the files you want removed from the archives. If you do not use an archive period with `cl_arch`, Gentran:Server displays the `cl_arch` invocation screen.

You can use `cl_arch` to purge archived files for a specific data manager (intelligent agent). The command also has an option to retain the transaction register records, which the system normally deletes along with the archived files.

---

## Command line format

The command line format is one of the following:

```
cl_arch -a<days>
```

```
cl_arch -a
```

```
cl_arch (-B<delete from date> -E<delete to date>)
```

---

## Retain transaction records

If you want to retain the transaction records, include the `-N` option.

### Example

```
cl_arch -a15 -N
```

---

## Purge archives for one data manager

If you want to delete archived files for a specific data manager, use the `-D` with the name of the data manager, which is `in32` in this example.

### Example

```
cl_arch -a1 -Din32
```

---

(Continued on next page)

**Command line options**

This table describes the command line options.

Option	Description
-a<days>	The number of days in the archive period. The program deletes any archive entries older than this.
-a	If you do not specify a number of days, <b>cl_arch</b> uses the number of days set in the data manager's ARCHIVE_PERIOD parameter. If the ARCHIVE_PERIOD parameter does not contain a value, <b>cl_arch</b> uses the value set in the EDI_ARCHIST environment variable. If neither the ARCHIVE_PERIOD parameter nor the EDI_ARCHIST environment variable is set, <b>cl_arch</b> uses 90 days as the default value.
-B<delete from date>	The first (earliest) date in a date range of entries to be deleted. The date format is [CC]YYMMDD.
-E<delete to date>	The most recent date in a date range of entries to be deleted. Deletion includes up to, but does not include the delete_to_date. The date format is [CC]YYMMDD.  <b>Comment</b> The <delete to date> should be more recent than the <delete from date>.
N	Do not delete transaction register records.
D<agent_name>	Indicates that archived data is to be purged for only the named intelligent agent (data manager).

**CAUTION**

The century in the argument string is optional. If you do not enter the century and YY is greater than 50, **cl\_arch** interprets YYMMDD to be 19YYMMDD. If YY is less than 50, **cl\_arch** interprets YYMMDD to be 20YYMMDD.

(Continued on next page)

---

**Command line examples**

Here are three examples of using `cl_arch` on the command line.

**Example 1**

```
cl_arch -a60
```

In this example, `cl_arch` deletes archive entries older than 60 days.

**Example 2**

```
cl_arch -a
```

In this example, `cl_arch` uses the value set in the `ARCHIVE_PERIOD` parameter of the data manager's initialization file, if it contains a value. If it does not contain a value, `cl_arch` uses the value set in the `EDI_ARCHIST` environment variable. If `EDI_ARCHIST` is not set, `cl_arch` deletes archive entries older than 90 days.

**Example 3**

```
cl_arch -B19970115 -E19970216
```

In this example, `cl_arch` deletes archive entries with dates of January 15, 1997 through February 15, 1997.

---



# Retrieving Archived Files

## Overview

---

### Introduction

You can retrieve files from:

- The current archive
- Long-term archive indexes or files that have been restored to a directory, such as the temporary directory.

---

### Using the archive index

Each data manager has its own archive index. The archive index provides a detailed record of the files on archive. You can access the information in this index by entering search criteria into the Archive Query screen. The search results are displayed on the Archive List screen. You select the archived data files you want to retrieve and start the retrieval process from the Archive List screen.

---

### Information you need to retrieve files

To retrieve files from archive, you must know the name of the:

- Data manager that processed the file
- Archive directory that contains the data manager's files.

---

### In this section

This section:

- Describes the retrieval process
  - Describes the Gentrans:Server screens that you use to view and retrieve archived files
  - Explains how to select and retrieve archived files
  - Explains how to view the retrieval log, which is the log file produced by the retrieval process.
-

## The Retrieval Process

---

**Description** The `rtv_arc` program is a background retrieval process that pulls selected entries from the archive directory and deposits them into a directory that you specify.

The process starts when you mark the files you want to retrieve on the Archive List screen and then press F2 to run the `rtv_arc` program.

**Reference**

You can also run `rtv_arc` from the command line. See the *Gentran:Server for UNIX and Workstation Technical Reference Guide* for information about command line programs.

---

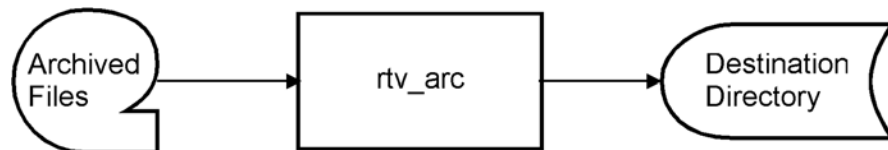
**How retrieved files are named** The name of a retrieved file is the name of the data manager that archived the file plus a unique numeric suffix that Gentran:Server generates.

---

**rtv\_arc.l file** The process generates the `rtv_arc.l` log file, which contains a description of the selected entries, where they were sent, and what they are called.

---

**Process diagram** This illustration shows the `rtv_arc` retrieval process.



## The Archive Query Screen

### Introduction

You specify the search criteria for the archived files you want to retrieve and start the search on the Archive Query screen. The Archive List screen displays the file names that match the search criteria.

### Reference

See [The Archive List Screen](#) topic for information.

### Archive Query screen

This illustration shows an example of the Archive Query screen.

```

Archive Query
Data Manager Name appm
Archive Directory
arch
New Location
arch
Starting Date CCYYMMDD 19990315 Ending Date CCYYMMDD 19990315
TP Code Mod
Doc Ref Num
F7:Log F9:Exit F10:Search
  
```

### Archive Query screen fields and functions

This table lists the fields of the Archive Query screen and their functions.

Field	Function
Data Manager Name	Defines the name of the data manager that processed the files.
Archive Directory	Defines the name of the directory that contains the archived files and archive index from which you want to retrieve files.
New Location	Defines the name of the directory that you want to hold the retrieved data.  <b>CAUTION</b> <b>If you have the EDI_ARCH_NEWLOC environment variable set, the default value for this field is the directory set in that variable.</b>
Starting Date	Defines the starting search date in CCYYMMDD or YYMMDD format.  <i>(Continued on next page)</i>

<b>(Contd) Field</b>	<b>Function</b>
Ending Date	Defines the ending search date in CCYYMMDD or YYMMDD format.
TP Code	Defines the Trading Partnership code of the archived data you want to retrieve. This field accepts partial values.
Mod	Defines the modifier to the Trading Partnership code, if any. This field accepts partial values.
Doc Ref Num	Defines the document reference number of the archived data you want to retrieve. This field accepts partial values.

### Archive Query screen function keys

This table describes the function keys on the Archive Query screen and their functions.

<b>Key</b>	<b>Function</b>
F7	Displays the <code>rtv_arc</code> log file.
F9	Exits the screen.
F10	Starts the search.

## The Archive List Screen

**Introduction** The Archive List screen displays the results of the search you started from the Archive Query screen.

Use this screen to:

- ▶ Select the archived files that you want to retrieve
- ▶ Start the retrieval process.

### Reference

See [The Archive Query Screen](#) topic in this chapter for information.

### Archive List screen

This illustration shows an example of the Archive List screen.

```

Archive List: arch/edii_arch
M Date Time TP Code Document Reference Number
980105 194020 INBND210 999123457
980105 194101 INBND837 P03F8A
980105 171425 INBND850 02431200
F2:Run F3:Clear F4:Copy & Remove TR Entry F5:Copy F7:Next F8:Prev F9:Exit

```

### Archive List screen fields and functions

This table lists the fields of the Archive List screen and their functions.

Field	Function
M	Selects the entry for retrieval.
Date	Displays the date that the data manager processed the entry.
Time	Displays the time the data manager processed the entry.

(Continued on next page)

<b>(Contd) Field</b>	<b>Function</b>
TP Code	Displays the Trading Partnership code or file name of the entry.
Document Reference Number	Displays the document reference number of the entry.

**Archive List  
screen function  
keys**

This table describes the function keys on the Archive List screen and their functions.

<b>Key</b>	<b>Function</b>
F2	Starts the retrieval process.
F3	Deselects the entry for retrieval.
F4	Selects entry for retrieval and marks the entry with a "r" to remove it from the transaction register as well as retrieve it.
F5	Selects entry for retrieval and marks the entry with a "c" to indicate the entry is selected.
F7	Displays the next group of 500 entries.
F8	Displays the previous group of 500 entries.
F9	Exits to the Archive Query screen.

# How to Select and Retrieve Archived Files

**Introduction** This topic explains how to use the Archive Query screen to search for the files you want to retrieve. It also explains how to use the Archive List screen to select specific files and start the retrieval process.

**Before you begin** To select files for retrieval, you need to know the:

- Name of the data manager that processed the document
- Name of the directory that contains the archived files.

**Retrieval options** When you mark the files you want to retrieve, you have two options:

- Retrieve the file and retain the transaction register entry
- Retrieve the file and remove the transaction register entry

The purpose of removing the entry from the transaction register is to prevent the system from viewing the entry you restored as a duplicate if the file passes through Gentran:Server again.

**Procedure** Use this procedure to search for archived files you want to retrieve, select specific files, and start the retrieval process.

Step	Action
1	Select <b>DataMgr</b> from the host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.
2	Select the data manager that processed the data.  <span style="color: red;">(Continued on next page)</span>

<b>(Contd) Step</b>	<b>Action</b>
3	Press F2 to display the <b>Archive Query</b> screen. <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> Archive Query Data Manager Name  appm Archive Directory arch New Location arch Starting Date CCYYMMDD 19990315 Ending Date CCYYMMDD 19990315 TP Code                               Mod Doc Ref Num F7:Log F9:Exit F10:Search </pre> </div>
4	Type the search criteria into the fields.
5	Press F10 to start the search. <p><b>System Response</b> If Gentran:Server finds entries matching the search criteria you entered, it displays the Archive List screen.</p>
6	Use the F7 and F8 keys to locate the archived files you want to retrieve. <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>		
7	Mark the files you want to retrieve.  Use this decision table to determine your action.		
	<b>IF you want to...</b>	<b>THEN select the file and press...</b>	<b>And Gentran:Server...</b>
	Retrieve the file and remove the transaction register entry	F4	Marks the file with an r.
	Retrieve the file and retain the transaction register entry	F5	Marks with file with a c.
<p><b>CAUTION</b></p> <p><b>These marks are valid for the current session only. If you exit the screen before completing Step 8 (running the <code>rtv_arc</code> process), you must mark your entries again the next time you access this screen. (The selections you marked in this step will appear to be marked, but the system does not recognize them.)</b></p>			
8	<p>Press F2 to run the <code>rtv_arc</code> background process, which retrieves the selected entries and moves them to the directory you specified in the New Location field.</p> <p><b>CAUTION</b></p> <p><b>The process begins one minute after you press F2 and can take several minutes to run. Once the operation is completed, Gentran:Server replaces the lowercase r and c marks with uppercase R and C. This is your confirmation that the <code>rtv_arc</code> process took place.</b></p>		

## The Retrieval Log

---

**Introduction** The *rtv\_arc* retrieval process produces a log file that you can use to confirm that the system retrieved the files you selected and deposited them into the correct directory.

---

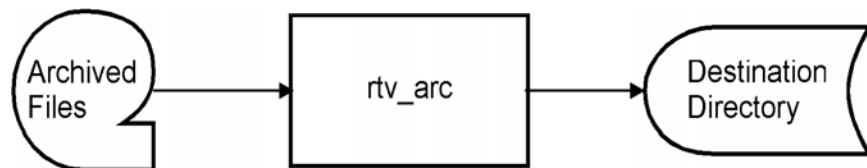
**Log name** The name of the retrieval log is *rtv\_arc.l*.

---

**rtv\_arc log contents** The retrieval log describes the archived files that the *rtv\_arc* process retrieved, lists where they were sent, and indicates the new file name.

---

**Example retrieval log** This is an example of the retrieval log.



## How to Display the Retrieval Log

### Displaying the retrieval log

Use this procedure to display the *rtv\_arc.l* retrieval log.

Step	Action
1	Select <b>DataMgr</b> from the host main menu.  <b>System Response</b> Gentran:Server displays the Data Manager Control screen.
2	Select the name of the data manager that originally archived the data.  <pre> Data Manager Control ----- Name A      Status  T Description ----- fmgr A ***** F Foreground Manager (IPC Control) alnm n ***** l Async Line Manager appm y ***** m Application Data Manager appt n ***** x Application Translator Data Manager arch n ***** a -Darch -Aarch -d0 base n ***** u Base Manager Model dl00 n ***** x Flow: stdappl Delivery Agent dl01 n ***** x Flow: stdapp2 Delivery Agent dl02 n ***** x Flow: edi_xl2 Delivery Agent dnld y ***** d UDF Data Manager <b>edii y ***** i Inbound Data Manager</b> edio y ***** i Outbound Data Manager file n ***** f File Data Manager hcmd n ***** h Host Command Card Data Manager inbd n ***** i Inbound Data Manager lnmn n ***** l Line Manager pr00 n ***** x Flow: stdappl Processing Agent ----- F2:Arch F4:EditMail F5:Config F6:Stat F7:Log F9:Quit </pre>
3	Press F2 to display the Archive Query screen.
4	Press F7 to display the <i>rtv_arc</i> log.
5	When you are finished with the log, press F9 to exit.



---

# Working with Scripts

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction . . . . .	3
	▶ Gentran:Server Scripts . . . . .	5
	▶ The Script Manager . . . . .	6
	▶ Ways to Use Scripts . . . . .	7
	<b>Parts of a Gentran:Server Script</b>	
	▶ Introduction . . . . .	8
	▶ ENVIRON Group . . . . .	11
	▶ LOCKS Group . . . . .	13
	▶ DATA Group . . . . .	17
	▶ DPROCS Group . . . . .	19
	▶ STEPS Group . . . . .	20
	▶ PROCS Group . . . . .	23
	▶ RESULTS Group . . . . .	27
	<b>Creating Gentran:Server Scripts</b>	
	▶ Overview . . . . .	30
	▶ The Flow of Work . . . . .	32
	▶ Guidelines for Writing Scripts . . . . .	33
	▶ The Script Maintenance Screen . . . . .	36
	<b>Procedures</b>	
	▶ 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
	<b>Working With Translation Scripts</b>	
	▶ Translation Scripts . . . . .	46
	▶ How to Create a Translation Script . . . . .	47
	<b>Maintaining Gentran:Server Scripts</b>	
	▶ How to Edit a Script with the Script Editor . . . . .	50

- ▶ How to Delete a 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 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.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
script directory	The directory that contains all Gentran:Server scripts.
script editor	The default editor that Gentran:Server calls when you create or edit a script.
script library	A file that lists and describes all the Gentran:Server scripts that have been added to the file.
Script Manager	The Gentran:Server program that directs the script interpreter to execute the commands in a script.
translation script	A special Gentran:Server for UNIX script associated with and invoked by a translation data manager. The script calls the translator, <b>lftran</b> , 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 Gentran:Server script operation.

---



# Gentran:Server Scripts

- 
- Definition**    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 Gentran:Server what to do.

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

- 
- Script language**    Gentran:Server scripts are written in a Sterling Commerce 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  
Gentran:Server  
scripts**    Every 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**    Gentran:Server scripts enable you to control processes and carry out commands automatically.

**Example**

You can have a data manager invoke a script after it processes data.

A script can invoke another script.

- 
- Translation  
scripts**    **Translation scripts** are special 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 Gentran:Server script
- A UNIX shell script
- The Permanent Schedule.

---

**Handles script errors** The Gentran:Server Script Manager returns a non-zero result code if a 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

---

<b>Introduction</b>	You can use Gentran:Server scripts to control many processes and commands. This topic suggests a number of ways to use Gentran:Server scripts.
<b>Start and terminate programs and processes</b>	Use Gentran:Server scripts to invoke data managers, the translator, programs, other Gentran:Server scripts, and shell scripts.
<b>Communicate with hosts</b>	Create a Gentran:Server script to pull host data through an Ethernet or other host connection.
<b>Communicate with Value Added Networks</b>	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 Gentran:Server scripts to invoke the communication scripts provided in the Communications Toolkit.
<b>Manage non-shareable resources</b>	You can use scripts to control access to a non-shareable resource such as a modem.
<b>Send notification of an event</b>	You can have scripts automatically generate a notification of an event, such as a communications failure.
<b>Handle files</b>	You can use 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 Gentran:Server Script

## Introduction

### Common parts

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 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 Sterling Commerce-supplied model scripts use these delimiters. Changing these delimiters can cause your scripts to fail or operate incorrectly.**

(Continued on next page)

**Sections** A 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.

(Continued on next page)

**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.  <b>Example</b> 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.  <b>Example</b> 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** 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.  <b>Example</b> This is the label name for the VAN variables:  :VAN_var
varname=var_value	The variable and the value to which it is set.  <b>Example</b> In this example, the variable errorDir, which represents the error directory, is set to commerror.  errorDir=commerror

(Continued on next page)

### 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 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 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.

---

(Continued on next page)

## How the lock types are used

This table describes how each lock type is used.

Lock	Use
Type 1	<p>Used to lock processes.</p> <p><b>Format</b> :type1lock 1 lock1_filename</p> <p><b>Example</b> Use a type 1 lock to lock a script because you want the script to finish running before another process restarts it.</p>
Type 2	<p>Used to lock devices such as modems, tape drives, and diskette drives.</p> <p><b>Format</b> :type2lock 2 lock2_filename &lt;variable&gt;</p> <p>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.</p> <p><b>Example</b> :type2lock 2 lock2_filename DEVICE</p> <p>Use a type 2 lock to lock a modem because you want to prevent another Gentran:Server script from using the same device.</p>

## 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
```

(Continued on next page)

## 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.  <b>Example</b> 2 file2_name DEVICE

(Continued on next page)

### LOCKS group example

This is an example of the LOCKS group.

```
!LOCKS -d120 -r5
:modems
## Lock the CLEO modem lines from other 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 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 Gentran:Server scripts.
2 ./LOCKS/MODEM1 MODEM
2 ./LOCKS/MODEM2 MODEM
2 ./LOCKS/MODEM3 MODEM
```

#### Reference

The 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 ( <b>build</b> ) 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
```

(Continued on next page)

### 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.

(Continued on next page)



**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.
```

(Continued on next page)

```
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
```

---

(Continued on next page)

### 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	<p>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.</p> <p>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.</p> <p><b>CAUTION</b></p> <p><b>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.</b></p>

### 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.

(Continued on next page)

<b>(Contd) Reserved Word or Function</b>	<b>Function</b>
end	The part of the PROCS action line syntax that tells the operating system to stop the script.
build(DPROC_label,file_name)	Assembles the file_name according to the list under the DPROC_label.  <b>Example</b> if build(build_label1,newfile) then...
release(LOCKS_label)	Unlocks the file or resource locked under LOCKS_label.  <b>Example</b> if release(modem_pool) then...
compare	Compares strings or system variables. If the values are equal, the compare is successful.  <b>Example</b> if compare(\$envar,"string value") then... or if compare(\$envar1,\$envar2) then...  <b>Note</b> In the above example, "string value" represents a literal value.
putenv(ENV_VAR=some_value)	Sets the environment variable ENV_VAR to some_value.  <b>Example</b> 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.**

---

(Continued on next page)

**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.  <b>CAUTION</b> <b>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.</b>

(Continued on next page)

### 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
```

(Continued on next page)



```
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 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, Sterling Commerce can provide special remote services (Communications Support within Customer Support) to develop the script for you.

---

### Permissions

The file permissions for a script are normally `rxw rxw 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 Gentran:Server because script files are ASCII text.

---

(Continued on next page)

---

### The `./script` directory

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

- ▶ When you use the editor accessed through Gentran:Server, 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 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, 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	<p>Break down the job into steps.</p> <p><b>Example</b></p> <ol style="list-style-type: none"> <li>1. Find a file to process.</li> <li>2. Translate the file.</li> <li>3. Transfer the file.</li> </ol>
4	Break the steps into executable commands.
5	Identify error conditions.
6	Create any directories or files that the script will use.
7	<p>Use the shell to test commands and error conditions.</p> <p><b>Comment</b></p> <p>You can use CTRL+A to access the shell.</p>
8	<p>Write the script.</p> <p>You can use the script editor in the Gentran:Server <b>Script Maintenance</b> facility, or write the script with another editor and then move the script into the <i>./script</i> directory.</p> <p><b>Reference</b></p> <p>See your UNIX documentation for UNIX commands to use in your script.</p> <p><b>Comment</b></p> <p>If you use another editor, make sure that you:</p> <ul style="list-style-type: none"> <li>▶ Add the <i>.scr</i> suffix to the script file name</li> <li>▶ Use the Script Maintenance screen to add the script to the script library.</li> </ul>
9	<p>Test the script.</p> <p>Execute it from the Script Maintenance screen, or run the Script Manager from the shell with the command <b>smgr -s&lt;scriptname&gt;</b>. (Omit the <i>.scr</i> suffix in the file name. DO NOT leave a space between -s and the file name.)</p>

# 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.

---

(Continued on next page)

---

**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
```

---

(Continued on next page)

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

### 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: <ul style="list-style-type: none"> <li>▶ Inactv means the script is not running</li> <li>▶ Active means that the script is running.</li> </ul>
Description	Describes the script.

(Continued on next page)



**Script  
Maintenance  
screen function  
keys**

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

<b>Key</b>	<b>Function</b>
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, Gentran:Server uses **vi** as the default editor.

#### Reference

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

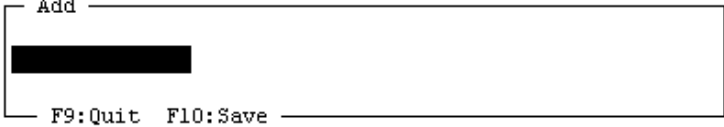

#### WARNING

**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	<p>Select <b>Script</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Script Maintenance screen, which lists all the scripts in the script library.</p> <pre> 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           </pre> <p style="text-align: right;">(Continued on next page)</p>

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

(Continued on next page)

<b>(Contd) Step</b>	<b>Action</b>
9	Exit the editor.
10	Does your script include LOCKS? <ul style="list-style-type: none"><li data-bbox="630 489 1382 552">▶ If YES, see the <a href="#">How to Add Lock File Names to the LOCKS Directory</a> topic.</li><li data-bbox="630 562 948 594">▶ If NO, you are finished.</li></ul>

---

## How to Add a Script with Another Editor

### Introduction

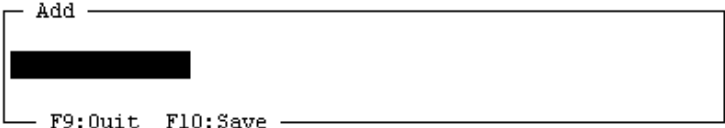
When you create a script with an editor other than the script editor, you must move or copy the script to the `./script` directory and add it to the Gentran:Server script library list.

### 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 <code>.scr</code> suffix and save it.
3	Copy or move the script into the <code>./script</code> directory.
4	Start Gentran:Server and access the host menu.
5	Select <b>Script</b> from the host main menu.  <b>System Response</b> Gentran:Server displays the Script Maintenance screen.
6	Press F2 to add the script to the library list.  <b>System Response</b> Gentran:Server displays a screen for the script name and description.   <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
7	Type the script name (without the .scr suffix) in the selected field. Type the script's description in the second field.  <b>Note</b> 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? <ul style="list-style-type: none"><li>▶ If YES, see the <a href="#">How to Add Lock File Names to the LOCKS Directory</a>.</li><li>▶ If NO, you are finished.</li></ul>

---

## 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, 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	<p>Select <b>Script</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays a list of the scripts in the script library and a list of function choices.</p> <pre> 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 </pre>
2	<p>Select the name of the script you want to copy.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Press F4 to copy the script.</p> <p><b>System Response</b> Gentran:Server displays a Copy screen.</p> <pre data-bbox="662 533 1393 659">Copy from:  appt_xltr _____            New Script Name:  _____            F9:Quit  F10:Save _____</pre>
4	Type the name of the new script, without the .scr suffix, into the <b>Script</b> 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.	<code>touch LOCKS/MODEM_10</code> 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.	<code>echo "/dev/tty0d" &gt; ./LOCKS/MODEM_1</code> 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 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 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	<p>Select <b>Script</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays a list of the scripts in the script library and a list of function choices.</p>
2	<p>Select the translation script you want to copy.</p> <pre> 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           </pre>
3	<p>Press F4 to copy the script.</p> <p><b>System Response</b> Gentran:Server displays a Copy screen.</p> <pre> Copy from:  appt_xltr ----- New Script Name: ----- F9:Quit  F10:Save           </pre> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
4	<p>Type the name of the new script, without the .scr suffix, into the New Script Name field.</p> <pre> Copy from:  appt_xltr  _____ New Script Name:  ap01_xltr F9:Quit  F10:Save  _____ </pre>
5	<p>Press F10 to save the new script.</p> <p><b>System Response</b> Gentran:Server adds the script name to the Script Maintenance screen.</p>
6	<p>Select the name of the new translation script and press F5.</p> <p><b>System Response</b> Gentran:Server displays a Modify screen with additional function keys.</p> <pre> Modify  _____ ap01_xltr  Outbnd App Translation Script F4:EditMail  F5:EditScr  F9:Quit  F10:SaveDesc  _____ </pre>
7	<p>Modify the script's description and then press F10 to save the description.</p>
8	<p>Press F5 to edit the new script.</p> <p><b>System Response</b> Gentran:Server displays the script editor.</p>
9	<p>Use the standard editor keys to edit the script.</p> <p><b>Comment</b> Here are some items you may want to modify:</p> <ul style="list-style-type: none"> <li>▶ Name of the translation data manager that invokes the script (dmName line)</li> <li>▶ Translation options in the <b>lftran</b> command line</li> <li>▶ Addition of the <b>xlld</b> command to run the Life Cycle update program from the translation script.</li> <li>▶ Messages in the RESULTS section.</li> </ul> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
10	Press F10 to save your changes.
11	<p>Check all the parameter settings in the translation data manager's initialization file. You may need to change the values for these parameters:</p> <ul style="list-style-type: none"><li>▶ XL_MODEL_SCR</li><li>▶ XL_RUN_DIR</li><li>▶ WORK_DIRECTORY</li></ul> <p><b>Reference</b> See the <a href="#">Maintaining Initialization Files</a> chapter in this guide for instructions on editing initialization files.</p>

---

# Maintaining 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, 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	<p>Select <b>Script</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Script Maintenance screen.</p> <pre data-bbox="643 1163 1411 1520"> 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 </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>						
2	<p>Select the name of the script you want to edit and then press F5.</p> <p><b>System Response</b> Gentran:Server displays a Modify screen with additional function keys.</p> <pre data-bbox="646 562 1421 674"> Modify _____ ap01_xltr          Outbnd App Translation Script F4:EditMail  F5:EditScr  F9:Quit  F10:SaveDesc _____ </pre>						
3	<p>Use this table to determine your next action.</p> <table border="1" data-bbox="623 768 1427 1272"> <thead> <tr> <th data-bbox="623 768 857 856"><b>IF you want to change...</b></th> <th data-bbox="857 768 1427 856"><b>THEN...</b></th> </tr> </thead> <tbody> <tr> <td data-bbox="623 856 857 1100">The description of the script</td> <td data-bbox="857 856 1427 1100">           Move the cursor to the character or characters you want to change.           <ul style="list-style-type: none"> <li>▶ Type the new characters over the existing ones. To erase a character or insert a blank space, press the space bar.</li> <li>▶ Press F10 to save the changes.</li> </ul> </td> </tr> <tr> <td data-bbox="623 1100 857 1272">The contents of the script</td> <td data-bbox="857 1100 1427 1272">           Press F5 to access the script editor and continue with Step 4.           <p><b>System Response</b> Gentran:Server displays the script editor.</p> </td> </tr> </tbody> </table>	<b>IF you want to change...</b>	<b>THEN...</b>	The description of the script	Move the cursor to the character or characters you want to change. <ul style="list-style-type: none"> <li>▶ Type the new characters over the existing ones. To erase a character or insert a blank space, press the space bar.</li> <li>▶ Press F10 to save the changes.</li> </ul>	The contents of the script	Press F5 to access the script editor and continue with Step 4. <p><b>System Response</b> Gentran:Server displays the script editor.</p>
<b>IF you want to change...</b>	<b>THEN...</b>						
The description of the script	Move the cursor to the character or characters you want to change. <ul style="list-style-type: none"> <li>▶ Type the new characters over the existing ones. To erase a character or insert a blank space, press the space bar.</li> <li>▶ Press F10 to save the changes.</li> </ul>						
The contents of the script	Press F5 to access the script editor and continue with Step 4. <p><b>System Response</b> Gentran:Server displays the script editor.</p>						
4	Use the standard editor keys to edit the script.						
5	Press F10 to save your changes.						
6	<p>Does your script include LOCKS?</p> <ul style="list-style-type: none"> <li>▶ If YES, see the <a href="#">How to Add a Script with Another Editor</a> topic to add the lock names to the LOCKS directory.</li> <li>▶ If NO, you are finished.</li> </ul>						

## How to Delete a 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	<p>Select <b>Script</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Script Maintenance screen.</p> <pre> 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 </pre>
2	<p>Select the script name you want to delete.</p> <p style="text-align: right;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
3	<p>Press F3 to delete the script.</p> <p><b>System Response</b> Gentran:Server displays a confirmation prompt for deleting the script name.</p> <pre> Delete: appg_xltr _____ Do you want to delete this entry? █ F9:Quit _____ </pre>
4	<p>Type <b>y</b> at the prompt to delete the script name.</p> <p><b>System Response</b> Gentran:Server displays a confirmation prompt for deleting the script.</p> <pre> _____ Confirmation _____ Delete the actual Script File (y/n) ? </pre>
5	<p>Do you want to delete the script?</p> <ul style="list-style-type: none"> <li>▶ If YES, enter <b>y</b> for at the prompt</li> <li>▶ If NO, enter <b>n</b> at the prompt</li> </ul>



# How to Add or Edit a UNIX Mail Script

## Introduction

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

Step	Action
1	<p>Select <b>Script</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Script Maintenance screen.</p> <pre> 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 </pre>
2	<p>Select the script associated with the UNIX mail script that you want to add or edit.</p>
3	<p>Press F5 to edit the script.</p> <p><b>System Response</b> Gentran:Server displays a Modify screen with additional function keys.</p> <pre> Modify ----- ap01_xltr      Outbnd App Translation Script ----- F4:EditMail F5:EditScr F9:Quit F10:SaveDesc </pre> <p style="text-align: right;">(Continued on next page)</p>



# 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.  <b>Example</b>  <code>rm ./mail_proc/&lt;mail_script_name&gt;</code>

---



---

# Running Scripts

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	2
	<b>Running Scripts</b>	
	▶ Overview .....	3
	▶ How to Have a Data Manager Run a Script after Processing .....	5
	▶ How to Have a Data Manager Run a Script for an Error .....	7
	▶ How to Run a Script from the Command Line .....	9
	▶ How to Run a Script from Another Script .....	10
	▶ How to Run a Script from the Script Maintenance Screen .....	12
	<b>Running Scripts on a Schedule</b>	
	▶ Overview .....	14
	▶ The Permanent Schedule .....	15
	▶ The Permanent Schedule Maintenance Screen .....	16
	▶ The Permanent Schedule Screen .....	19
	▶ How to Create an Environment File .....	22
	▶ How to Add a Script to the Permanent Schedule .....	23
	▶ How to Copy a Schedule .....	25
	▶ How to Change a Script's Processing Schedule .....	28
	▶ How to Remove a Script From the Permanent Schedule .....	30
	<b>Restarting Scripts Automatically</b>	
	▶ Overview .....	32
	▶ The Recover Script .....	33
	▶ How to Make a Script Restart Automatically .....	34

---

# 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 <b>crontab</b> at scheduled times.
crontab	A UNIX system file that contains the files listing all the programs to be run by the <b>cron</b> daemon. Gentran:Server submits entries in the permanent schedule to <b>crontab</b> .
Permanent Schedule	The 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:

- ▶ Structure a data manager to run a script based on the type of data it encounters
- ▶ 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...
Use the script to handle certain types of data	Structure a data manager to run the script based on the data type.
Perform a special operation on a file before it passes to the next data manager	<p><b>Reference</b> See the <a href="#">How to Have a Data Manager Run a Script after Processing</a> topic in this chapter.</p> <p>Also see the <a href="#">How to Have a Data Manager Run a Script for an Error</a> topic in this chapter.</p>
Run or not run a script, depending on the logic defined in another script	<p>Invoke the script from another script.</p> <p><b>Reference</b> See the <a href="#">How to Run a Script from Another Script</a> topic in this chapter.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd)</b> <b>IF you want to...</b>	<b>THEN...</b>
Run a script on a one-time basis	Start the script from the Script Maintenance screen.
Run a script without exiting the Gentran:Server menu system	<b>Reference</b> See the <a href="#">How to Run a Script from the Script Maintenance Screen</a> topic in this chapter.
Test a new script	Run the script from the command line.  <b>Reference</b> See the <a href="#">How to Run a Script from the Command Line</a> topic in this chapter.
Perform a one-time housekeeping operation	
Run a script at regular intervals  <b>Example</b> You may want to run a communications script on a schedule.	Start the script from the Permanent Schedule.  <b>Reference</b> See the <a href="#">Running Scripts on a Schedule</a> section in this chapter.
Run a script to perform housekeeping operations on a schedule  <b>Example</b> Run the longterm script to handle long-term archiving	

**In this section**

This section contains these topics:

- ▶ How to Have a Data Manager Run a Script After Processing
- ▶ How to Have a Data Manager Run a Script for an Error
- ▶ 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 Have a Data Manager Run a Script after Processing

## Introduction

You can enable a data manager to invoke a Gentran:Server script based on the Trading Partnership code and modifier or the file name by specifying the name of the script on the configuration record. The data manager runs the script after it processes and routes the data.

### Reference

To run a UNIX shell script after processing, see the [How to Run a Script from Another Script](#) topic in this chapter.

## When to use

Use this procedure when you want to reformat or extract data for a custom program, such as a report generator.

## Procedure

Use this procedure to have a data manager run a script after it routes processed data for a specific Trading Partnership code or file name.

Step	Action
1	<p>Create the script that you want the data manager to run.</p> <p><b>Reference</b> See the <a href="#">Working with Scripts</a> chapter in this guide for instructions.</p>
2	<p>Create one or more configuration records that describe how you want the data manager to handle the type of files.</p> <p><b>Reference</b> See the <a href="#">Working with Configuration Records</a> chapter in this guide for instructions.</p>
3	<p>Type the name of the script from Step 1 into the <b>Script Name</b> field of each configuration record.</p> <p><b>Example</b> This is an example of using a script name in a configuration record.</p> <pre>Script Name advsr_as Desc Advantis Async Script</pre> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
4	<p>Open the data manager's initialization file and set the SCRIPT_RUN_SWITCH parameter. This parameter controls when the data manager runs the scripts listed on the configuration records.</p> <ul style="list-style-type: none"><li>▶ 0 = Execute script for each file routed using ROUTING_METHOD</li><li>▶ 1 = Execute script for all data in the input file</li><li>▶ 2 = Execute script at the end of a process cycle (after all the files in the queue or work directory are processed and before the data manager sleeps)</li></ul> <p><b>Reference</b> See the <a href="#">Maintaining Initialization Files</a> chapter in this guide for instructions.</p>
5	Test the data flow to ensure that the data manager starts the script after the data manager has finished processing the data.

---

# How to Have a Data Manager Run a Script for an Error

**Introduction** You can construct the Handlers section of the data manager's initialization file to run a Gentran:Server script as the result of a specific data or error condition.

**When to use** Use this procedure when you want the data manager to invoke a beeper script or another script when it encounters a specific data condition or type of error.

**Procedure** Use this procedure to have a data manager run a script if an error or other data condition is encountered.

Step	Action
1	<p>Create the script that you want the data manager to run.</p> <p><b>Reference</b> For instructions, see the <a href="#">Working with Scripts</a> chapter in this guide.</p>
2	<p>Open the data manager's initialization file and locate the <b>Handlers</b> section.</p> <p><b>Reference</b> See the <a href="#">How to Access and Edit an Initialization File</a> topic in the <a href="#">Maintaining Initialization Files</a> chapter in this guide for instructions.</p>
3	<p>Type the name of the error handler in the <b>:handler_name</b> position.</p> <p><b>Comment</b> This is the format of the Handlers section:</p> <p>&lt;handler_name&gt;    &lt;action&gt;    &lt;script_name&gt; or &lt;dirfile_name&gt;</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action	
4	Use this decision table to determine your next step.	
	<b>IF you want to...</b>	<b>THEN...</b>
	Trigger a beeper script when the error is encountered  <b>Example</b> <pre>:e_struct beep script/beeper.num AU 9 T2 504 973 4400 911001</pre> <b>Note</b> In this example, the action “beep” invokes the <i>beeper.scr</i> script that Sterling Commerce provides. You must modify the beeper script to either invoke the shell script file or submit the file to a communication package to dial the beeper.  <b>References</b> <ul style="list-style-type: none"> <li>▶ See the <a href="#">How to Set Error Notification</a> topic in the <a href="#">Maintaining Initialization Files</a> chapter for more information.</li> <li>▶ See the <a href="#">Working with Scripts</a> chapter for instructions on modifying scripts.</li> </ul>	<ul style="list-style-type: none"> <li>▶ Type <b>beep</b> in the action position.</li> <li>▶ Type the path name of the beeper number file in the script_name position.</li> <li>▶ Type the beeper number on the line that follows beep. This line is appended to the beeper number file.</li> </ul>
	Run a script when the error is encountered  <b>Example</b> <pre>:notp script script/notp</pre>	<ul style="list-style-type: none"> <li>▶ Type <b>script</b> in the action position.</li> <li>▶ Type the path name of the script in the script_name position.</li> </ul>
5	Save your changes and exit the editor.	

# How to Run a Script from the Command Line

---

**Introduction** You can invoke a 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: <code>smgr -s&lt;scriptname&gt;</code> Where <scriptname> is the name of the script. <b>Example</b> To run the script named <i>mvedi.scr</i> , type the following at the command line: <code>smgr -smvedi</code> <b>Note</b> Do not include the <i>.scr</i> extension in the script name.

# How to Run a Script from Another Script

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

**When to use**    Use this procedure when you want to:

- ▶ Invoke UNIX shell script or Gentran:Server scripts from a Gentran:Server script
- ▶ Run or not run a UNIX shell script or a Gentran:Server script based on logic statements in a Gentran:Server script
- ▶ Use a 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 Gentran:Server script.

Step	Action
1	<p>Does the script you want to invoke exist?</p> <ul style="list-style-type: none"> <li>▶ If YES, continue with Step 2.</li> <li>▶ If NO, create the script that you want to invoke.</li> </ul> <p><b>Reference</b> See the <a href="#">Working with Scripts</a> chapter in this guide for instructions.</p>
2	<p>Does the script that will start the script exist?</p> <ul style="list-style-type: none"> <li>▶ If YES, continue with Step 3.</li> <li>▶ If NO, create the script and then continue with Step 3.</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
3	<p>Modify the script from Step 2 to include a step in the STEPS group that starts the script that you want to invoke.</p> <p><b>Example 1</b> This example shows a step to invoke the Cleo UNIX shell script.</p> <pre>:run_cleo ## Execute the Cleo shell script sh \$cleoDir/\$VAN.run</pre> <p><b>Example 2</b> This example shows a step to invoke a Gentran:Server script named check_ftp.</p> <pre>:run_check ## Start the Server check_ftp script smgr -scheck_ftp</pre>
4	<p>Add a PROC to the PROCS group to execute the new step.</p> <p><b>Example</b></p> <pre>:run_cleo if run_cleo then check_if_tried_send else check_if_tried_send</pre>
5	<p>Save your changes and exit the editor.</p>

# How to Run a Script from the Script Maintenance Screen

**Introduction** The Gentran:Server Script Maintenance screen enables you to start a 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	<p>Select <b>Script</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Script Maintenance screen.</p> <pre> 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                     </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
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. <b>System Response</b> Gentran:Server runs the script. The script's status is active while the script is running.

---

# Running Scripts on a Schedule

## Overview

---

**Introduction**    Gentran:Server enables you to run 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 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 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

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 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 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
Min          Hour          Day  Month  Week Day  Script File
00           *             *   *     *         Longterm
00,10,20,30, 00,12       01,1 *   *         cnetser_as
00,20,41      12         *   *     *         advsr_as

F2:Add F3>Delete F4:Copy F5>Edit F9:Quit F10:Save
    
```

**Permanent Schedule Maintenance screen fields and functions**

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

Field	Function
Min	<p>Displays the time within the hour that the script is to run.</p> <p><b>Examples</b></p> <p>The number 00 in this field means that the script runs on the hour.</p> <p>The number 30 in this field means that the script runs at 30 minutes past the hour.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Field</b>	<b>Function</b>
Hour	Displays the hours that the script runs, based on a 24-hour clock.  <b>Examples</b> 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.  <b>Examples</b> 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.  <b>Example</b> 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.  <b>Examples</b> 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.

---

(Continued on next page)

**Permanent  
Schedule  
Maintenance  
screen function  
keys**

This table describes the function keys on the Permanent Schedule Maintenance screen and their functions.

---

<b>Key</b>	<b>Function</b>
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 Gentran:Server script, you complete the Permanent Schedule screen.

### Permanent Schedule screen

This illustration shows the Permanent Schedule screen.

```

Permanent Schedule: Longterm
Minutes in the Hour
0      1      2      3      4      5      5
0 | 0 | 0 | 0 | 0 | 0 | 9
X

Hours of the Day      Days of the Month
0 0 1 1 2      0 1 2 3
0 | 6 | 2 | 8 | 3      1 | 0 | 0 | 0
*                      *
Months of the Year    Days of the Week
JFMAMJJASOND        $MTWTFS
*                      *
F9:Exit F10:Save
  
```

### 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	<p>Defines the time within the hour that the script is to run.</p> <p><b>Examples</b></p> <p>To run the script on the half hour, type <b>X</b> at 30.</p> <p>To run the script at 45 minutes past the hour, type <b>X</b> at the halfway point between 40 and 50.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Field</b>	<b>Function</b>
Hours of the Day	<p>Defines the hours that the script runs, based on a 24-hour clock.</p> <p><b>Examples</b> Type an <b>X</b> at 06 to run the script at 6 a.m. Type an <b>X</b> at 18 to run the script at 6 p.m. To run the script every hour, type an asterisk (*) in the first position.</p>
Days of the Month	<p>Defines the days of the month that the script is run, based on a 31-day calendar.</p> <p><b>Examples</b> Type an <b>X</b> at 01 to run the script on the first day of the month. Type an <b>X</b> 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.</p>
Months of the Year	<p>Defines the months the script is to run, January through December.</p> <p><b>Examples</b> Type an <b>X</b> 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.</p>
Days of the Week	<p>Displays the days of the week the script is to run, Sunday through Saturday.</p> <p><b>Examples</b> Type an <b>X</b> 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.</p>

---

(Continued on next page)

**Using the  
Permanent  
Schedule screen**

This table describes how to enter the run schedule on the Permanent Schedule screen.

<b>IF you want to...</b>	<b>THEN...</b>
Select the minutes, hours, days, or months	Type <b>X</b> in each field that applies.
Select all values for the <b>Hours of the Day, Days of the Month, Months of the Year, or Days of the Week</b> field	Type an asterisk (*) in the first column of the field. The system clears the field.  You cannot use this option in the <b>Minutes in the Hour</b> 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 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 <i>\$EDI_ROOT/.edi_env</i> .
2	Type into the <i>.edi_env</i> file the following: <ul style="list-style-type: none"> <li>▶ The environment variables in which you want the process to run</li> <li>▶ The location of EDI_ROOT and its path.</li> </ul>

### 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 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	<p>Select <b>Sched</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Permanent Schedule Maintenance screen.</p> <pre> Permanent Schedule Maintenance ----- Min      Hour      Day      Month   Week Day   Script File 00       *          *        *       *          Longterm 00,10,20,30, 00,12    01,1    *       *          cnetsr_as 00,20,41    12       *       *       *          advsr_as </pre> <p>F2:Add F3:Delete F4:Copy F5:Edit F9:Quit F10:Save</p>
2	<p>Press F2 to add the script to the schedule.</p> <p><b>System Response</b> Gentran:Server displays a screen to add the script name.</p> <pre> EDI-Server Script ----- Script Name [REDACTED] ----- F2&gt;Select F9:Quit F10:Save </pre> <p>(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Press F2 to display a list of script names that were added to the script library. Select the script name and then press ENTER.</p> <p><b>System Response</b> Gentran:Server displays a Permanent Schedule screen, which is the screen you use to set up the schedule for the script.</p> <div data-bbox="634 625 1419 978" style="border: 1px solid black; padding: 5px;"> <pre> Permanent Schedule: Longterm Minutes in the Hour 0      1      2      3      4      5      5 0   0   0   0   0   0   9 X  Hours of the Day      Days of the Month 0  0  1  1  2      0  1  2  3 0   6   2   8   3  1   0   0   0 *                      *  Months of the Year    Days of the Week JFMAMJJASOND        SMTWTFS *                      *           </pre> <p>F9:Exit F10:Save</p> </div>
4	<p>Type <b>Xs</b> and asterisks in the fields to select the schedule.</p> <p><b>Reference</b> See <a href="#">The Permanent Schedule</a> for information.</p>
5	<p>Press F10 to save the changes to the Permanent Schedule.</p>
6	<p>Press F10 again to submit the schedule to <b>crontab</b>.</p>

# 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	<p>Select <b>Sched</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Permanent Schedule Maintenance screen.</p> <pre> Permanent Schedule Maintenance ----- Min      Hour      Day      Month  Week Day  Script File ----- 00       *         *        *      *         Longterm 00,10,20,30, 00,12    01,1    *      *         cnetsr_as 00,20,41    12       *        *      *         advsr_as </pre> <p>F2:Add F3&gt;Delete F4:Copy F5&gt;Edit F9:Quit F10:Save</p>
2	<p>Select the line of the schedule you want to copy and then press F4.</p> <p><b>System Response</b> Gentran:Server copies the line and displays a Confirmation prompt.</p> <pre> ----- Confirmation ----- Make a copy of this record ? </pre> <p>(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	Type <b>y</b> to confirm the copy.  <b>System Response</b> Gentran:Server displays a screen for the new script name.  <pre>                     EDI-Server Script                     Script Name ██████████                     F2:Select  F9:Quit  F10:Save                     </pre>
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 <b>crontab</b> .

**Running a script on another schedule**

Use this procedure to copy an entry and set up another schedule for the same script.

<b>Step</b>	<b>Action</b>
1	Select <b>Sched</b> from the host main menu.  <b>System Response</b> Gentran:Server displays the Permanent Schedule Maintenance screen.
2	Select the line of the schedule you want to copy and then press F4.  <b>System Response</b> Gentran:Server copies the line and displays a Confirmation prompt.  <pre>                     Confirmation                     Make a copy of this record ?                     </pre> <p style="text-align: right; color: red;">(Continued on next page)</p>



(Contd) Step	Action
3	<p>Type <b>y</b> to confirm the copy.</p> <p><b>System Response</b> Gentran:Server displays a screen for the new script name.</p> <pre data-bbox="678 533 1377 642"> EDI-Server Script Script Name ██████████ F2:Select F9:Quit F10:Save                     </pre>
4	<p>Type the name of the script that you copied.</p>
5	<p>Press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the Permanent Schedule screen for the script.</p> <pre data-bbox="639 932 1416 1283"> Permanent Schedule: advsr_as Minutes in the Hour 0 1 2 3 4 5 5 0   0   0   0   0   0   9 X     X     X     Hours of the Day      Days of the Month 0 0 1 1 2           0 1 2 3 0   6   2   8   3   1   0   0   0 X     X     ██████████ Months of the Year    Days of the Week JFMAMJJASOND        SMTWTFS *                     * F9:Exit F10:Save                     </pre>
6	<p>Change the schedule for the duplicate entry.</p> <p><b>Reference</b> To change the processing schedule, see the <a href="#">How to Change a Script's Processing Schedule</a> topic in this chapter.</p>

# How to Change a Script's Processing Schedule

**Introduction**    You may change any part of a Gentran:Server script's processing schedule.

**Procedure**    Use this procedure to change a script's processing schedule.

Step	Action
1	<p>Select <b>Sched</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Permanent Schedule Maintenance screen.</p> <pre data-bbox="651 915 1409 1251"> Permanent Schedule Maintenance ----- Min      Hour      Day      Month  Week Day  Script File ----- 00      *          *        *      *          Longterm 00,10,20,30, 00,12    01,1    *      *          cnetst_as 00,20,41    12      *      *      *          advsr_as  F2:Add F3&gt;Delete F4:Copy F5&gt;Edit F9:Quit F10:Save                     </pre>
2	<p>Select the script line that you want to change.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

(Contd) Step	Action								
3	<p>Press F5 to edit the processing schedule.</p> <p><b>System Response</b> Gentran:Server displays the Permanent Schedule screen for the script.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> Permanent Schedule: advsr_as Minutes in the Hour 0      1      2      3      4      5      5 0   0   0   0   0   0   9 X      X      X Hours of the Day      Days of the Month 0 0 1 1 2      0 1 2 3 0   6   2   8   3   1   0   0   0       X      * Months of the Year      Days of the Week JFMAMJJASOND      SMTWTFS *      * F9:Exit F10:Save                     </pre> </div>								
4	<p>Use this table to change the schedule.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th data-bbox="620 1014 1024 1073">IF you want to...</th> <th data-bbox="1024 1014 1422 1073">THEN...</th> </tr> </thead> <tbody> <tr> <td data-bbox="620 1073 1024 1131">Select a value</td> <td data-bbox="1024 1073 1422 1131">Type an <b>X</b>.</td> </tr> <tr> <td data-bbox="620 1131 1024 1222">Select all values in any field (except the <b>Minutes</b> field)</td> <td data-bbox="1024 1131 1422 1222">Type an asterisk (*) in the first position.</td> </tr> <tr> <td data-bbox="620 1222 1024 1276">Remove a character</td> <td data-bbox="1024 1222 1422 1276">Press the space bar.</td> </tr> </tbody> </table>	IF you want to...	THEN...	Select a value	Type an <b>X</b> .	Select all values in any field (except the <b>Minutes</b> field)	Type an asterisk (*) in the first position.	Remove a character	Press the space bar.
IF you want to...	THEN...								
Select a value	Type an <b>X</b> .								
Select all values in any field (except the <b>Minutes</b> field)	Type an asterisk (*) in the first position.								
Remove a character	Press the space bar.								
5	<p>Press F10 to save your changes.</p> <p><b>System Response</b> Gentran:Server displays the Permanent Schedule menu and a message that asks if you would like to change the Gentran:Server script.</p>								
6	Type <b>n</b> at the prompt and continue with Step 7.								
7	Press F10 to submit the modified schedule to <b>crontab</b> .								

# How to Remove a Script From the Permanent Schedule

**Introduction** You can remove a 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	<p>Select <b>Sched</b> from the host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Permanent Schedule Maintenance screen.</p> <pre> Permanent Schedule Maintenance ----- Min      Hour      Day      Month   Week Day   Script File ----- 00      *          *        *       *          Longterm 00,10,20,30, 00,12    01,1    *       *          cnetstr_as 00,20,41    12      *       *       *          advsr_as           </pre> <p>F2:Add F3&gt;Delete F4:Copy F5&gt;Edit F9:Quit F10:Save</p>
2	<p>Select the script.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Press F3 to delete the script from the schedule.</p> <p><b>System Response</b> Gentran:Server displays a confirmation prompt.</p> <div data-bbox="867 541 1177 604" style="border: 1px solid black; padding: 5px; margin: 10px auto; text-align: center;"><code>Confirmation Delete this record ?</code></div>
4	<p>Type <b>y</b> at the prompt to confirm the deletion.</p> <p><b>System Response</b> Gentran:Server displays the Permanent Schedule menu.</p>
5	<p>Press F10 to save the new schedule and submit it to <b>crontab</b>.</p> <p><b>Comment</b> If you delete all the scripts, you must still press F10.</p>
6	<p>Press F9 to exit the screen.</p>

# Restarting Scripts Automatically

## Overview

---

**Introduction** Unexpected machine halts disrupt script processing. To restart Gentran:Server scripts after a machine halt, 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 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 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 <b>Script</b> from the host main menu.  <b>System Response</b> 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.  <b>Example</b>  <pre> :Run_script STEP Section Commands: smgr -s 'head -1 scriptlist' sed -e '1d' &lt;scriptlist &gt;scriptlist.tmp mv scriptlist.tmp scriptlist :Check_list test -s scriptlist                     </pre> <div style="text-align: right; color: red;">(Continued on next page)</div>



<b>(Contd) Step</b>	<b>Action</b>
6	<p>Do you want the recover script to generate a notice to let you know which scripts were running when the machine halted?</p> <ul style="list-style-type: none"> <li>▶ If YES, add a STEP that checks the script directory for files with a <i>.l extension</i> and then checks to see if the script manager is running the script that created the log.</li> </ul> <p><b>Example</b></p> <pre>:Find_Scripts ls script/*.l   cut -c7- &gt; filename cut -d -f1 &lt;filename&gt;scriptname ps -ef   grep 'head -1 scriptname'</pre> <ul style="list-style-type: none"> <li>▶ If NO, continue with Step 7.</li> </ul>
7	<p>Do you want the recover script to tell you which scripts still have logs and appear active, even though they aren't active currently?</p> <ul style="list-style-type: none"> <li>▶ 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.</li> </ul> <p><b>Example</b></p> <pre>:Mail_list echo "Recover scripts found these scripts not running" &gt;mailfile cat scriptname &gt;&gt;mailfile sh mail_proc/default mailfile If NO, continue with Step 8.</pre>
8	Press F10 to save the script.

---

(Continued on next page)

**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 Gentran:Server and access the UNIX command line. <b>WARNING</b> From Gentran:Server, press CTRL+A to access the UNIX command line.
2	Open the <i>/etc/inittab</i> or <i>/etc/rc</i> file. <b>Reference</b> See your UNIX documentation for instructions.
3	Add the recover script to the file. <b>Example</b> <code>smgr -srecover</code>

---

# Defining the Document Reference Number

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	3
	▶ The Document Reference Number .....	5
	▶ How the Document Reference Number is Set .....	6
	▶ Document Specifier Tables .....	10
	<b>Defining Document Reference Numbers</b>	
	▶ Overview .....	12
	▶ Document Reference Number Specifier Screen .....	13
	<b>Creating a Document Specifier Table</b>	
	▶ Overview .....	15
	▶ How to Add a Document Specifier Table .....	16
	▶ How to Display a Document Specifier Table .....	19
	▶ How to Copy a Document Specifier Table .....	22
	<b>Mapping Document Specifier Tables</b>	
	▶ Overview .....	24
	▶ EDI Add Screen .....	28
	▶ APP Add Screen .....	33
	▶ XML Add Screen .....	37
	▶ NCPDP Add Screen .....	40
	▶ How to Add a Map to a Document Specifier Table .....	43
	▶ Using the Qualifier and Occurrence Fields .....	46
	▶ How to Display a Map Picture .....	48
	<b>Attaching a Trading Partnership Code to a Table</b>	
	▶ How to Attach a Trading Partnership Code to a Table .....	51
	▶ How to Verify Trading Partnership Code Attachments .....	55
	<b>Maintaining Document Reference Number Tables</b>	

- ▶ Overview ..... 57
- ▶ How to Change a Trading Partnership Code and Table Attachment ..... 58
- ▶ How to Remove a Trading Partnership Code from a Table ..... 59
- ▶ How to Delete a Document Specifier Table ..... 61

### **Maintaining Document Specifier Maps**

- ▶ Overview ..... 62
  - ▶ EDI Mapping Screen ..... 63
  - ▶ APP Mapping Screen ..... 66
  - ▶ XML Mapping Screen ..... 69
  - ▶ NCPDP Mapping Screen ..... 72
  - ▶ How to Display a Document Specifier Map ..... 74
  - ▶ How to Change a Document Specifier Map ..... 76
  - ▶ How to Delete a Map from a Table ..... 79
-

# 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 Gentran:Server attaches to a document to identify the date and time the document was received.
document reference number	The unique number that 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 Gentran:Server processes reference to construct the document reference number.
document specifier utility	The 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 Gentran:Server generates to identify a session in which files were received and passed.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
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 Gentran:Server handles.

---

# The Document Reference Number

## Introduction

To track the movement of a document through the system, 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 Gentran:Server.

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

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	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 Gentran:Server data managers and the translator either:

- ▶ Extract the document reference number from the document itself
- ▶ 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. This topic describes how data managers or processes extract or produce the document reference numbers.

### Application data manager

This table shows how the Application data manager (*appm* personality) responds to the location of the Trading Partnership code.

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 or Not attached to a document specifier table	Does not create a document reference number. The document reference number field in the transaction register is blank. <ul style="list-style-type: none"> <li>▶ Gentran:Server cannot use the document reference number to check for duplicates.</li> <li>▶ If you use the Life Cycle facility, the Life Cycle update process fails.</li> <li>▶ Gentran:Server cannot use the document reference number to retrieve archived files.</li> </ul>

### Download data manager

The Download data manager (*dnld* personality) extracts the document reference number from the 01 record, character positions 19-58.

(Continued on next page)



**File and Translation data managers**

This table shows how the File (file personality) and Translation (*xltr* personality) data managers respond to the setting for the ADD\_MBAG\_TO\_DOC\_REF\_NUM parameter in the data manager's initialization file

Setting	Data manager response
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.

**Host Command Card data manager**

The Host Command Card data manager (*hcmd* personality) extracts the document reference number from the third field in the host command card.

**Inbound data manager**

This table shows how the inbd personality responds to the setting for the Trading Partnership code.

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.

(Continued on next page)

**Translator**

This table describes the source the translator (lfrtran component) uses for the document reference number.

<b>IF the document type is...</b>	<b>THEN the translator...</b>
Inbound EDI	Uses the document reference number in the temporary audit file from the instructions in a document specifier table.
Outbound EDI	Derives the document reference number from the document specifier table.  If not mapped or if the Trading Partnership is not attached to a document specifier table, Gentran:Server leaves the document reference number blank and writes a warning message to <i>xlcntl.err</i> .

**NCPDP data manager**

This table describes how the NCPDP data manager (npdp personality) derives the document reference number.

<b>IF the Trading Partnership code is...</b>	<b>THEN the data manager...</b>
Attached to a document specifier table	Derives the document reference number from the instructions in an NCPDP document specifier table.
Attached to a document specifier table, but no entry for the application exists in the table  or  Not attached to a document specifier table	Does not create a document reference number. The document reference number field in the transaction register is blank. <ul style="list-style-type: none"> <li>▶ Gentran:Server cannot use the document reference number to check for duplicates.</li> <li>▶ If you use the Life Cycle facility, the Life Cycle update process fails.</li> <li>▶ Gentran:Server cannot use the document reference number to retrieve archived files.</li> </ul>

(Continued on next page)

**XML data manager**

This table describes how the XML data manager (xmli personality) derives the document reference number.

<b>IF the Trading Partnership code is...</b>	<b>THEN the data manager...</b>
Attached to a document specifier table	Derives the document reference number from the instructions in an XML document specifier table.
Attached to a document specifier table, but no entry for the application exists in the table  or  Not attached to a document specifier table	Does not create a document reference number. The document reference number field in the transaction register is blank. <ul style="list-style-type: none"> <li>▶ Gentran:Server cannot use the document reference number to check for duplicates.</li> <li>▶ If you use the Life Cycle facility, the Life Cycle update process fails.</li> <li>▶ Gentran:Server cannot use the document reference number to retrieve archived files.</li> </ul>

# 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 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 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.

---

(Continued on next page)

**Default table for EDI documents**

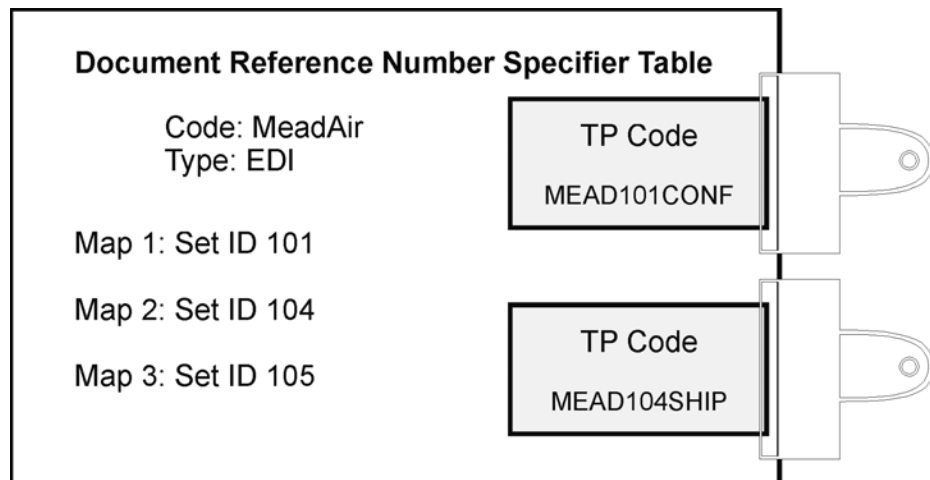
Gentran:Server has a default document specifier table for EDI documents. This table defines a basic way to extract the document reference number. You can use the default table or create your own to override the default settings.

**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. <b>Reference</b> See <a href="#">Creating a Document Specifier Table</a> .
2	Construct one or more set ID, application name, XML name, or NCPDP transaction code maps for the document specifier table. <b>Reference</b> See <a href="#">Mapping Document Specifier Tables</a> .
3	Attach one or more Trading Partnership codes to the table. <b>Reference</b> See <a href="#">Attaching a Trading Partnership Code to a Table</a> .

---

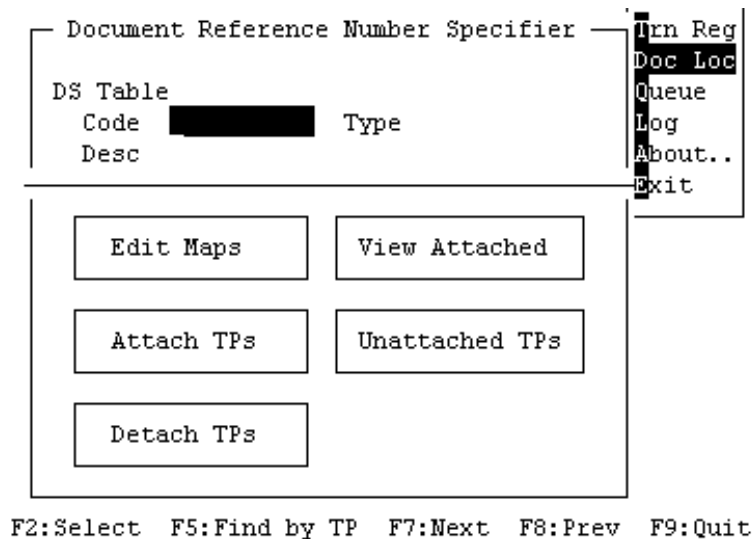
# Document Reference Number Specifier Screen

## 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: <ul style="list-style-type: none"> <li>▶ EDI</li> <li>▶ APP</li> <li>▶ XML (if you have the XML translation option)</li> <li>▶ NCP (if you use the NCPDP standard)</li> </ul>

(Continued on next page)

<b>(Contd) Field</b>	<b>Function</b>
DS Table Description	Describes the document specifier table.
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.

<b>Key</b>	<b>Function</b>
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.

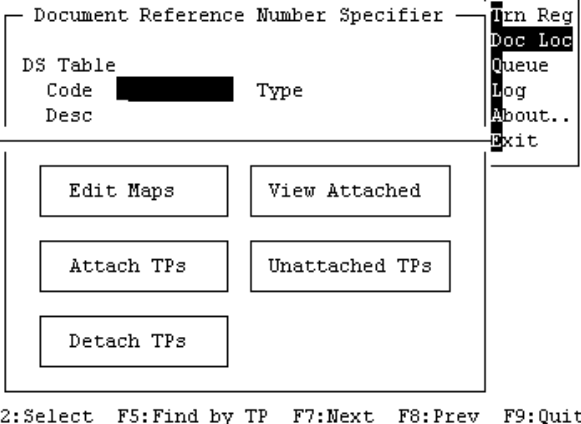
---

# 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	<p>Select <b>Doc Loc</b> from the host Util menu.</p> <p><b>System Response</b> Gentran:Server displays the Document Reference Number Specifier screen.</p>  <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>		
2	Type the name of the new table in the DS Table Code field and press ENTER.  <b>System Response</b> The utility searches a list of table names for a match.		
	<b>IF the DS Table Code...</b>	<b>THEN the system displays...</b>	<b>AND you should...</b>
	Exists	The table description and type on the Document Reference Number Specifier screen	Enter a different name for your table.
	Does not exist	A “Confirmation Partial Key Lookup” prompt to have Gentran:Server use the characters you entered as a table search key	Continue with Step 3.
3	Type <b>y</b> for yes and press ENTER.  <b>System Response</b> The system uses the characters as a search key for a table name.		
	<b>IF...</b>	<b>THEN the system...</b>	<b>AND you should...</b>
	No table names match the search characters	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	Continue with Step 4.
One or more table names match the search characters	Displays the first table type and description that matches	Quit, start again, and give the table a code name that does not exist.	
(Continued on next page)			

<b>(Contd) Step</b>	<b>Action</b>										
4	<p>Type <b>EDI, APP, XML, or NCP</b> in the Type field to specify the table type and press ENTER.</p> <p><b>System Response</b> The system displays a Confirmation prompt.</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p style="text-align: center;">— Confirmation —</p> <p>DS Table/Type Not Found Do You Wish to Add It ? _</p> </div>										
5	<p>Type <b>y</b> for yes and press ENTER.</p> <p><b>System Response</b> The system moves the cursor to the Desc field.</p>										
6	<p>Type the description of the new table in the Desc field.</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p style="text-align: center;">Document Reference Number Specifier —</p> <p>DS Table</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Code</td> <td style="width: 15%;">Act</td> <td style="width: 15%;">App</td> <td style="width: 15%;">Type</td> <td style="width: 15%;">APP</td> </tr> <tr> <td>Desc</td> <td colspan="4">Accounting Application Table</td> </tr> </table> </div>	Code	Act	App	Type	APP	Desc	Accounting Application Table			
Code	Act	App	Type	APP							
Desc	Accounting Application Table										
7	<p>Press F10 to save the new table.</p>										
8	<p>GO TO <a href="#">Mapping Document Specifier Tables.</a></p>										

# 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.

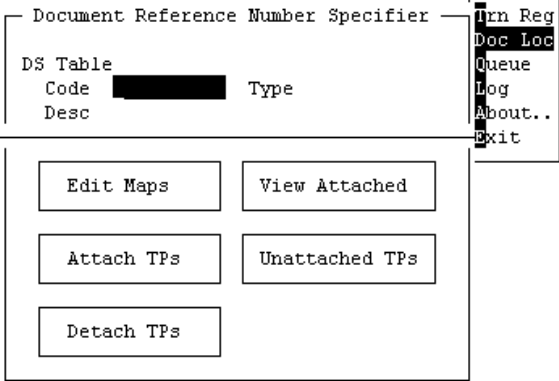
## Two ways to search for a table

There are two ways to search for and display a document specifier table from the Document Reference Number Specifier screen:

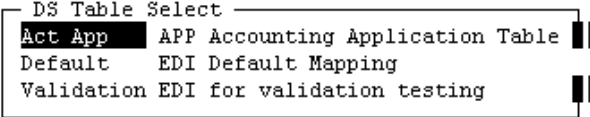
- ▶ Press F2, the select key, to display a list of all the existing tables.
- ▶ Press F5 to specify a Trading Partnership code that is attached to the table you want.

## Selecting a table with the select key

Use this procedure to select a table with F2, the select key.

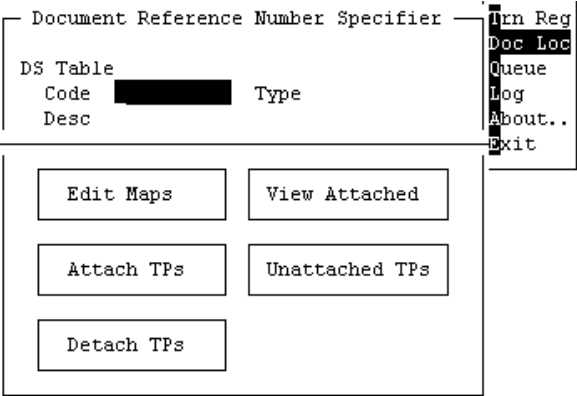
Step	Action
1	<p>Select <b>Doc Loc</b> from the host Util menu.</p> <p><b>System Response</b> Gentran:Server displays the Document Reference Number Specifier screen.</p>  <p style="text-align: center;">F2:Select F5:Find by TP F7:Next F8:Prev F9:Quit</p>
2	<p>Select the DS Table Code field.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>



(Contd) Step	Action
3	<p>Press F2 to display a list of document specifier tables.</p> 
4	<p>Select the name of the table and press ENTER.</p> <p><b>System Response</b> The system enters the table code, type, and description in the DS Table fields.</p>
5	GO TO <a href="#">How to Copy a Document Specifier Table.</a>

**Locating a table by the Trading Partnership code**

Use this procedure to find a table by the Trading Partnership code.

Step	Action
1	<p>Select <b>Doc Loc</b> from the host Util menu.</p> <p><b>System Response</b> Gentran:Server displays the Document Reference Number Specifier screen.</p>  <p>F2:Select    F5:Find by TP    F7:Next    F8:Prev    F9:Quit</p>
2	<p>Select the DS Table Code field.</p> <p style="text-align: right;"><b>(Continued on next page)</b></p>

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Press F5 to use the Trading Partnership code to locate the table.</p> <p><b>System Response</b> The system displays the Find Document Specifier Table by TP Search screen.</p> <div data-bbox="683 575 1362 863" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> Find Document Specifier Table by TP ----- TP Code   Name  Doc Ver Doc Typ  DS Table Code          Type   Desc F2:Select  F9:Quit  F10:Save </pre> </div>
4	Complete the <b>Type</b> 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	<p>Select the Trading Partnership code from the list.</p> <p><b>WARNING</b> <b>You can select only one Trading Partnership code.</b></p> <p><b>System Response</b> Gentran:Server displays the table values in the Find Document Specifier Table by TP screen.</p>
7	<p>Note the DS Table name displayed. Is this the table you want to copy?</p> <ul style="list-style-type: none"> <li>▶ If YES, press F10 to save the selection and continue.</li> <li>▶ If NO, press F9 to clear the fields and start over.</li> </ul> <p><b>System Response</b> When you press F10, the system enters the DS Table Code and Description in the Document Reference Number Specifier screen.</p>
8	GO TO <a href="#">How to Copy a Document Specifier Table</a> .

## 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 <a href="#">How to Display a Document Specifier Table</a> .
2	<p>Press F4 to copy the table.</p> <p><b>System Response</b> The system displays the Copy screen.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <pre>Copy _____ New DS Table Code [REDACTED] Type EDI Desc F9:Quit F10:Save _____</pre> </div>
3	<p>Enter the code and description for the new table in the appropriate fields.</p> <p><b>System Response</b> The system enters the new table code and description in the Document Reference Specifier screen.</p>
4	Press F10 to save the table.
5	<p>Do you want to edit the new table's maps?</p> <ul style="list-style-type: none"> <li>▶ If YES, see <a href="#">Maintaining Document Specifier Maps</a>.</li> <li>▶ If NO, continue with Step 6.</li> </ul> <p style="text-align: right;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
6	Do you want to delete any of the new table's maps? <ul style="list-style-type: none"><li>▶ If YES, see <a href="#">How to Delete a Map from a Table</a>.</li><li>▶ If NO, continue with Step 7.</li></ul>
7	GO TO the <a href="#">Attaching a Trading Partnership Code to a Table</a> 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.

```

EDI Mapping
-----
Set Id 104   Air Shipment Information

      Sub-  --EDI--  ---DR---  Qual  Qualifying  Occur-
Segment Elemnt Elem Start Len Start Mark Elemnt  Value    rence
G47      2          1   40   1     m

```

F3:Delete F4:Add F5:Edit F6:PicUp F7:PicDn F9:Quit F10:Save

(Continued on next page)

## 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 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.

```

APP Mapping
-----
Application Name 204outh

Record Id      Name      St Ln  St Mrk  Name      Value      Occur-
01             Record Id - 01 5   1   &      Name

```

F3:Delete F4:Add F5:Edit F6:PicUp F7:PicDn F9:Quit F10:Save

## XML name maps

If you have the 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.

(Continued on next page)

**Example**

This is an example of an XML name map.

```

XML Mapping
DDF Name: xmldata

XML Target Node          Use  ---XML--- --DocRef--
PCDATA Start Len Start Mark
Transmission             y    1  15   1   X

F1:Zoom F3:Delete F4:Add F5:Edit F6:PicUp F7:PicDn F9:Quit F10:Save 6.0
    
```

**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.

```

NCPDP Mapping
Transaction Code  B1  Billing

Segment  Field  --NCPDP--  --DocRef--
Start Len  Start Mark
  1     CY     1   10     1   X

F3:Delete F4:Add F5:Edit F6:PicUp F7:PicDn F9:Quit F10:Save 6.0
    
```

(Continued on next page)

### 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.

```

Add
Segment Id      [REDACTED]
Element #
Sub-Elem #
Elem Start      from
Length
Doc Ref Start
Visual Marker
Qual Elem #
Qual Value
Occurrence
F9:Quit  F10:Save
    
```

**EDI Add screen fields and functions**

This table lists the fields on the EDI Add screen and their functions.

Field	Description
Segment ID	<p>Defines the segment in the document that contains the unique value. The Segment ID has a maximum length of 4 characters.</p> <p><b>Examples</b> BEG, DTM, EDF</p>
Element #	<p>Identifies the element number in the segment that contains a unique value. The element number can be any number from 1 to 999, inclusive.</p>
Sub-Elem #	<p>The sub-element number in the element that contains a unique value. The sub-element can be any number from 1 to 99, inclusive.</p>

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
Elem Start from	<p>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 <b>from</b> the left or the right of the element value. The Elem Start from value can be any number from 1 to 99, inclusive. Enter <b>Left</b> or <b>right</b> for counting direction.</p> <p><b>Comments</b> 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.</p> <p><b>Example</b> The element you want to extract values from is variable length and you want to extract the last three characters from the right.</p> <pre>Elem Start = 1 from = right Length = 3</pre> <p>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.</p>
Length	<p>The length (in number of characters) of the unique value. This value can be any number from 1 to 40, inclusive</p> <p><b>Comments</b> 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).</p> <p>The total length of all lines in the map cannot exceed 40 characters.</p>
Doc Ref Start	<p>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.</p> <p style="text-align: right;"><i>(Continued on next page)</i></p>

<b>(Contd) Field</b>	<b>Description</b>
Visual Marker	<p>The symbol you want to use to mark this part of the document reference number in the map picture. The Visual Marker is only 1 character in length.</p> <p>Examples: %, &amp;, *, A, B</p> <p><b>Comment</b> The system does not allow you to use the same marker more than once. If you have used the marker for another line, Gentran:Server displays a message to let you know you must choose another marker.</p> <p><b>Reference</b> See <a href="#">How to Display a Map Picture</a>.</p>
Qual Element #	<p>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.</p> <p>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.</p> <p><b>Comment</b> 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.</p> <p><b>Reference</b> See <a href="#">Using the Qualifier and Occurrence Fields</a> for more information.</p> <p style="text-align: right;">(Continued on next page)</p>



<b>(Contd) Field</b>	<b>Description</b>
Qual Value	<p>The value that the element number in the Qual Element field must have before 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.</p> <p><b>Example</b> 102097 means “if 102097 is the qualifying value, then extract values from specified segment and element”</p>
Occurrence	<p>The occurrence of the segment from which the specified characters are to be extracted. This field can be up to 5 characters in length.</p> <ul style="list-style-type: none"> <li>▶ 1 means use the first occurrence of the segment.</li> <li>▶ 2 means use the second occurrence of the segment.</li> <li>▶ 0 means use the last occurrence of the segment.</li> </ul> <p><b>Comments</b> If this field is empty, Gentran:Server uses the last occurrence of the segment.</p> <p>For interchange segments, 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.</p>

---

### 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.

---

(Continued on next page)

**EDI Add screen  
function keys**

---

This table describes the function keys of the EDI Add screen and their functions.

<b>Key</b>	<b>Function</b>
F2	Displays a list of choices for the selected field.
F9	Exits the screen.
F10	Saves the line to the <b>Set ID</b> 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.

```

Add
Record Id      ██████████
Field Name
Field Start    from
Length
Doc Ref Start
Visual Marker
Qual Fld Name
Qual Value
Occurrence
F2:Select F9:Quit F10:Save
    
```

**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  <b>Example</b> 01 means the 01 record
Field Name	Defines the field name. This value can be up to 128 characters, though only 12 are displayed.  <b>Example</b> weight

(Continued on next page)



<b>(Contd) Field</b>	<b>Description</b>
<p>Field Start from</p>	<p>Defines the character number in the selected field that starts the unique value. You must also specify whether the position is counted <b>from</b> the left or the right of the field. This value can be any from 1 to 99, inclusive. Use <b>Left</b> or <b>right</b> for counting direction.</p> <p><b>Comments</b> 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.</p> <p><b>Example 1</b> The field you want to extract values from is variable length and you want to extract the last three characters from the right.</p> <p>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.</p> <p><b>Example 2</b> 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.</p> <p>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.</p>
<p>Length</p>	<p>Defines the number of characters in the unique value. This value can be any number from 1 to 40, inclusive.</p> <p><b>Comment</b> The total length of all record lines in the map cannot exceed 40.</p>
<p>Doc Ref Start</p>	<p>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.</p> <p style="text-align: right;"><i>(Continued on next page)</i></p>

<b>(Contd) Field</b>	<b>Description</b>
Visual Marker	<p>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.</p> <p>Examples: %, &amp;, *</p> <p><b>Comment</b> The system does not allow you to use the same marker more than once. If you already used the marker for another line, Gentran:Server displays a message. You must choose another marker.</p> <p><b>Reference</b> See <a href="#">How to Display a Map Picture</a>.</p>
Qual Fld Name	<p>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.</p> <p><b>Comment</b> When the field with this name has the value specified in the <b>Qual Value</b> field, the specified characters are extracted. Otherwise, the characters are skipped.</p> <p><b>Reference</b> See <a href="#">Using the Qualifier and Occurrence Fields</a> for more information.</p>
Qual Value	<p>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.</p> <p><b>Example</b> 40</p>
Occurrence	<p>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.</p> <p><b>Example</b> A 3 means that Gentran:Server uses the third occurrence of the record.</p> <p><b>Comment</b> If this field is empty, Gentran:Server uses the last occurrence.</p> <p style="text-align: right;"><a href="#">(Continued on next page)</a></p>

---

**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.

<b>Key</b>	<b>Function</b>
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.

```

Add
XML Node      ██████████
Use PCDATA
XML Start
Length
Doc Ref Start
Visual Marker
F2:Select F9:Quit F10:Save
  
```

### XML Add screen fields and functions

This table lists the fields on the XML Add screen and their functions.

Field	Description
XML Node	<p>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.</p> <p><b>Example</b> 01 means the 01 record</p>
Use PCDATA	<p>Indicates whether or not the PCDATA data in the target tag is used to build the document reference number. This value can be either <i>Yes</i> or <i>No</i>.</p> <p><b>Note</b> If you select <i>No</i> for this field, the system extracts the tag name for the document reference number instead of the PCDATA.</p>
XML Start	<p>Defines the character number in the selected tag that starts the unique value. This value can be any number from 1 to 99, inclusive.</p>

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
Length	<p>Defines the number of characters in the unique value. This value can be any number from 1 to 40, inclusive.</p> <p><b>Comment</b> The total length of all record lines in the map cannot exceed 40.</p>
Doc Ref Start	<p>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.</p>
Visual Marker	<p>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.</p> <p>Examples: %, &amp;, *</p> <p><b>Comment</b> The system does not allow you to use the same marker more than once. If you already used the marker for another line, Gentran:Server displays a message. You must choose another marker.</p> <p><b>Reference</b> See <a href="#">How to Display a Map Picture</a>.</p>

---

**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.

---

(Continued on next page)



**XML Add screen  
function keys**

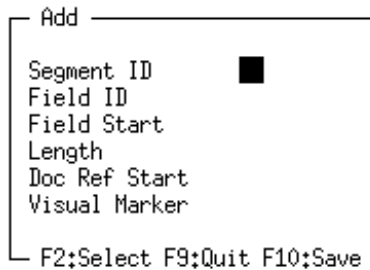
This table describes the function keys of the XML Add screen and their functions.

<b>Key</b>	<b>Function</b>
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.  <b>Examples</b> 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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
Length	<p>The length (in number of characters) of the unique value. This value can be any number from 1 to 40, inclusive.</p> <p><b>Comments</b></p> <p>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).</p> <p><b>Note</b></p> <p>The total length of all lines in the map cannot exceed 40.</p>
Doc Ref Start	<p>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.</p>
Visual Marker	<p>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.</p> <p><b>Comment</b></p> <p>The system does not allow you to use the same marker more than once. If you have used the marker for another line, Gentran:Server displays a message to let you know you must choose another marker.</p> <p><b>Reference</b></p> <p>See <a href="#">How to Display a Map Picture</a>.</p> <p><b>Examples</b></p> <p>%, &amp;, *, A, B</p>

---

## 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.

---

(Continued on next page)

**NCPDP Add  
screen function  
keys**

This table describes the function keys of the NCPDP Add screen and their functions.

---

<b>Key</b>	<b>Function</b>
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.

---

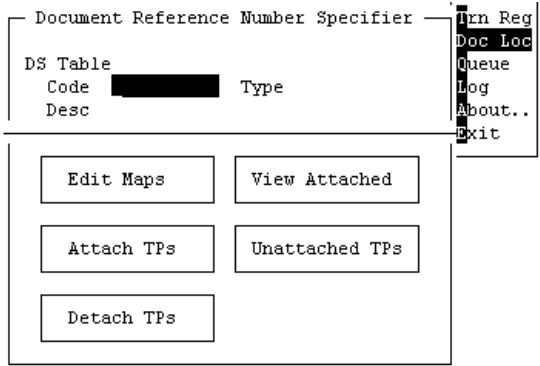
# How to Add a Map to a Document Specifier Table

**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 *Gentran:Server for UNIX and Workstation Application Integration User's 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	<p>Select <b>Doc Loc</b> from the host Util menu.</p> <p><b>System Response</b> Gentran:Server displays the Document Reference Number Specifier screen.</p>  <p>F2:Select F5:Find by TP F7:Next F8:Prev F9:Quit</p>
2	<p>Display the table you want to map.</p> <p><b>Reference</b> See <a href="#">How to Display a Document Specifier Table</a>.</p> <p style="text-align: right;">(Continued on next page)</p>



(Contd) Step	Action		
3	<p>Select <b>Edit Maps</b> and then press ENTER.</p> <p><b>System Response</b> The system displays a Mapping screen for the table type you specified (EDI, APP, XML, or NCP). The screen's fields are blank.</p> <p><b>Example</b> This is an example of the EDI Mapping screen.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> EDI Mapping Set Id ██████████            Sub-  --EDI--  ---DR---  Qual  Qualifying  Occur- Segment Elemnt Elem Start Len Start Mark Elemnt  Value  rence           </pre> </div> <p>F2:Select F5:Find F9:Quit</p>		
4	<p>Type the set ID, application name, DDF name, or NCPDP transaction code in the first field and then press ENTER.</p> <p><b>TIP</b> <b>To select from a list of all existing set IDs, application names, or DDFs, press F2.</b></p> <p><b>System Response</b></p>		
	<b>IF the set ID, application name, or DDF...</b>	<b>THEN</b> Gentran:Server...	<b>AND you should...</b>
	Is mapped to the document specifier table	Displays the map	Press F10 to clear the screen; then enter a different name.
	Is not mapped to the document specifier table	Displays a prompt that asks if you want to add the map	Type <b>y</b> to continue. <b>System Response</b> The system displays the Add screen for the first line.
	Does not describe the layout of an application file	Displays an error message	Select a different DDF.

(Continued on next page)

<b>(Contd) Step</b>	<b>Action</b>
5	<p>Complete the fields.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p>Add</p> <p>Segment Id      <span style="background-color: black; color: black;">██████████</span></p> <p>Element #</p> <p>Sub-Elem #</p> <p>Elem Start      from</p> <p>Length</p> <p>Doc Ref Start</p> <p>Visual Marker</p> <p>Qual Elem #</p> <p>Qual Value</p> <p>Occurrence</p> <p>F9:Quit F10:Save</p> </div>
6	<p>Press F10 to save the line to the map.</p> <p><b>System Response</b> Gentran:Server clears the fields in the Add screen so that you can add another segment or record line to the map.</p>
7	<p>Repeat Step 5 to add each subsequent line in your map.</p> <p><b>REMINDER</b> <b>The values that you define must result in a unique number for each document.</b></p>
8	<p>When your map is finished, press F10 to save the map and then press F9 to exit the Add screen.</p> <p><b>System Response</b> 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). Gentran:Server displays the mapping screen, which now lists all the lines you added to the map.</p> <p><b>TIP</b> <b>You can display a picture of the map by pressing F6.</b></p> <p><b>Reference</b> See <a href="#">How to Display a Map Picture</a> for details.</p>

## 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 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 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, 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 Gentran:Server to use for the document reference number.

### Example

A 4 in the Occurrence field means use the fourth occurrence.

---

(Continued on next page)



**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 Gentran:Server follows when you use the optional fields in a map line.

<b>IF you use...</b>	<b>THEN Gentran:Server...</b>
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.  <b>Example</b>  If the: Occurrence is 3 Qualifying Element # is 9 Qualifying Value is 38  then Gentran:Server looks for the third occurrence in which the value of the ninth element is 38.

**CAUTION**

**Gentran:Server cannot distinguish between interchange segments and other segments. For interchange segments, 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.

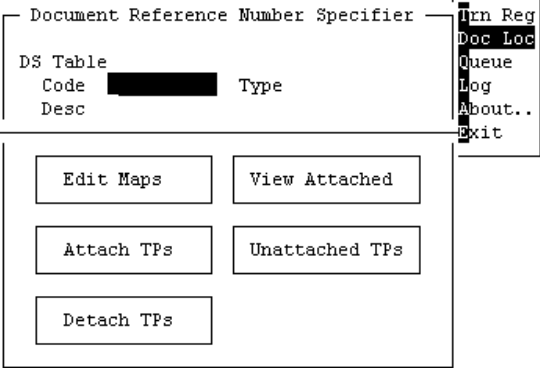
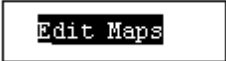
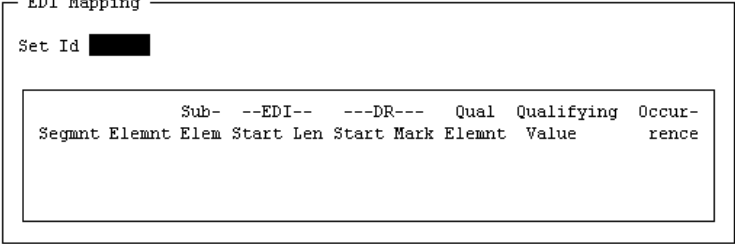
```
BEG                      EDF                      DTM
|                          |                          |
|                          |                          |
%                          * * * * *                ) ) ) ) ) ) ) ) )
-----
12345678901234567890123456789012345678901234567890
|                1                | 2                | 3                4
|                |                |                |
1                18               27
```

---

(Continued on next page)

**Displaying a map picture**

Use this procedure to display a map picture.

Step	Action
1	<p>Select <b>Doc Loc</b> from the host Util menu.</p> <p><b>System Response</b> Gentran:Server displays the Document Reference Number Specifier screen.</p>  <p>F2:Select F5:Find by TP F7:Next F8:Prev F9:Quit</p>
2	<p>Select the DS Table Code field.</p>
3	<p>Press F2 to display a list of document specifier tables and then select the table that has the map you want to view.</p>
4	<p>Select <b>Edit Maps</b> and then press ENTER.</p>  <p><b>System Response</b> Gentran:Server displays the Mapping screen.</p>  <p>F2:Select F5:Find F9:Quit</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
5	Select the <b>Set ID, Application Name, DDF Name, or Transaction Code</b> field.
6	<p>Press F5 to display a list of maps for the table, and then select the map you want to view.</p> <p><b>System Response</b> Gentrn:Server displays the segment or record lines in the map.</p> <div data-bbox="651 667 1393 919" style="border: 1px solid black; padding: 5px;"> <pre> EDI Mapping Set Id 104   Air Shipment Information            Sub-  --EDI--  ---DR---  Qual  Qualifying  Occur- Segment Elemnt Elem Start Len Start Mark Elemnt  Value   rence G47      2          1   40   1     m </pre> </div> <p>F3:Delete F4:Add F5:Edit F6:PicUp F7:PicDn F9:Quit F10:Save</p>
7	<p>Press F6 to display a picture of the map.</p> <p><b>System Response</b> The markers you used in the map identify the place in the document reference number that the segment or record values occupy.</p> <p><b>CAUTION</b> <b>You can edit the map while the picture is displayed.</b></p>
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.  <b>Reference</b> See <a href="#">How to Display a Document Specifier Table</a> in this chapter.  (Continued on next page)

<b>(Contd) Step</b>	<b>Action</b>
2	<p>Select <b>Attach Trading Partners</b>; then press ENTER.</p> <div style="text-align: center; border: 1px solid black; width: fit-content; margin: 0 auto; padding: 2px;"> <span>Attach TPs</span> </div> <p><b>System Response</b> The system displays the Trading Partnership Search screen.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <pre style="font-family: monospace; font-size: 0.9em;"> Trading Partnership Search ----- Do you wish to enter a range of ...  Trading Partnership Codes ?      N Interchange and/or Group Ids ?   N Organization Codes ?             N Standard Version and/or Set Ids ? N User Defined Categories ?        N  Target an Inbound Mapping Table ? N Target an Outbound Mapping Table ? N Search by Trading Partnership Name ? N                     </pre> </div> <p><b>Reference</b> See the <a href="#">Understanding the Basics</a> chapter in this guide for instructions on how to search for a Trading Partnership.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

(Contd) Step	Action																																																																						
3	<p>Type search criteria on the screen for the Trading Partnerships you want to see, then press F10 to continue.</p> <p><b>CAUTION</b>  <b>To list all the Trading Partnerships, press F10 without entering search criteria.</b></p> <p><b>System Response</b>                      The system displays a list of Trading Partnerships that matched the criteria you entered.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Trading Partnership Search</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">TP Code</th> <th style="text-align: left;">TP Name/Description</th> <th style="text-align: left;">Std Version</th> <th style="text-align: left;">Set Id</th> <th style="text-align: left;">Proc Record</th> </tr> </thead> <tbody> <tr> <td>4097out</td> <td></td> <td>ANA1_8</td> <td></td> <td>SRMHDR [N]</td> </tr> <tr> <td>INEND1911</td> <td>Sample Flow TP for 1911/invoic</td> <td>001911</td> <td></td> <td>INVOIC [N]</td> </tr> <tr> <td>INEND210</td> <td>Sample Flow TP for M2_8/210</td> <td>M2/8</td> <td></td> <td>210 [N]</td> </tr> <tr> <td>INEND210FA</td> <td>Sample Flow FA for M2_8/210</td> <td>M2/8</td> <td></td> <td>997 [N]</td> </tr> <tr> <td>INEND837</td> <td>Sample Flow TP for 03032/837</td> <td>003032</td> <td></td> <td>837 [N]</td> </tr> <tr> <td>INEND837FA</td> <td>Sample Flow FA for 03032/837</td> <td>003032</td> <td></td> <td>997 [N]</td> </tr> <tr> <td>INEND850</td> <td>Sample Flow TP for 02040/850</td> <td>002040</td> <td></td> <td>850 [N]</td> </tr> <tr> <td>INEND850FA</td> <td>Sample Flow FA for 02040/997</td> <td>002040</td> <td></td> <td>997 [N]</td> </tr> <tr> <td>OUTEND02856</td> <td>Sample Flow for TP 856 Div 2</td> <td>003030</td> <td></td> <td>856 [N]</td> </tr> <tr> <td>OUTEND03856</td> <td>Sample Flow for TP 856 Div 3</td> <td>003030</td> <td></td> <td>856 [N]</td> </tr> <tr> <td>OUTBOUND856</td> <td>Sample Flow for TP 856</td> <td>003030</td> <td></td> <td>856 [N]</td> </tr> <tr> <td>TDCC204-1</td> <td>Sample Flow for 1st TP for 204 M2/8</td> <td></td> <td></td> <td>204 [N]</td> </tr> <tr> <td>TDCC204-2</td> <td>Sample Flow for 2nd TP for 204 M2/8</td> <td></td> <td></td> <td>204 [N]</td> </tr> </tbody> </table> <p style="font-size: small; margin-top: 5px;">Esc-QUIT F1-HELP F2-MARK ALL F3-UNMARK ALL F10-CONTINUE</p> </div>	TP Code	TP Name/Description	Std Version	Set Id	Proc Record	4097out		ANA1_8		SRMHDR [N]	INEND1911	Sample Flow TP for 1911/invoic	001911		INVOIC [N]	INEND210	Sample Flow TP for M2_8/210	M2/8		210 [N]	INEND210FA	Sample Flow FA for M2_8/210	M2/8		997 [N]	INEND837	Sample Flow TP for 03032/837	003032		837 [N]	INEND837FA	Sample Flow FA for 03032/837	003032		997 [N]	INEND850	Sample Flow TP for 02040/850	002040		850 [N]	INEND850FA	Sample Flow FA for 02040/997	002040		997 [N]	OUTEND02856	Sample Flow for TP 856 Div 2	003030		856 [N]	OUTEND03856	Sample Flow for TP 856 Div 3	003030		856 [N]	OUTBOUND856	Sample Flow for TP 856	003030		856 [N]	TDCC204-1	Sample Flow for 1st TP for 204 M2/8			204 [N]	TDCC204-2	Sample Flow for 2nd TP for 204 M2/8			204 [N]
TP Code	TP Name/Description	Std Version	Set Id	Proc Record																																																																			
4097out		ANA1_8		SRMHDR [N]																																																																			
INEND1911	Sample Flow TP for 1911/invoic	001911		INVOIC [N]																																																																			
INEND210	Sample Flow TP for M2_8/210	M2/8		210 [N]																																																																			
INEND210FA	Sample Flow FA for M2_8/210	M2/8		997 [N]																																																																			
INEND837	Sample Flow TP for 03032/837	003032		837 [N]																																																																			
INEND837FA	Sample Flow FA for 03032/837	003032		997 [N]																																																																			
INEND850	Sample Flow TP for 02040/850	002040		850 [N]																																																																			
INEND850FA	Sample Flow FA for 02040/997	002040		997 [N]																																																																			
OUTEND02856	Sample Flow for TP 856 Div 2	003030		856 [N]																																																																			
OUTEND03856	Sample Flow for TP 856 Div 3	003030		856 [N]																																																																			
OUTBOUND856	Sample Flow for TP 856	003030		856 [N]																																																																			
TDCC204-1	Sample Flow for 1st TP for 204 M2/8			204 [N]																																																																			
TDCC204-2	Sample Flow for 2nd TP for 204 M2/8			204 [N]																																																																			
4	<p>Select the Trading Partnerships you want to attach to the table by entering <b>y</b> in the Trading Partnership's Proc Record (Process Record) field.</p> <p><b>Note</b>                      The following shortcut keys are available:</p> <ul style="list-style-type: none"> <li>▶ To select every Trading Partnership code, press F2 (Mark All).</li> <li>▶ To remove all the <b>y</b>'s from the Proc Record field, press F3 (Unmark All).</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>																																																																						



<b>(Contd) Step</b>	<b>Action</b>
5	<p>Press F10 to continue.</p> <p><b>System Response</b>            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.</p> <div data-bbox="667 632 1390 863" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>           DS Table Attachment Log: //tmp/attachlog.881440995           -----           DS Table Attachment Log           -----           Date: 12/06/1997           Time: 15:46:29           -----           F1:Help F3:Erase F9:Quit           </pre> </div> <p><b>Reference</b>            See the <a href="#">The Screen Viewer</a> topic in the <a href="#">Monitoring Processes</a> chapter of this guide for information about navigating the log.</p>
6	<p>Check the <b>Status</b> field of the DS Table Attachment Log to verify that the Trading Partnership codes attached successfully to the table.</p> <p><b>Note</b>            The DS Table Attachment Log is a temporary file. When you leave this screen, Gentran:Server deletes the file.</p>
7	<p>Press ESC to exit the table.</p>



# 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.  <b>Reference</b> See the <a href="#">How to Display a Document Specifier Table</a> topic in this chapter for instructions.
2	Select <b>View Attached</b> and then press ENTER.  <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">View Attached</div>  <b>System Response</b> The system displays the Attached TPs screen, which lists the Trading Partnership codes linked to the table.  <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 10px auto;"> <pre> Attached TPs ----- TP Code      Description ----- INBND850    Sample Flow TP for 02040/850 INBND850FA  Sample Flow FA for 02040/997 ----- F9:Quit                     </pre> </div>
3	Press F9 to exit the <b>Attached TPs</b> screen.

(Continued on next page)



**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 <b>Doc Loc</b> from the host Util menu to display the Document Reference Number Specifier screen.																		
2	<p>Click <b>Unattached TPs</b> and then press ENTER.</p> <div data-bbox="911 653 1146 716" style="border: 1px solid black; padding: 2px; text-align: center;"> <p>Unattached TPs</p> </div> <p><b>System Response</b> The system displays the Unattached TPs screen. It lists all the Trading Partnership codes that are not linked to any table.</p> <div data-bbox="662 915 1386 1304" style="border: 1px solid black; padding: 10px;"> <p>Unattached TPs</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">TP Code</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td style="background-color: black; color: white;">4097out</td> <td></td> </tr> <tr> <td>INBND1911</td> <td>Sample Flow TP for 1911/invoic</td> </tr> <tr> <td>INBND210</td> <td>Sample Flow TP for M2_8/210</td> </tr> <tr> <td>INBND210FA</td> <td>Sample Flow FA for M2_8/210</td> </tr> <tr> <td>INBND837</td> <td>Sample Flow TP for 03032/837</td> </tr> <tr> <td>INBND837FA</td> <td>Sample Flow FA for 03032/837</td> </tr> <tr> <td>OUTBND02856</td> <td>Sample Flow for TP 856 Div 2</td> </tr> <tr> <td>OUTBND03856</td> <td>Sample Flow for TP 856 Div 3</td> </tr> </tbody> </table> <p>F9:Quit</p> </div>	TP Code	Description	4097out		INBND1911	Sample Flow TP for 1911/invoic	INBND210	Sample Flow TP for M2_8/210	INBND210FA	Sample Flow FA for M2_8/210	INBND837	Sample Flow TP for 03032/837	INBND837FA	Sample Flow FA for 03032/837	OUTBND02856	Sample Flow for TP 856 Div 2	OUTBND03856	Sample Flow for TP 856 Div 3
TP Code	Description																		
4097out																			
INBND1911	Sample Flow TP for 1911/invoic																		
INBND210	Sample Flow TP for M2_8/210																		
INBND210FA	Sample Flow FA for M2_8/210																		
INBND837	Sample Flow TP for 03032/837																		
INBND837FA	Sample Flow FA for 03032/837																		
OUTBND02856	Sample Flow for TP 856 Div 2																		
OUTBND03856	Sample Flow for TP 856 Div 3																		
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**

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.  <b>Reference</b> See the <a href="#">How to Display a Document Specifier Table</a> topic in this chapter.
2	Select <b>Detach Trading Partners</b> and press ENTER.  <div style="text-align: center; border: 1px solid black; width: fit-content; margin: 0 auto; padding: 5px;">Detach TPs</div>  <b>System Response</b> The system displays a list of all the Trading Partnership codes attached to the table.  <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <pre> Detach TPs ----- TP Code:      TP Desc:      Table Code:  Detach? ----- 1             x12 to fixed app DSTLB1      [N] 10            Edifact to x12: DSTLB1      [N] 10174        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 -----                     </pre> </div>  <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
3	<p>Select each Trading Partnership you want to detach from the table by entering <b>y</b> in the Detach field.</p> <p><b>Note</b> The following shortcut keys are available:</p> <ul style="list-style-type: none"><li>▶ To select every Trading Partnership code, press F2 (Mark All).</li><li>▶ To remove all the <b>y</b>'s from the Detach field, press F3 (Unmark All).</li></ul>
4	<p>Press F10 to continue.</p> <p><b>System Response</b> 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.</p>
5	<p>Check the Status field of the log to verify that the Trading Partnership codes were successfully removed from the table.</p>
6	<p>Press ESC to exit the table.</p>

---

## 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.  <b>Reference</b> To display a table, see <a href="#">How to Display a Document Specifier Table</a> .
2	Press TAB until the Desc field is selected.
3	Press F3 to delete the document specifier table.  <b>System Response</b> The system counts the number of Trading Partnership records that use the table and displays this information in a Confirmation prompt.  <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <pre> Confirmation Delete DS Table: Default EDI Default Mapping Are You Sure (y/n) ? _ </pre> </div>
4	Type <b>y</b> 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:

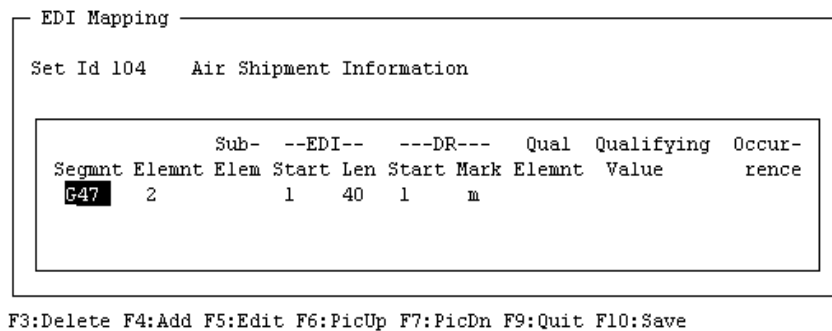
- ▶ EDI Mapping Screen
  - ▶ APP Mapping Screen
  - ▶ XML Mapping Screen
  - ▶ Transaction Code Mapping Screen
  - ▶ How to Display a Document Specifier Map
  - ▶ How to Change a Document Specifier Map
  - ▶ How to Delete a Map from a Table
-



# 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. <b>Example</b> 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. <b>Examples</b> 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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Function Max. Length Values</b>
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	<p>The length (in number of characters) of the unique value. This value can be any number from 1 to 40, inclusive.</p> <p><b>Comments</b> The system uses the full length unless it encounters a sub-element separator, an element separator, or a segment terminator (end of line).</p> <p><b>Note</b> The sum total length of the lines in the map cannot exceed 40.</p>
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	<p>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.</p> <p><b>Examples</b> %, &amp;, *</p>
Qual Elemnt	<p>The element number used with the qualifying value. This value helps determine when Gentran:Server uses the segment and element characters in the document reference number. This value can be any number from 1 to 999, inclusive.</p> <p><b>Comment</b> When the element with this number has the value specified in the Qualifying Value field, Gentran:Server extracts the specified characters. Otherwise, the system skips the characters.</p> <p style="text-align: right;"><b>(Continued on next page)</b></p>

<b>(Contd) Field</b>	<b>Function Max. Length Values</b>
Qualifying Value	The value that the element number in the Qual Elemnt field must have before 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	<p>The occurrence of the segment from which Gentran:Server extracts the specified characters. This value can be any number up to five characters in length.</p> <p><b>Comment</b> If this field is empty, Gentran:Server uses the last occurrence of the segment.</p> <p><b>Examples</b></p> <ul style="list-style-type: none"> <li>▶ 1 means use the first occurrence of the segment.</li> <li>▶ 2 means use the second occurrence.</li> <li>▶ 0 means use the last occurrence.</li> </ul>

### EDI Mapping screen function keys

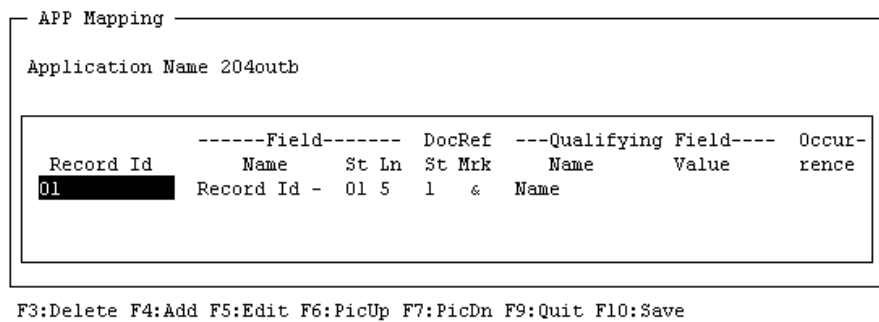
This table describes the function keys on the EDI Mapping screen and their functions.

<b>Key</b>	<b>Function</b>
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.  <b>Comment</b> The sum total length of the record lines in the map cannot exceed 40.

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
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	<p>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.</p> <p><b>Examples</b> %, &amp;, *</p>
Qualifying Field Name	<p>Defines the field name used with the qualifying value (Value field). These values help determine when 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.</p> <p><b>Comment</b> When the field with this name has the value specified in the Value field, Gentran:Server extracts the specified characters. Otherwise, Gentran:Server skips the characters.</p>
Value	Defines the value that the field named in the Qualifying Field Name field must have before Gentran:Server extracts the specified characters in the field. This value can be up to ten characters in length.
Occurrence	<p>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.</p> <p><b>Comment</b> If this field is empty, Gentran:Server uses the last occurrence.</p> <p><b>Examples</b></p> <ul style="list-style-type: none"> <li>▶ 1 means use the first occurrence of the segment.</li> <li>▶ 2 means use the second occurrence.</li> <li>▶ 0 means use the last occurrence.</li> </ul>

(Continued on next page)

**APP Mapping  
screen function  
keys**

This table describes the function keys on the APP Mapping screen and their functions.

<b>Key</b>	<b>Function</b>
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
DDF Name:  xmldata

XML Target Node      Use  ---XML---  --DocRef--
Transmission        y    1    15    1    X

F1:Zoom F3:Delete F4:Add F5:Edit F6:PicUp F7:PicDn F9:Quit F10:Save  6.0
Edit the map to define the document specification
  
```

### 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 <i>y</i> for <i>Yes</i> or <i>n</i> for <i>No</i> .  <b>Note</b> 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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
Len	<p>Defines the number of characters in the unique value. This value can be any number from 1 to 40.</p> <p><b>Comment</b> The total length of all record lines in the map cannot exceed 40.</p>
Doc Ref Start	<p>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.</p>
Visual Marker	<p>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.</p> <p><b>Comment</b> The system does not allow you to use the same marker more than once. If you already used the marker for another line, Gentran:Server displays a message. You must choose another marker.</p> <p><b>Reference</b> See <a href="#">How to Display a Map Picture</a>.</p> <p><b>Examples</b> % &amp; *</p>

### XML Mapping screen function keys

This table describes the function keys on the XML Mapping screen and their functions.

<b>Key</b>	<b>Function</b>
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.

(Continued on next page)



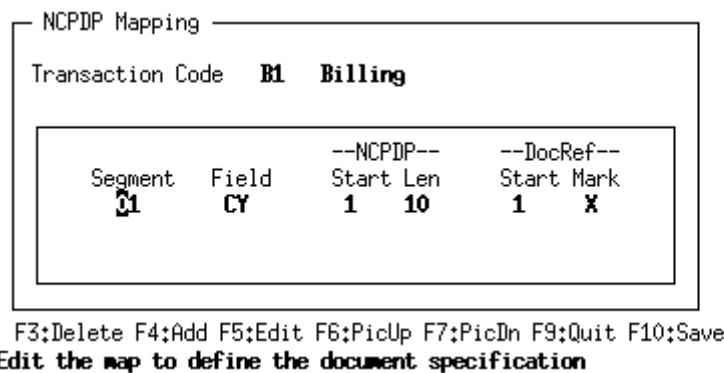
<b>(Contd) Key</b>	<b>Function</b>
F5	Displays the Edit screen for the selected line so that you can change the values in the line.
F6	Displays a pictorial representation of the map.
F7	Removes the map picture from the display.
F9	Exits the screen.
F10	Saves your changes to the map.

---

# 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.



6.0

## 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.  <b>Example</b> 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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
NCPDP Len	<p>The length (in number of characters) of the unique value. This value can be any number from 1 to 40, inclusive.</p> <p><b>Comments</b></p> <ul style="list-style-type: none"> <li>▶ The system uses the full length unless it encounters a separator or terminator character.</li> <li>▶ The sum total length of the lines in the map cannot exceed 40.</li> </ul>
DocRef Start	<p>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.</p>
Mark	<p>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.</p> <p><b>Examples</b> %, &amp;, *</p>

**NCPDP Mapping  
screen function  
keys**

This table describes the function keys on the NCPDP Mapping screen and their functions.

<b>Key</b>	<b>Function</b>
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.  <b>Reference</b> See the <a href="#">How to Display a Document Specifier Table</a> topic in this chapter for instructions.
2	Select Edit Maps and press ENTER.  <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;">                         Edit Maps                     </div> <p><b>System Response</b></p> Gentran:Server displays the Mapping screen for the table type. <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <pre> EDI Mapping Set Id ██████████            Sub-  --EDI--  ---DR---  Qual  Qualifying  Occur-           Segmnt Elemnt Elem Start Len Start Mark Elemnt Value  rence           -----</pre> </div> <p>F2:Select F5:Find F9:Quit</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>	
3	Select the <b>Set ID, Application Name, DDF Name, or Transaction Code</b> field and then press F5 to display a list of names mapped to the current table.  <b>CAUTION</b> <b>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.</b>	
	<b>IF you want to...</b>	<b>THEN...</b>
	To scroll the list	Use the cursor keys.
	Display a map	Select the set ID, application name, DDF name, or NCPDP transaction code; then press ENTER.
	<b>System Response</b> The system redisplay 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.  <b>WARNING</b> <b>You cannot edit the information on this screen. To change the values in the map, see the topic <a href="#">How to Change a Document Specifier Map</a> in this chapter.</b>	

# 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.  <b>Reference</b> See <a href="#">How to Display a Document Specifier Map</a> .
2	Press F4 to add a line to the map.  <b>System Response</b> The system displays the appropriate Add screen.  <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <pre> Add Segment Id      [REDACTED] Element # Sub-Elem # Elem Start      from Length Doc Ref Start Visual Marker Qual Elem # Qual Value Occurrence  F9:Quit  F10:Save</pre> </div> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
3	Complete the fields on the Add screen; then press F10 to save the line.  <b>System Response</b> The system adds the segment or record line and then clears the screen so that you can add another segment or record line.  <b>CAUTION</b> <b>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.</b>
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.

<b>Step</b>	<b>Action</b>
1	Display the map that you want to edit.  <b>Reference</b> See <a href="#">How to Display a Document Specifier Map</a> .
2	Select the line you want to edit and then press F5 to edit the line.  <b>System Response</b> The system displays the Edit screen.  (Continued on next page)

<b>(Contd) Step</b>	<b>Action</b>
3	<p>Change the fields on the Edit screen; then press F10 to save your changes.</p> <p><b>Note</b> You cannot change the values in the first three fields (Segment or Record ID, Element or Field #, and Sub-Elem#).</p> <p><b>System Response</b> The system adds the segment or record line and then clears the screen so that you can add another segment or record line.</p> <p><b>CAUTION</b> <b>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.</b></p>
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.

<b>Step</b>	<b>Action</b>
1	<p>Display the map that you want to edit.</p> <p><b>Reference</b> See <a href="#">How to Display a Document Specifier Map</a>.</p>
2	<p>Select the line you want to delete and then press F3 to delete the line.</p> <p><b>System Response</b> The system displays a confirmation prompt.</p>
3	<p>Type <b>y</b> at the prompt to confirm the deletion.</p> <p><b>System Response</b> Gentran:Server deletes the line.</p>
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, 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.  <b>Reference</b> See <a href="#">How to Display a Document Specifier Map</a> .
2	Select a line and then press F3 to delete the line.  <b>System Response</b> The system displays a confirmation prompt.  <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <pre> Confirmation Delete Segment: G47 Element: 2 Sub-Element: Are You Sure (y/n) ? _ </pre> </div>
3	Type <b>y</b> at the prompt to confirm the deletion.  <b>System Response</b> 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.  <b>System Response</b> Gentran:Server removes the document specifier map from the document specifier table.



---

# Using Transaction Registers to Track Documents

---

<b>Contents</b>	<b>Overview</b>
	▶ Introduction ..... 2
	▶ Transaction Registers ..... 3
	▶ Transaction Register Screen ..... 6
	▶ Transaction Register Search Screen ..... 8
	<b>Working with a Single Transaction Register</b>
	▶ How to Search for an Entry ..... 9
	▶ How to Add an Entry ..... 11
	▶ How to Delete an Entry ..... 13
	▶ How to Delete Multiple Entries ..... 14
	<b>Working with Multiple Transaction Registers</b>
	▶ How to Add a Transaction Register ..... 18
	▶ How to Delete a Transaction Register ..... 20

---

# Overview

## Introduction

---

**In this chapter** This chapter describes transaction registers and explains how to work with them.

---

**Key terms** This table lists the key terms used in this chapter.

Term	Description
clean_trn	The clean transaction register utility, which is used to delete transaction register entries older than a specified date and time.
default transaction register	The transaction register that a data manager uses unless you specify a different directory for the transaction register in the data manager's initialization file.
deltrn	The transaction register delete utility, which is used to delete transaction register entries.
document reference number	The unique number that Gentran:Server assigns to each data set or document to track the movement of the data set or document through the system.
Trading Partnership code	A user-defined code that uniquely identifies a Trading Partnership record.
transaction register	An indexed file used to keep track of documents that Gentran:Server handles.

---

# Transaction Registers

---

**Description** A **transaction register** is an indexed file used to keep track of the documents that pass through Gentran:Server. It has five fields:

- ▶ Trading Partnership code or file name
- ▶ Document reference number
- ▶ Unique ID
- ▶ Input or output direction character
- ▶ Date and time.

---

**When to use a transaction register**

You use the transaction register to:

- ▶ Verify that a data manager handled a transaction
- ▶ Check for duplicate transactions
- ▶ Record or delete a particular transaction activity.

---

**Multiple transaction registers**

Normally, you have only one transaction register; however, you can add alternate transaction registers and delete them when you no longer need them.

**When to use**

Use multiple transaction registers when you want data manager flows to be independent of each other. Multiple transaction registers allow identical transactions to pass through more than one data manager flow without registering as a duplicate.

---

**Default transaction register**

If you do not have a transaction register, Gentran:Server creates one the first time you select **Trn Reg** from the host Utility menu. The default transaction register resides in the \$EDI\_ROOT directory. It is named *trn.dat/trn.idx*.

The data managers use the default transaction register unless you specify a different directory for the transaction register in the data manager's initialization file. The initialization file parameter that establishes the directory is TRANS\_REGISTER\_DIR.

**Reference**

See the [Maintaining Initialization Files](#) chapter in this guide for information about the data manager's initialization file.

---

(Continued on next page)

### How entries are made

If your system is configured to use a transaction register, a data manager creates an entry in the transaction register each time it processes a set or file successfully, unless the entry is a duplicate. This table describes how the number of entries for a document is determined.

IF the data manager...	THEN the data manager...
Can recognize sets in a document (inbd, appm, or dnld personality)	Writes one entry for each set in the file to a transaction register, regardless of the routing method. Each set must have a unique document reference number.
Can't recognize the sets in a document (file or xltr personality)	Writes one entry to the transaction register for the entire file.
Is a host command card (hcmd) personality	Writes one entry for each host command card in the file.

### Examples

- ▶ An inbound data manager writes all sets to the transaction register.
- ▶ A file data manager writes one entry to the transaction register for the entire file.

### How duplicates are handled

You set the way a data manager handles duplicate entries in the Duplicates Checking Parameters of the data manager's initialization file. This table describes the ways data managers can handle duplicate entries.

IF the data manager...	THEN it...
Does not allow duplicates (ALLOW_DUPLICATES parameter is set to 0)	<ul style="list-style-type: none"> <li>▶ Flags the document as a duplicate</li> <li>▶ Generates a duplicate error message</li> <li>▶ Does not write the entry to the transaction register.</li> </ul>
Allows duplicates and updates transaction record (ALLOW_DUPLICATES parameter is set to 1)	Overwrites the date and time of the previously received data with the duplicate data's date and time.
Allows duplicates but does not write to transaction register (ALLOW_DUPLICATES parameter is set to 2)	Does not write to or update the transaction record.

(Continued on next page)

### Duplicates in routing method

When checking for duplicate entries, the application (appm), download (dnld), and inbound (inbd) data managers also check for duplicate sets within the routing method. This table explains how these data managers deal with duplicate sets.

IF...	AND...	THEN the data manager...
The routing method is group or interchange	A duplicate document reference number exists in the same group or interchange	Generates a duplicate error message for the routing method.
The routing method is group or interchange	One or more sets is a duplicate	Does not write any of the sets in the file to the transaction register and routes the entire file to the error directory.
A duplicate exists in the same group or interchange	ROUTING_METHOD is set	Routes only the set that is a duplicate to the error directory and routes the other sets normally.

## Transaction Register Screen

**Introduction** The Transaction Register screen enables you to:

- ▶ Search for a transaction entry
- ▶ Add one or more transaction entries to the register
- ▶ Delete one or more entries.

**Illustration** This illustration shows an example Transaction Register screen.

Transaction Register		
TP Code	Document Reference Number	IO Date Time
INBOUND850	00431200	i 199711181122
INBOUND850	01431200	i 199711181122
INBOUND850	01631200	i 199711181122
INBOUND850	01831200	i 199711181122
INBOUND850	01931200	i 199711181122
INBOUND850	02031200	i 199711181122
INBOUND850	02131200	i 199711181122
INBOUND850	02231200	i 199711181122
INBOUND850	02331200	i 199711181122
INBOUND850	02431200	i 199711181122
OUTBOUND856	07	i 199711161807

F2:Add F3:Del F4:First F5>Last F6:Scch F7:Next F8:Prev F9:Quit

### Transaction Register screen fields and functions

This table describes the fields of the Transaction Register screen and their functions.

Field	Function
TP Code/Filename	Displays the Trading Partnership code or file name (depending on the data manager's personality) for the transaction entry.
Document Reference Number	Displays the document reference number for the transaction entry.
IO	Displays the input or output code: <ul style="list-style-type: none"> <li>▶ Inbound = i</li> <li>▶ Outbound = o</li> </ul>
Date Time	Displays the date and time that the data manager handled the transaction. The format is CCYYMMDDHHMM.

(Continued on next page)



**CAUTION**

The Unique ID field is not displayed on the Transaction Register screen. To view the contents of the Unique ID field, you must use the isops command, `isops -u -f trn`.

**Reference**

See the [isops](#) topic in the [Command Reference](#) chapter of the *Gentran: Server for UNIX and Workstation Technical Reference Guide* for information about the isops command.

**Transaction Register screen function keys**

This table describes the function keys of the Transaction Register screen.

Key	Function
F2	Adds an entry to the transaction register.
F3	Deletes the selected transaction register entry.
F4	Displays the first group of 500 transaction register entries that matched the search criteria.
F5	Displays the last group of 500 transaction register entries that matched the search criteria.
F6	Displays a search screen that enables you to enter search criteria and search for a transaction entry.
F7	Displays the next group of 500 transaction register entries that matched the search criteria.
F8	Displays the previous group of 500 transaction register entries that matched the search criteria.
F9	Exits the Transaction Register screen.

## Transaction Register Search Screen

**Introduction** The transaction register Search screen is used to enter search criteria that you want Gentran:Server to use to find one or more transaction entries.

**Illustration** This illustration shows the transaction register Search screen with search data entered into the fields.

Search

INBOUND850 01831200 i

F9:Quit F10:Search

### Transaction Register Search screen fields and functions

This table describes the fields of the transaction register Search screen and their functions.

Field	Function
TP Code	Defines the Trading Partnership code for the transaction entries. Required.
Document Reference Number	Defines the document reference number for the transaction entry. Optional.
IO	Defines the input or output code. Optional. <ul style="list-style-type: none"> <li>▶ Inbound = i</li> <li>▶ Outbound = o</li> </ul>

### Transaction Register Search screen function keys

This table describes the function keys of the Transaction Register screen.

Key	Function
F9	Exits the Search screen.
F10	Starts the search.

# Working with a Single Transaction Register


## How to Search for an Entry

**Introduction** To verify that a data manager handled a transaction, you can search for the entry in the transaction register.

### Searching for a transaction register entry

Use this procedure to search for a transaction register entry.

Step	Action
1	Select <b>Util</b> from the host main menu.
2	<p>Select <b>Trn Reg</b> from the Utility menu.</p> <p><b>System Response</b> Gentran:Server displays the Select menu, which lists the available transaction registers.</p> <p><b>Comment</b> If the default register does not exist, Gentran:Server creates an empty one when you select <b>Trn Reg</b>.</p>
3	<p>Select the name of the transaction register you want to search and press ENTER.</p> <p><b>System Response</b> Gentran:Server displays the Transaction Register screen.</p> <pre> Transaction Register ----- TP Code      Document Reference Number      IO Date Time ----- INBOUND850  00431200                       i 199711181122 INBOUND850  01431200                       i 199711181122 INBOUND850  01631200                       i 199711181122 INBOUND850  01831200                       i 199711181122 INBOUND850  01931200                       i 199711181122 INBOUND850  02031200                       i 199711181122 INBOUND850  02131200                       i 199711181122 INBOUND850  02231200                       i 199711181122 INBOUND850  02331200                       i 199711181122 INBOUND850  02431200                       i 199711181122 OUTBOUND856  07                             i 199711161807  F2:Add F3:Del F4:First F5:Last F6:Srch F7:Next F8:Prev F9:Quit </pre> <p>(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
4	<p>Press F6 to start the search.</p> <p><b>System Response</b> Gentran:Server displays the Search screen.</p> 
5	<p>Complete the search fields.</p> <p><b>WARNING</b> <b>You must specify a value in the TP Code field to perform a search. The Document Reference Number and IO fields are optional</b></p> <p><b>Reference</b> See the <a href="#">Transaction Register Search Screen</a> topic in this chapter for a description of the fields and functions.</p>
6	<p>Press F10 to start the search.</p> <p><b>System Response</b> The system searches for and displays all entries that match the search criteria you enter on the Search screen. The matches are displayed in groups of 500.</p>
7	<p>Locate the entry.</p> <p><b>Comment</b> Use the arrow keys to scroll through the group of entries. Use the function keys to display another group of entries that match your search criteria.</p>


## How to Add an Entry

### Introduction

To account for an entry that was received through other than normal means or to test detection of duplicates, you can manually add an entry to a transaction register.

### Adding a transaction entry

Use this procedure to add an entry to a transaction register.

Step	Action
1	Select <b>Util</b> from the host main menu.
2	Select <b>Trn Reg</b> from the Utility menu.  <b>System Response</b> Gentran:Server displays the Select menu, which lists all the directories that contain transaction registers.  <b>Comment</b> If the default register does not exist, Gentran:Server creates it when you select <b>Trn Reg</b> from the Select menu.
3	Select the name of the transaction register you want to work with and press ENTER.  <b>System Response</b> Gentran:Server displays the Transaction Register screen.
4	Press F2 to add an entry.  <b>System Response</b> Gentran:Server displays the Add screen.   <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
5	Complete the fields. <ul style="list-style-type: none"><li>▶ The first field is the Trading Partnership code or file name.</li><li>▶ The second field is the document reference number.</li><li>▶ The third field is the direction (<b>i</b> inbound or <b>o</b> for outbound).</li></ul> <b>Note</b> Gentran:Server supplies the current date and time. You cannot override this value.
6	Press F10 to save the entry.

---

## How to Delete an Entry

### Introduction

When you test your data flow, you can induce transaction entries that you do not want to keep in the transaction register. This topic explains how to delete these unwanted entries.

### CAUTION

**You cannot recover deleted transaction entries.**

### Deleting a transaction register entry

Use this procedure to delete a transaction register entry.

Step	Action
1	<p>Search the transaction register for the entry you want to delete.</p> <p><b>Reference</b> See the <a href="#">How to Search for an Entry</a> topic in this chapter for instructions.</p>
2	<p>Select the entry you want to delete; then press F3 to delete the entry.</p> <p><b>System Response</b> Gentran:Server displays a Delete confirmation prompt.</p> <pre> Delete ----- INBOUND850      00431200      i, y/n? ----- F9:Quit </pre>
3	<p>Type <b>y</b> to confirm the deletion.</p> <p><b>System Response</b> Gentran:Server deletes the entry and removes it from the Transaction Register screen.</p>

## How to Delete Multiple Entries

---

**Introduction** Gentran:Server has two purge utilities to mark multiple transaction register entries for deletion:

- ▶ `deltrn`
- ▶ `clean_trn`.

---

**Used with isops** The `deltrn` and `clean_trn` programs mark entries for deletion, but do not reduce the size of the `trn.dat` and `trn.idx` files. To actually remove the entries, you use these programs to mark the entries and then use the **isops** utility to unload and reload the records you want to keep.

---

**Ways to run the utilities** To automate the purge process, we suggest that you run these utilities from a cleanup facility, such as the **longterm** script. You can also run them from the command line.

### Reference

See the [Command Reference](#) chapter in the *Gentran:Server for UNIX and Workstation Technical Reference Guide* for more information about **deltrn** and **clean\_trn**.

For information about modifying the **longterm** script, see the [Archiving Your Data](#) chapter in this guide.

---

**The deltrn utility** The transaction register delete utility, **deltrn**, has three operating options:

- ▶ Mark a specific entry based on the Trading Partnership code and document reference number. You can also include the data flow direction and the relative path of the transaction register, if you are using an alternate transaction register.
- ▶ Mark entry for a file that failed translation according to the translation return code
- ▶ Mark all the entries with a unique ID number less than a specified value.

---

(Continued on next page)



**deltrn command  
line format  
options**

This table lists the format options for the **deltrn** utility.

<b>IF you want to mark for deletion...</b>	<b>THEN use this format...</b>
A specific entry for a Trading Partnership code and document reference number	<pre>deltrn -t&lt;Trading Partnership code&gt; -d&lt;document reference number&gt; [-x&lt;direction&gt;] [-l&lt;transaction register&gt;]</pre> <p><b>Comment</b> The -x and -l arguments are optional.</p>
An entry for a file that failed translation	<pre>deltrn -a&lt;translator audit file&gt;</pre> <p><b>Example</b> An inbound data manager routes a file for translation and writes one entry for each set in the file to a transaction register. If a translation error occurs, <b>deltrn</b> uses the audit file, <i>edistat.i</i> or <i>edistat.o</i>, to identify the inbound data manager's entry associated with the translation error and remove it from the transaction register.</p>
All entries with unique ID numbers less than a specified ID	<pre>deltrn -u&lt;unique ID number&gt;</pre>

**deltrn arguments**

This table defines the **deltrn** arguments.

<b>Argument</b>	<b>Definition</b>
-t<Trading Partnership code>	The user-defined code that uniquely identifies a Trading Partnership record
-d<document reference number>	The unique number that Gentran:Server assigns to each data set or document to track the movement of the data set or document through the system.

(Continued on next page)

(Contd) Argument	Definition
-x<direction>	The direction of the data in the process flow. <ul style="list-style-type: none"> <li>▶ Inbound = i</li> <li>▶ Outbound = o</li> </ul>
-l<transaction register>	The relative path of the transaction register (from EDI_ROOT). Omit this argument if the transaction register is in EDI_ROOT.
-a<translator audit record>	The translator audit file, <i>edistat.i</i> or <i>edistat.o</i> . The deltrn program gets file information from <i>edistat.i</i> or <i>edistat.o</i> to remove the entry from the transaction register when the return code from translation indicates that translation failed.  <b>Example</b> deltrn -aedistat.i
-u<unique ID number>	The unique ID that Gentran:Server appended to a destination file name. The deltrn program deletes entries with unique ID numbers less than this number.

**clean\_trn utility**

The clean transaction register utility, **clean\_trn**, flags transaction register entries for deletion that are older than a specified date and time.

**Running clean\_trn from the command line**

To run the clean transaction register utility from the command line, enter the command in this format:

```
clean_trn -d <CCYYMMDD[HHMMSS]>
```

**Comment**

The time component (HHMMSS) is optional.

**Example 1**

In this example, the command flags for deletion transaction register entries older than August 1, 1997.

```
clean_trn -d 19970801
```

(Continued on next page)



**Example 2**

In this example, the command flags for deletion transaction register entries older than September 5, 1997, at 2:00 p.m.

```
clean_trn -d 19970905140000
```

**Procedure**

Use this procedure to mark items for deletion and then remove them from a transaction register.

Step	Action
1	<p>Make a backup copy of <i>trn.dat</i> and <i>trn.idx</i>.</p> <p><b>Example</b></p> <pre>cp trn.dat trndat.old ; cp trn.idx trnidx.old</pre>
2	<p>Run the <b>clean_trn</b> or the <b>deltrn</b> utility to mark the records that you want to delete.</p>
3	<p>Run the <b>isops</b> utility to unload the records that are NOT marked for deletion.</p> <p><b>Example</b></p> <pre>isops -u -f trn &gt; &lt;newtrn&gt;</pre> <p>where &lt;newtrn&gt; is the name of the file that the unload process creates.</p> <p><b>Comment</b></p> <p>The isops utility unloads the records you want to retain.</p> <p><b>Reference</b></p> <p>See the <i>Gentran:Server Technical Reference Guide</i> for information about <b>isops</b>.</p>
4	<p>Remove or delete the <i>trn.dat</i> and <i>trn.idx</i> files.</p> <p><b>Example</b></p> <pre>rm trn.dat ; rm trn.idx</pre>
5	<p>Use the <b>isops</b> utility to rebuild the <i>trn.dat</i> and <i>trn.idx</i> files from the file you created in Step 3.</p> <p><b>Example</b></p> <pre>cat &lt;newtrn&gt;   isops -l -f trn</pre> <p>where &lt;newtrn&gt; is the name of the file that the <b>isops</b> unload process created.</p>

# Working with Multiple Transaction Registers

## How to Add a Transaction Register

---

**Introduction**    You can use more than one transaction register in your system. However, each transaction register must reside in a separate directory. Also, you must perform maintenance activities, such as purging entries, on each transaction register separately.

---

**Data managers check initialization files**    A data manager uses the default transaction register unless you specify a different directory in the TRANS\_REGISTER\_DIR parameter of the data manager's initialization file.

---

**About this topic**    This procedure explains how to add a new transaction register and the directory in which it resides. After you create the transaction register and directory, enter the directory name into the initialization file of each data manager that will use the new transaction register.

---

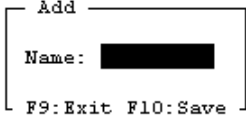
**When to use**    Use this procedure when you want a data manager to use a transaction register other than the default transaction register.

---

(Continued on next page)

## Adding a transaction register

Use this procedure to add a transaction register to your system.

Step	Action
1	Select <b>Util</b> from the host Main Menu.  <b>System Response</b> Gentran:Server displays the Utility menu.
2	Select <b>Trn Reg</b> from the Utility menu.  <b>System Response</b> Gentran:Server displays the Select menu, which lists all the directories that contain transaction registers.
3	Press F2 to add a transaction register.  <b>System Response</b> Gentran:Server displays the Add screen.  
4	Type the name of the new transaction register's directory in the <b>Name</b> field.
5	Press F10 to create the new transaction register.  <b>System Response</b> Gentran:Server: <ul style="list-style-type: none"> <li>▶ Creates the directory under EDI_ROOT</li> <li>▶ Adds the <i>trn.dat</i> and <i>trn.idx</i> files to the directory.</li> </ul>

## The next step

To have a data manager use the new transaction register to log entries and check for duplicates, you must enter the transaction register's name in the TRANS\_REGISTER\_DIR parameter of the data manager's initialization file.

### Reference

See the [Maintaining Initialization Files](#) chapter in this guide for information about setting the TRANS\_REGISTER\_DIR parameter.

## How to Delete a Transaction Register

---

**Introduction** If you no longer need an alternate transaction register, you can delete it.

---

**Cautions** **Before you delete a transaction register, observe these cautions:**

- Do not delete a transaction register when data manager is using it; this causes the data manager to halt.
- Deleting a transaction register may permit the system to process a duplicate entry.
- Do not delete the default transaction register.

---

**Before you begin** Update the TRANS\_REGISTER\_DIR parameter in a data manager's initialization file before deleting the transaction register. A data manager will not start if its transaction register does not exist.

**Deleting a transaction register**

---

Use this procedure to delete a transaction register.

Step	Action
1	Select <b>Util</b> from the host Main Menu.  <b>System Response</b> Gentran:Server displays the Utility menu.
2	Select <b>Trn Reg</b> from the Utility menu.  <b>System Response</b> Gentran:Server displays the Select menu, which lists all the directories that contain transaction registers.  <span style="color: red;">(Continued on next page)</span>

(Contd) Step	Action
3	<p>Select the directory that contains the transaction register you want to delete and then press F3 to delete the transaction register.</p> <p><b>System Response</b> Gentran:Server displays the <b>Delete</b> confirmation prompt.</p> <pre data-bbox="883 579 1172 695"> Delete _____ tranout y/n? █ F9:Exit _____ </pre>
4	<p>Type <b>y</b> at the prompt to confirm the deletion.</p> <p><b>System Response</b> Gentran:Server deletes the <i>trn.dat</i> and <i>trn.idx</i> files from the directory and removes the directory name from the list of transaction registers, but does not delete the directory itself.</p>
5	<p>Do you want to delete the directory that held the transaction register?</p> <ul style="list-style-type: none"> <li>▶ If YES, go to the UNIX command line and delete the directory.</li> <li>▶ If NO, skip this step.</li> </ul> <p><b>WARNING</b> <b>Do not delete the \$EDI_ROOT directory!</b></p> <p><b>Reference</b> See your UNIX manuals for instructions on how to delete a directory.</p>





---

# Setting Up Life Cycle

---

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	2
	▶ Database Software .....	4
	▶ Life Cycle Components .....	6
	<b>Life Cycle Tables</b>	
	▶ Life Cycle Table lc221 .....	8
	▶ Life Cycle Table: Informix .....	10
	▶ Life Cycle Table: Oracle .....	14
	▶ Life Cycle Table: Sybase .....	18
	▶ LCDestInfo Table .....	22
	<b>Life Cycle Configuration</b>	
	▶ Overview .....	24
	▶ Life Cycle Configuration Process .....	25
	▶ How to Set Up Life Cycle for Oracle .....	28
	▶ How to Set Up Life Cycle for Informix .....	34
	▶ How to Set Up Life Cycle for Sybase .....	37
	▶ Data Manager Settings for Life Cycle Activities .....	40
	<b>Testing Life Cycle Setup</b>	
	▶ How to Test Outbound Processing .....	43
	▶ How to Test Inbound Processing .....	45
	▶ Troubleshooting Life Cycle .....	47

---

# 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 Gentran:Server's 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 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.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
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.
lclld	The 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 <b>lclld</b> and <b>xlld</b> , 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 Gentran:Server command line program that enables you to run a statistical report on the translation traffic.
translation audit files	The event files, <i>edistat.i</i> and <i>edistat.o</i> , that the translator produces. These files are also referred to as temporary audit files or status records.
xlld	The Gentran:Server program that updates the Life Cycle table records that <b>lclld</b> loaded with information from the translation audit files.

## Database Software

**Introduction** To take advantage of Gentran:Server's 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 Gentran:Server, but you must have it to view your Life Cycle event records from 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

Gentran:Server's Life Cycle auditing facility interfaces with:

- Informix, versions 5.x, 6.0, 7.0 and higher
- Oracle, versions 7.1 and higher

Support for Oracle databases does not include support for the Oracle Exadata platform.

- 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

---

(Continued on next page)

---

**Location of the  
database  
software**

The database software may reside on:

- The same machine as Gentran:Server  
or
- Another machine on the same network as the UNIX host that contains 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 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 `lcl`.

---

**Event file is loaded to database** When you configure your system to use a relational database to audit your Gentran:Server activity, 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 *Gentran:Server for UNIX and Workstation Technical Reference Guide* for a description of the layout of the Life Cycle table.

---

(Continued on next page)

---

<b>LCDestInfo table</b>	The LCDestInfo table is a subordinate table to the Life Cycle table. The LCDestInfo table holds destination file routing information for ODBC translation audit records.
<b>Load program</b>	The Life Cycle load program, <b>lclid</b> , loads event records to the Life Cycle database table.
<b>Translation audit files</b>	<p>The Gentran:Server translation process generates two types of audit files: <i>edistat.i</i> (inbound translation) and <i>edistat.o</i> (outbound translation). Translation also produces <i>edistat.o</i> for inbound translations if translation creates functional acknowledgments. The Life Cycle process uses information in <i>edistat.i</i> and <i>edistat.o</i> to update event records in the Life Cycle table.</p> <p><b>Reference</b> See the <i>Gentran:Server for UNIX and Workstation Application Integration User's Guide</i> for more information about the audit files produced during translation.</p>
<b>Update program</b>	The Life Cycle update program, <b>xlld</b> , uses information in the translation audit files to update records that <b>lclid</b> loaded to the Life Cycle table. The <b>xlld</b> program also updates the LCDestInfo 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. 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 [Life Cycle Configuration Process](#) topic in this chapter 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.

---

(Continued on next page)



**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 **lcl** 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 **lcl** 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 <ul style="list-style-type: none"> <li>▶ i = inbound</li> <li>▶ o = outbound</li> <li>▶ x = other</li> </ul>
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)

(Continued on next page)

<b>(Contd) Name</b>	<b>Type</b>	<b>Description</b>
SFIL	VARCHAR(60)	Source file name (drop-off name)
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: <ul style="list-style-type: none"> <li>▶ A = Accept</li> <li>▶ P = Partially accepted</li> <li>▶ R = Reject</li> <li>▶ E = Accepted with errors</li> <li>▶ N = Inbound: no acknowledgment generated outbound: no acknowledgment expected</li> <li>▶ Y = Inbound: acknowledgment to be generated, outbound: acknowledgment expected</li> </ul>
SEGCNT	VARCHAR(9)	Segment count
CHARCNT	VARCHAR(9)	Character count
MAILGROUP	VARCHAR(9)	(Reserved for Advanced Data Distribution mailbox operations)

(Continued on next page)

**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 <ul style="list-style-type: none"> <li>▶ i = inbound</li> <li>▶ o = outbound</li> <li>▶ x = other</li> </ul>
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

(Continued on next page)

<b>(Contd) Name</b>	<b>Type</b>	<b>Description</b>
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 <ul style="list-style-type: none"> <li>▶ i = inbound</li> <li>▶ o = outbound</li> <li>▶ x = other</li> </ul>
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)

(Continued on next page)

(Contd) Name	Type	Description
SFIL	VARCHAR2(60)	Source file name (drop-off name)
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 <ul style="list-style-type: none"> <li>▶ A = Accept</li> <li>▶ P = Partially accepted</li> <li>▶ R = Reject</li> <li>▶ E = Accepted with errors</li> <li>▶ N = inbound: No acknowledgment generated, outbound: No acknowledgment expected</li> <li>▶ Y = inbound: Acknowledgment to be generated, outbound: Acknowledgment expected</li> </ul>
SEGCNT	VARCHAR2(9)	Segment count
CHARCNT	VARCHAR2(9)	Character count
MAILGROUP	VARCHAR2(9)	(Reserved for Advanced Data Distribution mailbox operations)

(Continued on next page)

**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 <ul style="list-style-type: none"> <li>▶ i = inbound</li> <li>▶ o = outbound</li> <li>▶ x = other</li> </ul>
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  (Continued on next page)



<b>(Contd) Name</b>	<b>Type</b>	<b>Description</b>
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 <ul style="list-style-type: none"> <li>▶ i = inbound</li> <li>▶ o = outbound</li> <li>▶ x = other</li> </ul>
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)

(Continued on next page)

<b>(Contd) Name</b>	<b>Type</b>	<b>Description</b>
SFIL	CHAR(60)	Source file name (drop-off name)
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 <ul style="list-style-type: none"> <li>▶ A = accept</li> <li>▶ P = Partially accepted</li> <li>▶ R = Reject</li> <li>▶ E = Accepted with errors</li> <li>▶ N = inbound: No acknowledgment generated outbound: No acknowledgment expected</li> <li>▶ Y = inbound: Acknowledgment to be generated, outbound: Acknowledgment expected</li> </ul>
SEGCNT	CHAR(9)	Segment count
CHARCNT	CHAR(9)	Character count
MAILGROUP	CHAR(9)	(Reserved for Advanced Data Distribution mailbox operations)

---

(Continued on next page)

**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 <ul style="list-style-type: none"> <li>▶ i = inbound</li> <li>▶ o = outbound</li> <li>▶ x = other</li> </ul>
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  (Continued on next page)

<b>(Contd) Name</b>	<b>Type</b>	<b>Description</b>
DestEncrypt	CHAR(2)	Reserved for future use (Encryption Flag)
DestCmp	CHAR(2)	Reserved for future use (Compression Flag)

---

## LCDestInfo Table

**Introduction** The **LCDestInfo table** is a subordinate table to the parent lc221 Life Cycle table. The table holds destination file routing information for ODBC translation audit records. The **xlld** update program loads data to this table.

**The LCDestInfo table** This table describes the columns in the LCDestInfo table.

**Note**

The character type of the LCDestInfo table fields differs among the databases:

- ▶ Informix uses VARCHAR
- ▶ Oracle uses VARCHAR2
- ▶ Sybase uses CHAR.

Field	Characters	Description
ActivitySysKey	16	Future use
tp	36	Trading Partnership code (expanded)
doc	81	Document reference number
iox	2	Direction of the data (i=in, o=out, x=other)
mailbagid	10	Mailbag ID
DestType	4	Destination type (future use)
DestHost	128	Destination host name
DestID	16	Destination file unique key
DestUser	128	Destination user (future)
DestDir	60	Destination directory name
DestFile	128	Destination file name
DestCharCnt	16	Destination character count
DestRecCnt	16	Destination record count
DestResult	3	Translation result

(Continued on next page)

<b>(Contd) Field</b>	<b>Characters</b>	<b>Description</b>
DestContentType	125	File type (binary, EDI, etc.)
DestContentSubType	125	Future use
DestDesc	80	Description or subject
DestEncrypt	2	Encryption flag (future use)
DestCmp	2	Compression flag (future use)

---

# 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.

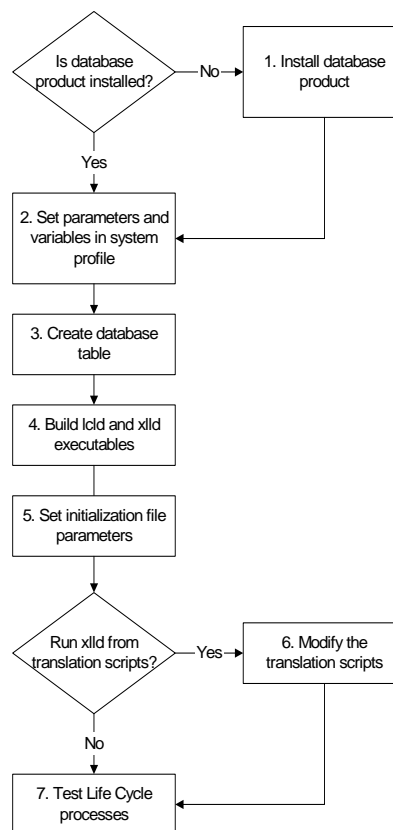
## Reference

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.



(Continued on next page)

**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	<p>Is the database product installed?</p> <ul style="list-style-type: none"> <li>▶ If YES, continue to Task 2.</li> <li>▶ If NO, install the database product on the same machine as Gentran:Server or on another machine that is on the same network as the machine that has Gentran:Server.</li> </ul> <p><b>Reference</b> See the installation instructions for your database product.</p>
2	<p>Configure the Gentran:Server system profile for database paths and environment variables.</p> <p><b>Reference</b> See the appropriate topic in this chapter for instructions:</p> <ul style="list-style-type: none"> <li>▶ <a href="#">How to Set Up Life Cycle for Oracle</a></li> <li>▶ <a href="#">How to Set Up Life Cycle for Informix</a></li> <li>▶ <a href="#">How to Set Up Life Cycle for Sybase</a></li> </ul>
3	<p>Create the Life Cycle database table. Grant write and update access privileges for the database table to any user who has permission to activate Gentran:Server data managers. Create a unique table index.</p> <p><b>Reference</b> See the appropriate topic in this chapter for instructions:</p> <ul style="list-style-type: none"> <li>▶ <a href="#">How to Set Up Life Cycle for Oracle</a></li> <li>▶ <a href="#">How to Set Up Life Cycle for Informix</a></li> <li>▶ <a href="#">How to Set Up Life Cycle for Sybase</a></li> </ul>
4	<p>Build the Life Cycle load and update programs, <b>lclld</b> and <b>xllld</b>.</p> <p><b>Reference</b> See the appropriate topic in this chapter for instructions.</p> <ul style="list-style-type: none"> <li>▶ <a href="#">How to Set Up Life Cycle for Oracle</a></li> <li>▶ <a href="#">How to Set Up Life Cycle for Informix</a></li> <li>▶ <a href="#">How to Set Up Life Cycle for Sybase</a></li> </ul> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Task</b>	<b>Description</b>
5	<p>Edit the initialization file of each data manager (agent) in your flows to set the Life Cycle parameters.</p> <p><b>Reference</b> See the <a href="#">Data Manager Settings for Life Cycle Activities</a> topic in this chapter for the parameter values.</p> <p>See the <a href="#">Maintaining Initialization Files</a> chapter in this guide for instructions on modifying a data manager's initialization file.</p>
6	<p>Do you want to run <b>xlld</b> from a translation script?</p> <ul style="list-style-type: none"> <li>▶ If YES, modify the translation script to run <b>xlld</b>. Make the <b>xlld</b> command run AFTER the <b>prep_xlldfile</b> command and AFTER translation has occurred. We recommend that you run <b>xlld</b> after <b>ediarc</b>. Also, make sure that the path to the translation audit file is correct in the script.</li> <li>▶ If NO, continue with Task 7.</li> </ul> <p><b>Reference</b> See the <a href="#">Working with Scripts</a> chapter in this guide for information about modifying scripts.</p> <p><b>Note</b> When choosing when to run <b>xlld</b>, keep in mind that <b>xlld</b> cannot access and update a Life Cycle record if the <b>lclld</b> process has not finished. You will get an SQL error if <b>xlld</b> cannot locate the complete record. This problem most often affects sites that process large amounts of data.</p>
7	<p>Test the Life Cycle processes.</p> <p><b>Reference</b> See the <a href="#">Testing Life Cycle Setup</a> section in this chapter for instructions.</p>

# 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. <b>WARNING</b> 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 <i>bin</i> directory. <b>Reference</b> See your Oracle installation documentation for instructions.
	If you use Gentran:Server on the RS6000 with AIX 4.x, make sure you have a minimum of revision 6.2 of <b>xlld.pc</b> . <b>How to check your revision number</b> To check your revision number, change to the \$EDI_ROOT/src/oracle directory and enter this command: <pre>what xlld.pc</pre> If you do not have a minimum of revision 6.2 , contact Gentran:Server product support. <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Done</b>	<b>Task</b>
	<p>If you use 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.</p> <p><b>Note</b> Gentran:Server uses both the UNIX ANSI C compiler and the <b>Pro*C</b> compiler to build <b>xlld</b>.</p> <p><b>How to check your version number</b></p> <ol style="list-style-type: none"> <li>1. To find the UNIX ANSI C compiler, enter this command: <b>which cc</b></li> <li>2. Change directories to the directory displayed by the previous command.</li> <li>3. Enter the following command: <b>what cc</b></li> </ol> <p><b>System response</b> The system displays the version number.</p>
	<p>Check the path to the database linker program.</p> <p><b>How to check the linker</b> Enter the following command: <b>which ld</b></p> <p>If the resulting path is <i>/usr/uscb</i> or <i>/usr/uscb/bin</i>, then you need to edit the path so that the following path appears first: <i>/usr/ccs/bin</i></p>

---

(Continued on next page)

**Procedure** Use this procedure to configure your system to use an Oracle Life Cycle table.

Step	Action
1	<p>Configure Gentran:Server system profile to contain the following:</p> <ul style="list-style-type: none"> <li>▶ The path to Oracle <i>bin</i> directory</li> <li>▶ The Oracle environment variables.</li> <li>▶ Set ORACLE_HOME. This is mandatory.</li> <li>▶ Set ORACLE_SID if the Oracle database is on a different UNIX machine that Gentran:Server.</li> <li>▶ You may need to set ORAENV_ASK, TWO_TASKS, or ORACLE_LPPROG, depending on your Oracle environment.</li> <li>▶ We recommend that you put the Oracle libraries in the shared library path: SHLIB_PATH (HP), LIBPATH (AIX) or LD_LIBRARY_PATH (Solaris). Also add \$EDI_ROOT/odbc/lib to this path.</li> </ul> <p><b>Reference</b> Refer to your Oracle documentation, or ask your database administrator for the correct values for the environment variables.</p>
2	<p>Add the environment variables from Step 1 to the profile of the owner of the Gentran:Server Foreground Manager.</p> <p><b>Comment</b> This step ensures that the database user has access to both the Gentran:Server environment and the database environment.</p>
3	<p>Make sure that the Oracle binaries are in \$PATH.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
4	<p>Do you want to use the default configuration?</p> <ul style="list-style-type: none"> <li>▶ If YES, <ul style="list-style-type: none"> <li>— Grant connection to one instance of the database table for both Gentran:Server's login ID and the Gentran:Server administrator's ID.</li> </ul> </li> </ul> <p><b>Example</b> Grant connection privileges to OPS\$&lt;edi_server&gt;, identified by password, where &lt;edi_server&gt; is the Server account name. This creates a link from the UNIX account to the database account.</p> <p><b>Reference</b> See your Oracle documentation for information about passwords.</p> <ul style="list-style-type: none"> <li>— Grant resource privileges to Gentran:Server's login ID (for example, grant resource to OPS\$&lt;edi_server&gt;, where &lt;edi_server&gt; is the Server account name).</li> <li>— Continue with Step 3.</li> </ul> <ul style="list-style-type: none"> <li>▶ If NO, GO TO to Step 6.</li> </ul>
5	<p>Log in as the administrative user you created to own security.</p> <p><b>Reference</b> See the <a href="#">How to Install Security Administration Software</a> topic in the <a href="#">Installing Gentran:Server On the UNIX Host</a> chapter in the <i>Gentran:Server for UNIX Getting Started Guide</i>.</p>
6	<p>Execute the following command to create the Life Cycle database table:</p> <pre>sqlplus / @./src/oracle/mksrvdb221</pre> <p><b>System Response</b> The system displays these messages:</p> <pre>Table created Index created Grant succeeded</pre> <p><b>Tip</b> If you want only one record per document, create a unique key using the fields <b>tp</b>, <b>doc</b>, <b>iox</b>, and <b>seq</b> as the key.</p>
7	<p>If you did not define the database account, or if the last step resulted in errors, execute the following command:</p> <pre>sqlplus userid/password @./src/oracle/mksrvdb221</pre> <p style="text-align: right;"><b>(Continued on next page)</b></p>

<b>(Contd) Step</b>	<b>Action</b>
8	Copy the executable <code>\$EDI_ROOT/bin/load</code> to <code>\$EDI_ROOT/bin/lcl.d</code> .
9	<p>Ensure that the shell scripts <b>lcl.d.sh</b> and <b>lcl.d.ctl</b> are copied to the <code>EDI_ROOT/bin</code> directory and that <b>lcl.d.sh</b> has execute permissions.</p> <p><b>WARNING</b></p> <p><b>Please use the attached lcl.d.sh and lcl.d.ctl examples. They are designed to eliminate a potential concurrency problem that could cause files to be overwritten.</b></p>
10	<p>Copy the Oracle <b>make</b> file from the source file in the Oracle environment into <code>\$EDI_ROOT/src/oracle</code>.</p> <p><b>Note</b> For Oracle versions 7.x, the <b>make</b> file is named <i>proc.mk</i>. For Oracle versions 8.0 or higher, the <b>make</b> file is named <i>demo_proc.mk</i>.</p> <p><b>CAUTION</b></p> <p><b>The exact source location of this file depends upon the Oracle version and release. Check your Oracle documentation or ask your database administrator.</b></p>
11	Change to the <code>\$EDI_ROOT/src/oracle</code> directory.
12	<p>Do you have ORACLE version 7.3 or higher?</p> <ul style="list-style-type: none"> <li>▶ If YES: <ul style="list-style-type: none"> <li>— Open the <i>proc.mk</i> or <i>demo_proc.mk</i> file.</li> <li>— Search for the string “SAMPLES=sample1” and replace it with the string “SAMPLES=xlld”. (Do not type the quotes.)</li> </ul> </li> <li>▶ If NO, continue with Step 13.</li> </ul> <p style="text-align: right;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>											
13	Execute the command to create the translation load and FA update program, <b>xlld</b> .  <b>Note</b> <b>xlld</b> has optional parameters U and P: <ul style="list-style-type: none"> <li>▶ -U &lt;Oracle userID&gt;</li> <li>▶ -P &lt;Oracle password&gt;</li> </ul> <table border="1" data-bbox="623 642 1427 995"> <thead> <tr> <th data-bbox="623 642 954 730"><b>IF you have this version of ORACLE...</b></th> <th data-bbox="954 642 1427 730"><b>THEN use this command, entered on one line...</b></th> </tr> </thead> <tbody> <tr> <td data-bbox="623 730 954 821">7.1 or 7.2</td> <td data-bbox="954 730 1427 821">make -f proc.mk USERID=user/ password objs=xlld.o exe=xlld</td> </tr> <tr> <td data-bbox="623 821 954 877">7.3 through 7.3.2</td> <td data-bbox="954 821 1427 877">make -f proc.mk xlld</td> </tr> <tr> <td data-bbox="623 877 954 934">7.3.3 or 7.3.4</td> <td data-bbox="954 877 1427 934">make -f proc.mk xlld LLIBXA=</td> </tr> <tr> <td data-bbox="623 934 954 995">8.0 or higher</td> <td data-bbox="954 934 1427 995">make -f demo_proc.mk SAMPLES</td> </tr> </tbody> </table>		<b>IF you have this version of ORACLE...</b>	<b>THEN use this command, entered on one line...</b>	7.1 or 7.2	make -f proc.mk USERID=user/ password objs=xlld.o exe=xlld	7.3 through 7.3.2	make -f proc.mk xlld	7.3.3 or 7.3.4	make -f proc.mk xlld LLIBXA=	8.0 or higher	make -f demo_proc.mk SAMPLES
<b>IF you have this version of ORACLE...</b>	<b>THEN use this command, entered on one line...</b>											
7.1 or 7.2	make -f proc.mk USERID=user/ password objs=xlld.o exe=xlld											
7.3 through 7.3.2	make -f proc.mk xlld											
7.3.3 or 7.3.4	make -f proc.mk xlld LLIBXA=											
8.0 or higher	make -f demo_proc.mk SAMPLES											
14	Move the <b>xlld</b> executable to the <i>\$EDI_ROOT/bin</i> directory.											
15	Set the Life Cycle parameters in the data manager's initialization file.  <b>Reference</b> See the <a href="#">Data Manager Settings for Life Cycle Activities</a> topic in this chapter for the parameter values.  See the <a href="#">Maintaining Initialization Files</a> chapter in this guide for instructions on modifying a data manager's initialization file.											
16	Modify the translation script if you want to run <b>xlld</b> from the translation script.  <b>Reference</b> See the <a href="#">Working with Scripts</a> chapter in this guide for information about modifying scripts.											

## 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.  <b>Reference</b> See your Informix installation documentation for additional information.
2	Configure Gentran:Server's system profile to contain the following: <ul style="list-style-type: none"> <li>▶ Path to the Informix <i>bin</i> directory.</li> <li>▶ Informix environment variables TERMCAP and INFORMIXDIR.</li> <li>▶ Set SHLIB_PATH (HP), LIBPATH (AIX) or LD_LIBRARY_PATH (Solaris) to the path to shared database libraries. Also add \$EDI_ROOT/odbc/lib to this path.</li> <li>▶ Other environment variables that your database administrator identifies.</li> </ul>
3	Add the environment variables from Step 2 to the profile of the owner of the Gentran:Server Foreground Manager.
4	Make sure that the Informix binaries are in \$PATH.  <span style="color: red;">(Continued on next page)</span>

<b>(Contd) Step</b>	<b>Action</b>						
5	<p>Change to the <code>\$EDI_ROOT/src/infx</code> subdirectory and execute the <b>make</b> command, <b>makeinfx</b>.</p> <p><b>CAUTION</b>  <b>The makeinfx command located in the infx directory is configured to build xlld, lcld and mksrvdb.</b></p> <p><b>Example</b>  <code>make -f makeinfx</code></p> <p><b>Tip</b>                      If you want only one record per document in the Life Cycle table, create a unique key using the fields <b>tp</b>, <b>doc</b>, <b>iox</b>, and <b>seq</b> as the key.</p>						
6	<p>After you have successfully built <b>xlld</b>, <b>lcld</b>, and <b>mksrvdb</b>, move the resulting executable files to <code>\$EDI_ROOT/bin</code>.</p>						
7	<p>Log in as the administrative user you created to own security.</p> <p><b>Reference</b>                      See the <a href="#">How to Install Security Administration Software</a> topic in the <a href="#">Installing Gentran:Server On the UNIX Host</a> chapter in the <i>Gentran:Server for UNIX Getting Started Guide</i>.</p>						
8	<p>Use this table to determine the appropriate command.</p> <table border="1" data-bbox="610 1226 1425 1398"> <thead> <tr> <th data-bbox="610 1226 1019 1281"><b>IF your Informix version is...</b></th> <th data-bbox="1019 1226 1425 1281"><b>THEN run...</b></th> </tr> </thead> <tbody> <tr> <td data-bbox="610 1281 1019 1335">6.0 and above, openview</td> <td data-bbox="1019 1281 1425 1335">mksrvdb6</td> </tr> <tr> <td data-bbox="610 1335 1019 1398">5.x, 6.0, 7.0 standard engine</td> <td data-bbox="1019 1335 1425 1398">mksrvdb4</td> </tr> </tbody> </table> <p><b>System Response</b>                      The system displays the following messages:                      Create Database: edisrv2, sqlcode=0                      Create Table: edisrv2: lc221, sqlcode=0                      Finished</p> <p style="text-align: right;"><i>(Continued on next page)</i></p>	<b>IF your Informix version is...</b>	<b>THEN run...</b>	6.0 and above, openview	mksrvdb6	5.x, 6.0, 7.0 standard engine	mksrvdb4
<b>IF your Informix version is...</b>	<b>THEN run...</b>						
6.0 and above, openview	mksrvdb6						
5.x, 6.0, 7.0 standard engine	mksrvdb4						

<b>(Contd) Step</b>	<b>Action</b>
9	<p>Set the Life Cycle parameters in the data manager's initialization file.</p> <p><b>Reference</b> See the <a href="#">Data Manager Settings for Life Cycle Activities</a> topic in this chapter for the parameter values.</p> <p>See the <a href="#">Maintaining Initialization Files</a> chapter in this guide for instructions on modifying a data manager's initialization file.</p>
10	<p>Modify the translation script if you want to run <b>xlld</b> from the translation script.</p> <p><b>Reference</b> See the <a href="#">Working with Scripts</a> chapter in this guide for information about modifying scripts.</p>

---

# 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.  <b>Reference</b> See your Sybase installation documentation.
2	Configure Gentran:Server's system profile to contain the following: <ul style="list-style-type: none"> <li>▶ Path to Sybase <i>bin</i></li> <li>▶ Sybase environment variables DSQUERY, SYBASE, SYBROOT, and SYBPLATFORM</li> <li>▶ Set SHLIB_PATH (HP), LIBPATH (AIX) or LD_LIBRARY_PATH (Solaris) to the path to shared database libraries. Also add <code>\$EDI_ROOT/odbc/lib</code> to this path.</li> </ul> <b>Note</b> The valid values for SYBPLATFORM are: <pre>sun_svr4      (Sun) dce_sun_svr4 (Sun using DCE) rs6000       (AIX) dce_rs6000   (AIX Thread Safe) dce_hp800    (HP Thread Safe)</pre>
3	Add the environment variables from Step 2 to the profile of the owner of the Gentran:Server Foreground Manager.  <div style="text-align: right;"><i>(Continued on next page)</i></div>

<b>(Contd) Step</b>	<b>Action</b>
4	Make sure that the Sybase binaries are in \$PATH.
5	Change to the <code>\$EDI_ROOT/src/sybase</code> subdirectory.
6	Copy these files from <code>\$SYBASE/sample/esqlc/</code> to <code>\$EDI_ROOT/src/sybase</code> : <ul style="list-style-type: none"> <li>▶ <code>makefile</code></li> <li>▶ <code>sybsqlx.h</code></li> <li>▶ <code>sybopts.sh</code></li> </ul>
7	Globally change “sample 1” to “lclld” and “sample 2” to “xlld” in <code>makefile</code> .  <b>Note</b> Do not type the quotes.
8	Build the <b>lclld</b> and <b>xlld</b> executable files by running the <b>make</b> command.  <b>Example</b>  <pre>make makefile lclld, make makefile xlld</pre>
9	Change “lclld” to “mksrvdb221” and run the <b>make</b> command again to build the database table.  <b>Tip</b> If you want only one record per document in the Life Cycle table, create a unique key using the fields <b>tp</b> , <b>doc</b> , <b>iox</b> , and <b>seq</b> as the key.
10	After you have successfully built <b>xlld</b> , <b>lclld</b> , and <b>mksrvdb221</b> , move the executable files to <code>\$EDI_ROOT/bin</code> .
11	Log in as the administrative user you created to own security.  <b>Reference</b> See the <a href="#">How to Install Security Administration Software</a> topic in the <a href="#">Installing Gentran:Server On the UNIX Host</a> chapter in the <i>Gentran:Server for UNIX Getting Started Guide</i> .
12	Execute the following command:  <pre>mksrvdb221 -U&lt;userid&gt; -P&lt;password&gt;</pre> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
13	<p>Set the Life Cycle parameters in the data manager's initialization file.</p> <p><b>Reference</b> See the <a href="#">Data Manager Settings for Life Cycle Activities</a> topic in this chapter for the parameter values.</p> <p>See the <a href="#">Maintaining Initialization Files</a> chapter in this guide for instructions on modifying a data manager's initialization file.</p>
14	<p>Modify the translation script if you want to run <b>xlld</b> from the translation script.</p> <p><b>Reference</b> See the <a href="#">Working with Scripts</a> chapter in this guide for information about modifying scripts.</p>

---

## Data Manager Settings for Life Cycle Activities

### Introduction

You must set several parameters in the Routing section of the data manager's initialization file to use the Life Cycle facility with a data manager. This topic explains which parameters to set.

### Reference

For instructions on working with a data manager's initialization file, see the [Maintaining Initialization Files](#) chapter in this guide.

### Initialization file parameters

This table lists the initialization file parameters that you must set for the data manager.

Parameter	Function
LIFE_CYCLE_DIR	<p>Defines the full path name of the directory in which the Life Cycle event files are placed for processing. The default program is <b>lclid</b>.</p> <p><b>Example</b>  <code>LIFE_CYCLE_DIR \$EDI_ROOT/bin/xltr2</code></p>
MULTIPLE_LIFE_CYCLE_FILES	<p>Determines whether each set, group, or interchange has its own Life Cycle file.</p> <ul style="list-style-type: none"> <li>▶ 1 = Yes</li> <li>▶ 0 = No</li> </ul> <p><b>Comment</b>            If you set this parameter to 1 (the default), <b>lclid</b> 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 and run <b>lclid</b> on a scheduled basis.</p> <p>If you want to run <b>lclid</b> from the <code>:LIFE_CYCLE_EXEC_LINE</code>, set this parameter to 0 (No).</p> <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Parameter</b>	<b>Function</b>
LIFE_CYCLE_CALL_SWITCH	<p>Determines the point at which the Life Cycle command is run.</p> <ul style="list-style-type: none"> <li>▶ 0 = Execute at the end of each routing pass</li> <li>▶ 1 = Execute at the end of each input file</li> <li>▶ 2 = Execute at the end of each process cycle</li> </ul>
LIFE_CYCLE_EXEC_LINE	<p>Defines the Life Cycle command to be executed.</p> <p><b>Example</b></p> <pre>LIFE_CYCLE_EXEC_LINE lclld -f&lt;dmm&gt; -U&lt;user&gt; -P&lt;password&gt;</pre> <p><b>Note</b> You must enter this command on the line that follows the label name.</p> <p><b>WARNING</b> <b>If you omit this parameter, Gentran:Server does not call the Life Cycle load program.</b></p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Parameter</b>	<b>Function</b>
USE_RECON_IDS	<p>For application (appm) and download (dnld) data managers.</p> <p>Determines which values are entered in Life Cycle event records:</p> <ul style="list-style-type: none"><li>▶ The extracted values</li><li>▶ The alternate reconciliation IDs you entered in the Inbound Acknowledgment tab of the Trading Partnership dialog box.</li></ul> <p><b>When to use</b> 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.</p> <p><b>Values</b></p> <ul style="list-style-type: none"><li>▶ 1 = Yes</li><li>▶ 0 = No</li></ul> <p>The default is 0.</p> <p><b>Reference</b> See the <i>Gentran:Server for UNIX and Workstation Application Integration User's Guide</i> for more information.</p>

---

# 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 must:

- ▶ Create the Life Cycle table
- ▶ Configure the data managers in the outbound process flow
- ▶ Configure the translator.

---

**Procedure** Use this procedure to test outbound processing.

Step	Action
1	Pass a data file through the application data manager (source agent).  <b>WARNING</b> <b>Make sure the data 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.</b>
2	After the application data manager has processed the data, check the data manager's log file in the Life Cycle directory to make sure that <b>lcl.d.sh</b> has added the transaction set records to the Life Cycle database.
3	Perform a translation on the outbound file.
4	Copy the <i>edistat.o</i> file to <i>xlto.v</i> .

(Continued on next page)

<b>(Contd) Step</b>	<b>Action</b>
5	Run the Life Cycle update program <b>xlld</b> .  <b>Example</b> <code>xlld -fxlto</code>
6	Check the <b>xlld</b> log file to make sure that <b>xlld</b> updated the Life Cycle records from the application data manager with the translation date/time, control numbers, and Set ID.

---

# 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 data managers and other components of the inbound process flow
- ▶ Configure the translator.

If you are using an Oracle database, you must also make sure that **lclid.sh** is configured for a minimum of read/execute (-rwxr-x----) permissions.

**Procedure** Use this procedure to test the Life Cycle facility for inbound processing.

Step	Action
1	Pass an EDI file through the inbound data manager (source agent).  <b>WARNING</b> <b>Make sure the location file used during inbound processing matches the location file used to create the translation map table.</b>
2	After the inbound data manager has processed the data, check the inbound data manager's log file in the Life Cycle directory to make sure that <b>lclid.sh</b> has added the transaction set records to the Life Cycle database.
3	Run translation on the inbound file.
4	Copy the <i>edistat.i</i> file to <i>xlti.v</i> .

(Continued on next page)

<b>(Contd) Step</b>	<b>Action</b>
5	<p>Run the Life Cycle update program <b>xlld</b>.</p> <p><b>Example</b></p> <pre>xlld -fxlti</pre> <p><b>CAUTION</b></p> <p><b>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 Gentran:Server archive program (ediarc) updates its archive file.</b></p>
6	<p>Check the <b>xlld</b> log file to make sure that <b>xlld</b> updated the Life Cycle records from the inbound data manager with the translation date/time, control numbers, Set ID, and acceptance status.</p>

---

## Troubleshooting Life Cycle

Use this checklist to resolve Life Cycle errors.

Error type	Action
All	Check the version of the database. Is the version supported by Life Cycle?
Connect error	<p>Check permissions.</p> <p>Does the owner of the Gentran:Server Foreground Manager have access to the database you use for Life Cycle?</p> <p>To what group does the owner of the Foreground Manager belong? Is the person using Gentran:Server in the same group as the owner of the Foreground Manager?</p>
The <b>xlld</b> or <b>lclid</b> run directory contains files named <i>&lt;data_manager_name&gt;.v</i> <i>.&lt;datetime&gt;</i>	<p>Check the database to determine if it is full.</p> <p>Check the disk space to make sure it is sufficient.</p> <p>Examine the data manager's log to determine why the load process failed and make necessary corrections.</p>
<b>lclid</b> failed to load the Life Cycle table	<p>Completely clear the Life Cycle table and then run <b>lclid</b> again to reload the table.</p> <p>If this resolves the problem, the Life Cycle table may have been corrupted.</p>





---

# Tracking Data with Life Cycle Files

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	3
	▶ Understanding Life Cycle Records .....	5
	▶ The Life Cycle Facility .....	8
	▶ The Life Cycle Table .....	9
	▶ LCDestInfo Table .....	10
	<b>The Life Cycle Load Programs</b>	
	▶ Overview .....	11
	▶ The lclid and xlld Life Cycle Process .....	13
	▶ The lclid Program .....	15
	▶ How to Run lclid from the Command Line .....	17
	▶ The xlld Program .....	18
	▶ How xlld Updates the Life Cycle Table .....	19
	▶ How xlld Updates the ODBC LCDestInfo Table .....	21
	▶ How xlld Handles Functional Acknowledgments .....	23
	▶ How xlld Handles Update Failures .....	26
	▶ The xlld Log File .....	27
	▶ How to Run xlld from the Command Line .....	28
	▶ How Life Cycle Handles Inbound Database Audit Records .....	30
	<b>Life Cycle Reports</b>	
	▶ Overview .....	31
	▶ The Translation Activity Report .....	33
	▶ The FA Exception Report .....	35
	▶ The FA Due Report .....	37
	▶ The TP List Report .....	39
	▶ The Translation Traffic Report .....	41
	<b>Running Life Cycle Reports</b>	
	▶ Overview .....	44
	▶ How to Run Informix Life Cycle Reports .....	46

- ▶ How to Run Oracle Life Cycle Reports . . . . . 48
  - ▶ How to Run the Translation Traffic Report . . . . . 49
-

# 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 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.
lclld	The Gentran:Server executable program that loads new Life Cycle event files to the Life Cycle database table.
lclld.sh	The Gentran:Server shell script that loads data manager's Life Cycle event records to the Life Cycle table.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
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 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 <b>lclid</b> and <b>xlld</b> . 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.
ODBC translation audit file	The Life Cycle audit file that the translator produces for inbound database translations.
tracker	The Gentran:Server command line program that enables you to run a statistical report on the translation traffic.
translation audit files	The translation audit files, <i>edistat.i</i> , <i>edistat.o</i> , and <i>dbaudit.i</i> that the translator produces. These files are also referred to as temporary audit files or status records.
xlld	The Gentran:Server program that updates the Life Cycle table with information from translation audit files, including functional acknowledgments. The <b>xlld</b> program updates Life Cycle event records that already exist.

# Understanding Life Cycle Records

---

**Introduction** Gentran:Server processes generate two types of Life Cycle records:

- ▶ Life Cycle event records
- ▶ Translation audit records.

If you use Gentran:Server's 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, 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 the Gentran:Server Advanced Data Distribution product) 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 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 **lcl**.

---

(Continued on next page)

**Name of Life Cycle event file**

This table describes how the Life Cycle event file is named.

IF the...	THEN the Life Cycle event file name is...
<p>MULTIPLE_LIFE_CYCLE_FILES parameter in the data manager's initialization file is set to 0</p>	<p>The data manager's name with a ".v" suffix.</p> <p><b>Example</b> If the MULTIPLE_LIFE_CYCLE_FILES parameter is set to 0, the dnld data manager produces a Life Cycle event file named <i>dnld.v</i>.</p>
<p>MULTIPLE_LIFE_CYCLE_FILES parameter in the data manager's initialization file is set to 1</p>	<p>The data manager's name with a ".v" suffix followed by a system-generated unique ID.</p> <p><b>Example</b> If the MULTIPLE_LIFE_CYCLE_FILES parameter is set to 1, the dnld data manager produces a Life Cycle event file for each set, each of which is named with a system-generated unique ID (<i>dnld.v.&lt;uniqueid&gt;</i>).</p> <ul style="list-style-type: none"> <li>▶ dnld.v.839781</li> <li>▶ dnld.v.839782</li> <li>▶ dnld.v.839883</li> </ul>
<p>The database is full, disk space is insufficient, or the Life Cycle program fails for any other reason</p>	<p>The data manager's name with a ".v" suffix followed by the date and time that the file was created.</p> <p><b>Note</b> This is a temporary file.</p> <p><b>Example</b> If the Life Cycle process fails on December 31, 1999, the dnld data manager produces a Life Cycle event file named <i>dnld.v.991231032310</i>. The date format is YYMMDDHHMMSS.</p>

(Continued on next page)

---

**Translation audit files**

The 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 *Gentran:Server for UNIX and Workstation Application Integration User's 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 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 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 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 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 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 Gentran:Server, the administrator used a Gentran:Server script or program named **mksrvdb** to create a database table called the Life Cycle table. The same program created a subordinate table to hold information resulting from inbound database translations.

## 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 ESQ-L-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 Gentran:Server's Life Cycle facility.

---

## LCDestInfo Table

---

**Introduction** The LCDestInfo table is a subordinate table to the parent lc221 Life Cycle table. The LCDestInfo table holds destination file routing information for ODBC translation audit records.

---

**How the Life Cycle table is created** Your organization creates the LCDestInfo 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.

---

**Contents** The LCDestInfo table holds destination file routing information, such as the destination host name, directory name, and file name.

---

**Joined to the Life Cycle table** The LCDestInfo table is joined to the parent Life Cycle table with these fields:

- ▶ tp
- ▶ doc
- ▶ iox.

# 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:

- ▶ lclid
- ▶ xlld.

**Functions** This table describes the functions of the **lclid** and **xlld** programs.

This program...	Does this...	With information from the...
lclid	Loads NEW Life Cycle events records	Inbound, download, file, application, and host command card data managers.
xlld	Updates EXISTING Life Cycle records that already exist in the database	Translation audit files ( <i>edistat.i</i> and <i>edistat.o</i> ) that the translator produces.
xlld	Loads destination file routing information into the LCDestInfo table	Translation ODBC audit record ( <i>dbaudit.i</i> )

(Continued on next page)

---

**Load failure**

If the database is full or if the database disk space is insufficient, the **xlld** and **lcl** cannot update or load Life Cycle event records to the database table. In this case,

- **lcl** 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.
  - **lcl** program writes a message to the data manager log files to record the unsuccessful load attempt.
-

## The lclid and xlld Life Cycle Process

**Introduction** The **lclid** and **xlld** programs update the Life Cycle database table in different ways.

**Process** This table describes how **lclid** and **xlld** are called in the Life Cycle process.

Stage	Description
1	<p>The data manager processes all files in its work directory and produces a Life Cycle event file named <code>&lt;data_manager_name&gt;.v</code> or <code>&lt;data_manager_name&gt;.v.&lt;uniqueID&gt;</code>.</p> <p><b>Note</b> A <code>&lt;data_manager_name&gt;.v</code> file contains all records for all the sets. A <code>&lt;data_manager_name&gt;.v.&lt;uniqueID&gt;</code> file contains an event record for one interchange, group, or set, according to the routing method.</p>
2	<p>The data manager passes its Life Cycle event file to the Life Cycle load programs, <b>lclid</b>.</p>
3	<p>The <b>lclid</b> program attempts to load NEW Life Cycle event files to the database table.</p> <p><b>Note</b> If <b>lclid</b> fails, it saves the data in a temporary file named <code>&lt;data_manager_name&gt;.v.&lt;datetime&gt;</code></p>
4	<p>The program <b>xlld</b> attempts to update records that already exist in the database with information from the translation audit files (<code>edistat.i</code>, <code>edistat.o</code>).</p> <p><b>Note</b> The <b>xlld</b> program handles inbound functional acknowledgments the same any other record.</p> <p><b>Reference</b> See <a href="#">How xlld Handles Functional Acknowledgments</a> for information.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Stage	Description	
5	<b>IF functional acknowledgment information is...</b>	<b>THEN xlld updates...</b>
	Present	These columns: <ul style="list-style-type: none"> <li>▶ The Functional Acknowledgment Date/ Time (FADT)</li> <li>▶ The Functional Acknowledgment Request Flag (FAREQ).</li> </ul> <p><b>Comment</b> The detail depends on the relationship established with the trading partner.</p>
	Not present	These columns: <ul style="list-style-type: none"> <li>▶ The three control numbers (Interchange, Application, and Set)</li> <li>▶ The Translation Date and Translation Time</li> <li>▶ The Functional Acknowledgment Request Flag.</li> </ul>
6	<p>If the database is full, disk space is insufficient, or the load fails for any other reason, the <b>lclld</b> and <b>xlld</b> processes:</p> <ul style="list-style-type: none"> <li>▶ Fail</li> <li>▶ Save the data in a temporary file. The temporary file is <i>&lt;data_manager_name&gt;.v.&lt;datetime&gt;</i>, where <i>data_manager_name</i> is the name of the data manager and <i>datetime</i> is the date and time that the temporary file was created.</li> <li>▶ Write a message to the data manager's log file to record the event.</li> </ul> <p>The <b>lclld</b> program also writes a message to the data manager log files to record the load failure.</p> <p><b>CAUTION</b> <b>For information about unsuccessful lclld 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.</b></p>	

## The IclD Program

---

**Introduction** The **IclD** program loads new data manager Life Cycle event records to the Life Cycle table.

---

**How IclD is called** How Gentran:Server calls **IclD** depends on whether or not your database is on the same machine as Gentran:Server.

This table describes how Gentran:Server calls **IclD**.

IF the database is...	AND you...	THEN...
On the same machine as Gentran:Server	—	The <b>IclD</b> 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 <b>IclD</b> 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 Gentran:Server, your database administrator must install SQL network services to link the machines so that IclD and xIld can find the Life Cycle database table.**

(Continued on next page)

**The lclid run log**

The **lclid** 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 `lclid` 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 **lclid** processing and any errors that occurred.

**Example**

This is a sample line from an **lclid** run log:

```
lclid:Lifecycle:OK=0:dup=0:other=3
```

**lclid versions**

Each database uses a different version of **lclid**. This table describes the **lclid** version, companion files, and location.

Database	lclid program version	Location
Informix	<b>lclid.ec</b> , which is an embedded SQL C program  <b>lc221.per</b> , which describes the order and length of the fields in the <code>lc221</code> table	<code>\$EDI_ROOT/src/infx</code>
Oracle	<b>lclid.sh</b> , which is a shell script  <b>lclid.ctl</b> , which describes the order in which fields appear in the <code>lc221</code> table	<code>\$EDI_ROOT/src/oracle</code>
Sybase	<b>lclid.cp</b> , which is an embedded SQL C program	<code>\$EDI_ROOT/src/sybase</code>



# How to Run lclld from the Command Line

---

**Introduction** Normally, Gentran:Server runs **lclld** as part of the data manager processing. You can also run the **lclld** program from the command line.

---

**Format** This is the command line format:

```
lclld -f<file name>
```

where <file name> is the path name of the data manager event file without the .v extension.

**Example**

```
lclld -flclld/edii
```

**Reference**

See the [Command Reference](#) chapter in the *Gentran:Server for UNIX and Workstation 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 and the LCDestInfo table. The **xlld** program can update just the Life Cycle table, just the LCDestInfo table, or both tables.

---

**Translation script renames files** The model translation scripts that came with your 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 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 a Gentran:Server support representative for other ways to call **xlld**, such as from a file data manager.

---

**Linking labels** You can link the labels DestDir, DestDesc, DestHost, and DestUser to any other labels in the ODBC initialization file by setting them to the corresponding item in the alias file.

### **Example**

If you specify:

DestDesc = Description

in your alias file, the program searches for the <Description> entry in the ODBC initialization file and assigns the value of Description to DestDesc.

---

## 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: <ul style="list-style-type: none"> <li>▶ The Functional Acknowledgment Date/Time (FADT)</li> <li>▶ The Functional Acknowledgment Request Flag (FAREQ)</li> <li>▶ Other columns, depending on the relationship established with the trading partner. See the <a href="#">How xlld Handles Functional Acknowledgments</a> topic.</li> </ul>
Not present	These columns: <ul style="list-style-type: none"> <li>▶ The three control numbers (Interchange Code, Application Code, and Set Identifier)</li> <li>▶ Translation date (tdt)</li> <li>▶ Translation time (ttm)</li> <li>▶ The Functional Acknowledgment Request Flag (for outbound only).</li> </ul>

### CAUTION

**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.**

(Continued on next page)

**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 Updates the ODBC LCDestInfo Table

### Introduction

The **xlld** program updates the LCDestInfo Life Cycle table under certain conditions.

### Conditions for LCDestInfo table updates

If the translator produces an ODBC translation audit file, **xlld** inserts a record in the LCDestInfo table for every record it reads from the ODBC translation audit record only if:

- ▶ The xlld command includes values for the `-o<ODBC_trans_audit_file>`, `-i<ODBC_init_file>`, and `-s<ODBC_data_source>` arguments
- ▶ The record exists in the Life Cycle table.

### Sources of LCDestInfo information

The Life Cycle update program **xlld** derives translation and transaction detail from the:

- ▶ ODBC translation audit file (*dbaudit.i*)
- ▶ Initialization file (*ODBC.in*)
- ▶ xlld command arguments.

This table describes the source of the information for the table's fields.

Source	Item name	LCDestInfo field
ODBC translation audit file	tpcode	tp
	docref	doc
	filename	DestFile
	character count	DestCharCnt
	record count	DestRecCnt
	result	DestResult

(Continued on next page)

<b>(Contd) Source</b>	<b>Item name</b>	<b>LCDestInfo field</b>
Initialization file ( <i>ODBC.ini</i> )	Destination Directory/ Database name	DestDir
	Driver Type	DestDesc
	Database Server Name	DestHost
	Database User ID	DestUser
Command line arguments	Mailbagid extracted from event file name	mailbagid

---

## 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 present in the translation audit record (*edistat.o*), 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.**

(Continued on next page)

---

**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**

**Gentran:Server combines the FA date and time into one field (FADT) before xlld updates the record.**

---

**How xlld locates the correct record**

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”.

---

(Continued on next page)



**Method 2**

The xlld program uses the criteria described in this table to find the appropriate Life Cycle record:

<b>IF the Trading Partner Code field of the audit record is...</b>	<b>AND...</b>	<b>THEN xlld performs a...</b>	<b>USING the ...</b>
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 or the LCDestInfo 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>.ddhmmss*.
- ▶ 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 <i>&lt;data_manager_name&gt;.l</i> .
The command line	Run log called <i>xlld.l</i> . This file resides in the <b>xlld</b> 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 lclld Program](#) and the [The xlld Program](#) topics in this chapter for information about how **lclld** 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 -o<ODBC_trans_audit_file>
-i<ODBC_init_file> -s<ODBC_data_source> -a<ODBC_alias_file_name>
```

Where...	Is the...
-f<file_name>	Path of the translation audit file without the .v extension. If you specify this option, <b>xlld</b> updates the Life Cycle table lc221 first.
-u[userID]	User ID for the Life Cycle database account
-p[password]	User's password for the Life Cycle database account
-e	Argument that causes <b>xlld</b> to post only the records in error to the log file. The default is to post the entire event file.
-o<ODBC_trans_audit_file>	Name of the ODBC translation audit file without the ".v" extension
-i<ODBC_init_file>	Name of the ODBC initialization file
-s<ODBC_data_source>	Name of the data source

(Continued on next page)

(Contd) Where...	Is the...
-a<ODBC_alias_file_name	<p>Name of the alias file that defines the meaning of entries in the ODBC initialization file. If you do not specify a file name, <b>xlld</b> uses the data source file name. If the ODBC alias file does not exist, <b>xlld</b> explains the entries in the ODBC initialization file this way:</p> <ul style="list-style-type: none"> <li>▶ DestDir contains the name of the database</li> <li>▶ DestDesc contains the description of the database</li> <li>▶ DestHost contains the name of the database server</li> <li>▶ DestUser contains the user ID for the database account.</li> </ul>

**Example**

```
xlld -f../xlld/xi01
```

**Reference**

See the *Gentran:Server for UNIX and Workstation Technical Reference Guide* for more information about the **xlld** command.

# How Life Cycle Handles Inbound Database Audit Records

---

## Introduction

You can configure the translator program **Iftran** to generate translation audit records for inbound database translations. The translator generates one record per document per database table.

## Reference

See the [Command Reference](#) chapter in the *Gentran:Server for UNIX and Workstation Technical Reference Guide* for information about the options for the **Iftran** command.

---

## Record contents

A translation audit record for a database contains the:

- Trading Partnership code
  - Document reference number
  - Database name (directory, if available)
  - Table name (and file name, if available)
  - Character count
  - Record count.
-

# Life Cycle Reports

## Overview

---

**Introduction** 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 your support representative 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.

(Continued on next page)

<b>(Contd) Report</b>	<b>Function</b>
tplist	<p>Lists the interchange and group IDs of you and your trading partners along with the GS version and set IDs. Used to:</p> <ul style="list-style-type: none"> <li>▶ Identify Trading Partnership records that you no longer need</li> <li>▶ Understand the contents of a data file when you can determine only part of the Trading Partnership identifying information.</li> </ul>
tracker	<p>Summarizes translation traffic for a specified date by Trading Partnership code or by data manager name. Used to view the:</p> <ul style="list-style-type: none"> <li>▶ Number of good and bad sets for inbound and outbound translation</li> <li>▶ Total number of segments and total number of characters processed for the Trading Partnership code or data manager name</li> <li>▶ File names, if any, that are in error.</li> </ul>

**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.

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	16510001	o	C:\MENTOR20\TEM /		08/20/1997

15 ROWS SELECTED.

(Continued on next page)

**Translation  
Activity Report  
field  
descriptions**

This table describes the fields of the Translation Activity Report.

<b>Field</b>	<b>Description</b>
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. <ul style="list-style-type: none"> <li>▶ I = Inbound EDI</li> <li>▶ O = Outbound EDI</li> <li>▶ X = Application</li> </ul>
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.

CODE	MYISID	MYGSID	TPISID	TPGSID	VERSION	SET	DOCUMENT	DATE	IS CTRL#	GS CTRL#	ST CTRL#	FAREQ	FADT
3ND850	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

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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
Version	The document version number.
Set	The transaction set number that identifies the type of business document.
Document	The document reference number.
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. <ul style="list-style-type: none"><li>▶ Y = FA has been requested</li><li>▶ N = FA has not been requested</li></ul>
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.
XIDate	The date that the document was translated
XITime	The time that the document was translated.

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
FaReq	The functional acknowledgment request flag.
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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
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 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.

(Continued on next page)

<b>(Contd) Field</b>	<b>Description</b>
Total Outbound	The total number of outbound transaction sets that the translator processed.
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	Informix	<a href="#">How to Run Informix Life Cycle Reports.</a>
FA Exception	Oracle	<a href="#">How to Run Oracle Life Cycle Reports</a>
FA Due		
TP List		
Translation Traffic	--	<a href="#">How to Run the Translation Traffic Report</a>

(Continued on next page)

---

**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 *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	<p>At the UNIX command line, open the directory that contains the files that will generate the Life Cycle report files.</p> <p><b>CAUTION</b>  <b>The name of the directory that holds the .ace files is <i>\$EDI_ROOT/src/infx</i>.</b></p>
2	<p>Type <b>isql</b> at the prompt and press ENTER.</p> <p><b>System Response</b>  The system displays the Informix interactive environment screen.</p>
3	<p>Select <b>Report</b> from the menu and then press ENTER.</p> <p><b>System Response</b>  The system displays the Report menu.</p> <p style="text-align: right;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>
4	Select <b>Run</b> from the Report menu and then press ENTER.  <b>System Response</b> The system displays a list of reports.
5	Select the report that you want to run and then press ENTER.  <b>System Response</b> 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 <b>sqlplus</b> and then press ENTER.  <b>System Response</b> The system displays the Oracle sign-in screen.
3	Enter your user name and password.  <b>System Response</b> 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, <i>.sql</i> .  <b>Example</b> To run the Translation Activity Report, enter the following:  <i>@&lt;full_path&gt;/activity.sql</i>  Where <i>&lt;full_path&gt;</i> is the full path to the Life Cycle report files.  <b>System Response</b> 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.

---



---

# Monitoring Processes

<b>Contents</b>	<b>Overview</b>	
	▶ Introduction .....	3
	▶ The Screen Viewer .....	5
	<b>Monitoring Data Manager Processes</b>	
	▶ Overview .....	7
	<b>Data Manager Control Screen</b>	
	▶ Data Manager Log Files .....	11
	▶ How to Check a Data Manager's Status .....	13
	▶ How to View a Data Manager's Log File .....	14
	<b>Starting and Stopping Data Managers</b>	
	▶ Overview .....	16
	▶ How to Use the Data Manager Control Screen .....	17
	▶ How to Stop Data Managers .....	20
	▶ How to Start Data Managers .....	23
	<b>Maintaining Data Manager Log Files</b>	
	▶ Overview .....	26
	▶ How to Purge Data Manager Log Entries .....	27
	▶ How to Delete a Data Manager's Log File .....	29
	<b>Monitoring Scripts</b>	
	▶ Overview .....	30
	▶ Script Logs and Journals .....	31
	▶ How to Check the Status of a Script .....	33
	▶ How to View a Script Log or Journal .....	35
	▶ How to Display a Script's Processing Time Statistics .....	36
	<b>Maintaining Script Logs and Journals</b>	
	▶ Overview .....	38

- ▶ How to Purge Entries from Journals ..... 39
  - ▶ How to Delete a Log or Journal ..... 41
-

# 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: <ul style="list-style-type: none"> <li>▶ Routing data</li> <li>▶ Invoking scripts</li> <li>▶ Archiving data</li> <li>▶ Handling data errors.</li> </ul>
log file	A file that contains a record of process activity and messages produced by that activity. <p><b>Example</b> A data manager log contains a record of the data manager's status at a given date and time and any message produced.</p>
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.

(Continued on next page)

<b>(Contd) Term</b>	<b>Description</b>
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.1
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.

(Continued on next page)

<b>Keys</b>	<b>Action</b>
H or h or Left Arrow	Scroll viewer window left one character.
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.

```

Data Manager Control
Name A      Status  T Description
fmgr A ***** F Foreground Manager (IPC Control)
alnm n ***** l Async Line Manager
appm y ***** m Application Data Manager
appt n ***** x Application Translator Data Manager
arch n ***** a -Darch -Aarch -d0
base n ***** u Base Manager Model
dl00 n ***** x Flow: stdappl Delivery Agent
dl01 n ***** x Flow: stdapp2 Delivery Agent
dl02 n ***** x Flow: edi_xl2 Delivery Agent
dnld y ***** d UDF Data Manager
edii y ***** i Inbound Data Manager
edio y ***** i Outbound Data Manager
file n ***** f File Data Manager
hcmd n ***** h Host Command Card Data Manager
inbd n ***** i Inbound Data Manager
lnmn n ***** l Line Manager
pr00 n ***** x Flow: stdappl Processing Agent
F2:Arch F4:EditMail F5:Config F6:Stat F7:Log F9:Quit
    
```

**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. <ul style="list-style-type: none"> <li>▶ Y = Yes</li> <li>▶ N = No</li> </ul>

(Continued on next page)

(Contd) Field	Function	
Status	Indicates the running status of the data manager.	
	<b>Tip</b> Press F6 to update the status.	
	<b>IF the column displays...</b>	<b>THEN the data manager is...</b>
	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.	
T	Defines the data manager's personality type.	
	<b>Type Code</b>	<b>Description</b>
	F	The foreground manager
	d	Download
	i	Inbound
	a	Archive
	f	File
	x	Translator
	l	Line manager (Mailbox System only)
	h	Host command card
	m	Application
Description	Describes the data manager. Enter up to 50 characters. <b>CAUTION</b> <b>For an archive data manager, this field contains the archive data manager's processing parameters.</b> <b>Reference</b> For information about using this field to set the processing parameters for an archive data manager, see the <a href="#">Archiving Your Data</a> chapter in this guide.	

(Continued on next page)

**Function keys of  
the Data  
Manager Control  
screen**

This table describes the function keys of the **Data Manager Control** screen.

Key	Function
F2	Displays the Query screen for the selected archive handler.
F3	Stops a running data manager.
F4	Opens the default editor so that you can edit the data manager's UNIX mail script.
F5	If pressed while fmgr is selected, displays the Data Manager Configuration screen, which enables you to copy, modify, and delete data managers.  If pressed while a data manager is selected, enables you to open, edit, and delete configuration records for the selected data manager.
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

**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.**

**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.1
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

```

(Continued on next page)

**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. <ul style="list-style-type: none"> <li>▶ 0 implies success</li> <li>▶ Any other code implies an error.</li> </ul>
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	<p>Select <b>DataMgr</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen. This screen lists all the data managers added to Gentran:Server.</p> <pre> Data Manager Control Name A Status T Description ----- fmgr A ***** F Foreground Manager (IPC Control) alna n ***** l Async Line Manager appm y ***** m Application Data Manager appt n ***** x Application Translator Data Manager arch n ***** a -Darch -Aarch -d0 base n ***** u Base Manager Model d100 n ***** x Flow: stdappl Delivery Agent d101 n ***** x Flow: stdapp2 Delivery Agent d102 n ***** x Flow: edi_xl2 Delivery Agent dnld y ***** d UDF Data Manager edi y ***** i Inbound Data Manager edio y ***** i Outbound Data Manager file n ***** f File Data Manager hcmd n ***** h Host Command Card Data Manager inbd n ***** i Inbound Data Manager lnmn n ***** l Line Manager pr00 n ***** x Flow: stdappl Processing Agent ----- F2:Arch F4:EditMail F5:Config F6:Stat F7:Log F9:Quit </pre>
2	Check the status column of the data manager to determine whether or not it is active.
3	<p>Press F6 to update the Status column.</p> <p><b>System Response</b> Gentran:Server replaces the displayed status with the current status.</p>
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	<p>Select <b>DataMgr</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen. This screen lists all the data managers added to Gentran:Server.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> Data Manager Control Name A      Status  T Description ----- fmgr A      1239 F Foreground Manager (IPC Control) alnm n      ***** l Async Line Manager ap00 n      ***** m Flow: Test Q! flow Source Agent ap01 n      ***** m Flow: 'nother fifteen Translate Agent appm n      ***** m Application Data Manager appt n      ***** x Application Translator Data Manager arch n      ***** a -Darch -Aarch -d0 base n      ***** u Base Manager Model cfin n      ***** i Flow: chris_flow Translate Agent dnld n      ***** d UDF Data Manager edii n      ***** i Inbound Data Manager edio n      ***** i Outbound Data Manager file n      ***** f File Data Manager hcmd n      ***** h Host Command Card Data Manager in00 n      ***** i Flow: flow Source Agent in01 n      ***** i Flow: fifteen digits. Translate Agent in02 n      ***** i Flow: test0505 Translate Agent F2:Arch F3:Stop F4:EditMail F5:Config F6:Stat F7:Log F8:Start F9:Quit                     </pre> </div>
2	<p>Select the data manager and then press F7 to view the log.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>



<b>(Contd) Step</b>	<b>Action</b>
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, 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/<User Files>/fmgr.l log` file contains information about the termination.

---

(Continued on next page)

**Procedure**    Use this procedure to start or stop a data manager.

Step	Action
1	<p>Select <b>DataMgr</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Data Manager Control screen.</p> <pre> Data Manager Control Name A      Status  T Description ----- Emgr A      1239 F Foreground Manager (IPC Control) alnm n ***** l Async Line Manager ap00 n ***** m Flow: Test Q! flow Source Agent ap01 n ***** m Flow: 'nother fifteen Translate Agent appa n ***** m Application Data Manager appt n ***** x Application Translator Data Manager arch n ***** a -Darch -Aarch -d0 base n ***** u Base Manager Model cfin n ***** i Flow: chris_flow Translate Agent dnld n ***** d UDF Data Manager edii n ***** i Inbound Data Manager edio n ***** i Outbound Data Manager file n ***** f File Data Manager hcmd n ***** h Host Command Card Data Manager in00 n ***** i Flow: flow Source Agent in01 n ***** i Flow: fifteen digits. Translate Agent in02 n ***** i Flow: test0505 Translate Agent F2:Arch F3:Stop F4:EditMail F5:Config F6:Stat F7:Log F8:Start F9:Quit                     </pre>
2	<p>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.</p> <p><b>Reference</b> See the <a href="#">How to Check a Data Manager's Status</a> topic in this section for more information.</p>
3	<p>Select the data manager you want to start or stop.</p>
4	<p>Press the appropriate function key:</p> <ul style="list-style-type: none"> <li>▶ F3 to stop the data manager</li> <li>▶ F8 to start the data manager.</li> </ul>

(Continued on next page)

**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, 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 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.

---

(Continued on next page)

## Running stopserver or stopserver.sh from the command line

Use this procedure to run **stopserver** or **stopserver.sh** from the command line.

Step	Action	
1	Go to the command line.	
2	Type the <b>stopserver</b> command or the <b>stopserver.sh</b> script command.	
	<b>IF you want to stop...</b>	<b>THEN type...</b>
	A single data manager	<pre>stopserver &lt;dmname&gt;</pre> OR <pre>stopserver.sh -d &lt;dmname&gt;</pre> where <i>&lt;dmname&gt;</i> is the name of the data manager. <b>Examples</b> <pre>stopserver edii</pre> <pre>stopserver.sh -d edii</pre>
	The foreground manager and all other data managers	<pre>stopserver</pre> OR <pre>stopserver.sh</pre>
	<b>System Response</b>	
	<b>IF you...</b>	<b>THEN stopserver or stopserver.sh...</b>
	Specified a data manager	Has the foreground manager wait until the data manager finishes processing and then stops the data manager.
Did not specify a data manager	Displays a list of data managers that are running and prompts you to choose a timeout value. Continue with Step 3.	
3	Type the number of the timeout value you want <b>stopserver</b> or <b>stopserver.sh</b> to use or press ENTER to use the default timeout value. <b>System Response</b> Gentran:Server sends a termination message to the foreground manager and the data managers. When the data managers stop, <b>stopserver</b> or <b>stopserver.sh</b> displays this message: The agents of foreground manager (xxxxx) are stopped.	

(Continued on next page)

**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	<pre>stopserver &lt;dmname&gt;</pre> <p>OR</p> <pre>stopserver.sh -d &lt;dmname&gt;</pre> <p>where <i>&lt;dmname&gt;</i> is the name of the data manager.</p> <p><b>Examples</b></p> <pre>stopserver edii</pre> <pre>stopserver.sh -d edii</pre>
The foreground manager and all data managers	<pre>stopserver</pre> <p>OR</p> <pre>stopserver.sh</pre>

**Note**

For more information about the **stopserver** command or the **stopserver.sh** script, see the [Command Reference](#) chapter of the *Gentran:Server for UNIX and Workstation Technical Reference Guide*.



# How to Start Data Managers

---

## Introduction

Your 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 *Gentran:Server for UNIX and Workstation 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 Gentran:Server **recover.scr** recover script. The recover script restarts (from the beginning) specified 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

---

(Continued on next page)

**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	<pre>startserver &lt;dmname&gt;</pre> <p>OR</p> <pre>startserver.sh -d &lt;dmname&gt;</pre> <p>where &lt;dmname&gt; is the name of the data manager.</p> <p><b>Examples</b>  <pre>startserver edii</pre> <pre>startserver.sh -d edii</pre></p>
The foreground manager and all data managers configured to autostart with the foreground manager	<pre>startserver</pre> <p>OR</p> <pre>startserver.sh</pre>

**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	<pre>startserver &lt;dmname&gt;</pre> <p>OR</p> <pre>startserver.sh -d &lt;dmname&gt;</pre> <p>where &lt;dmname&gt; is the name of the data manager.</p> <p><b>Examples</b>  <pre>startserver edii</pre> <pre>startserver.sh -d edii</pre></p>
The foreground manager and all data managers configured to autostart with the foreground manager	<pre>startserver</pre> <p>OR</p> <pre>startserver.sh</pre>

(Continued on next page)

**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. 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 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.

---

(Continued on next page)

**Purging log entries**

Use this procedure to purge log entries.

Step	Action										
1	Go to the UNIX command line and make the EDI_ROOT directory the current directory.										
2	<p>From the following table, choose the type of purge you want to perform and enter the command at the command line.</p> <p><b>Rules</b> Enter the name of the log in the &lt;logfile&gt; portion of the command. Enter dates in the format MMDDYY or MMDDCCYY.</p> <table border="1" data-bbox="618 808 1422 1312"> <thead> <tr> <th data-bbox="618 808 1024 863">To purge...</th> <th data-bbox="1024 808 1422 863">Use this command...</th> </tr> </thead> <tbody> <tr> <td data-bbox="618 863 1024 951">All entries older than 90 days in all logs</td> <td data-bbox="1024 863 1422 951">ksh cleanlog.sh</td> </tr> <tr> <td data-bbox="618 951 1024 1077">All entries older than the specified number of days in all logs</td> <td data-bbox="1024 951 1422 1077">ksh cleanlog.sh &lt;days&gt;</td> </tr> <tr> <td data-bbox="618 1077 1024 1203">All entries older than the specified number of days in a single log</td> <td data-bbox="1024 1077 1422 1203">cleanlog &lt;logfile&gt; -a&lt;days&gt;</td> </tr> <tr> <td data-bbox="618 1203 1024 1312">All entries for a specific date range in a single log</td> <td data-bbox="1024 1203 1422 1312">cleanlog &lt;logfile&gt; -B&lt;delete from date&gt; -E&lt;delete to date&gt;</td> </tr> </tbody> </table>	To purge...	Use this command...	All entries older than 90 days in all logs	ksh cleanlog.sh	All entries older than the specified number of days in all logs	ksh cleanlog.sh <days>	All entries older than the specified number of days in a single log	cleanlog <logfile> -a<days>	All entries for a specific date range in a single log	cleanlog <logfile> -B<delete from date> -E<delete to date>
To purge...	Use this command...										
All entries older than 90 days in all logs	ksh cleanlog.sh										
All entries older than the specified number of days in all logs	ksh cleanlog.sh <days>										
All entries older than the specified number of days in a single log	cleanlog <logfile> -a<days>										
All entries for a specific date range in a single log	cleanlog <logfile> -B<delete from date> -E<delete to date>										
	<p><b>Reference</b> For more information about the cleanlog command, see the <a href="#">Command Reference</a> chapter of the <i>Gentran: Server for UNIX and Workstation Technical Reference Guide</i>.</p>										

# 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 <b>DataMgr</b> from the Gentran:Server host main menu. <b>System Response</b> Gentran:Server displays the Data Manager Control screen. This screen lists all the data managers added to Gentran:Server.
2	Select the data manager and then press F7 to view its log file.
3	Press F3 to delete the log file. <b>System Response</b> 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** Gentran:Server scripts record their process activity in script logs and script journals.

---

**Script log** When a 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. 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.num s script/tmp.num s :
sint:4502:10121998:151144:0:STEP CNT:0:STEP in:qnd-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
```

(Continued on next page)

**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.  <b>Example</b> The name of the script manager process is <b>smgr</b> .
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. <ul style="list-style-type: none"> <li>▶ 0 implies success</li> <li>▶ Any other code implies an error</li> </ul>
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, 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 Gentran:Server script is running, you can check its status on the Script Maintenance screen. 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	<p>Select <b>Script</b> from the host main menu to access the Script Maintenance screen.</p> <pre data-bbox="641 934 1398 1285"> 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           </pre> <p><b>System Response</b> Gentran:Server:</p> <ul style="list-style-type: none"> <li>▶ Scans the <code>./temp</code> directory to determine which scripts have log files</li> <li>▶ Displays a list of the scripts with the status and description of each script on the Script Maintenance screen.</li> </ul> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd) Step</b>	<b>Action</b>	
2	Locate the name of the script and check the status in the Status column.	
	<b>IF the status is...</b>	<b>THEN the script is...</b>
	Active	Running
	Inactv	Not running
	<p><b>CAUTION</b>  <b>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.</b></p> <p><b>WARNING</b>  <b>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.</b></p> <p><b>Reference</b>            See the <a href="#">How to Delete a Log or Journal</a>.</p>	

**Updating the status column**

You can recheck the status by pressing F6. When you press this key, Gentran:Server scans the directory again and updates the Status column.

## How to View a Script Log or Journal

**Introduction** The 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	<p>Select <b>Script</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Script Maintenance screen.</p> <pre> 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           </pre>
2	Select the name of the script that you want to view.
3	<p>Press F7 to display the script's log in the Screen Viewer.</p> <p><b>System Response</b> If the script is running, Gentran:Server displays the script log (<i>script_name.l</i>). If the script is not running, Gentran:Server displays the script journal (<i>script_name.old</i>).</p>
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 Gentran:Server
- Check the efficiency of a script.

---

**Example** This is an example of processing time statistics for a Gentran:Server script.

The time is shown in minutes (m) and seconds (s).

```
real    0m0.31s
user    0m0.06s
sys     0m0.07s
```

---

(Continued on next page)

**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 Gentran:Server script.,

Step	Action
1	Go to the UNIX command line.
2	<p>Enter the <b>time</b> command to run the script manager, <b>smgr</b>, and the script. This is the command format:</p> <pre>time smgr -s&lt;server_script&gt;</pre> <p><b>Example</b> This example times the script named xltr.scr.</p> <pre>time smgr -sxltr</pre> <p><b>Comment</b> Omit the .scr extension in the script's name. Do not leave a space between the -s argument and the script's name.</p>

**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** 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	<pre>ksh cleanlog.sh</pre> <p><b>Note</b> cleanlog.sh is a script that calls the cleanlog program.</p> <p style="text-align: right; color: red;">(Continued on next page)</p>

<b>(Contd)</b> <b>IF you want to purge entries...</b>	<b>THEN use this command...</b>
Older than a specific number of days from all journals	<pre>ksh cleanlog.sh &lt;days&gt;</pre> <p><b>Note</b> cleanlog.sh is a script that calls the cleanlog program.</p> <p><b>Example</b> To purge from all entries from a journal older than 60 days:</p> <pre>ksh cleanlog.sh 60</pre>
Older than a specified number of days in a single journal	<pre>cleanlog &lt;logname&gt; -a&lt;number of days&gt;</pre> <p><b>Example</b> To purge entries older than 90 days from the ftp_to.scr script:</p> <pre>cleanlog ftp_to.scr -a90</pre>
In a specific date range in a single journal	<pre>cleanlog &lt;logname&gt; -B&lt;delete _begin_ date&gt; -E&lt;delete_end_ date&gt;</pre> <p><b>Example</b> To purge from the ftp_to.scr script entries dated May 1, 1997 through May 15, 1997:</p> <pre>cleanlog ftp_to.scr -B04011997 -E04161997</pre>

**Reference**

For more information, see the [Command Reference](#) chapter in the *Gentran: Server for UNIX and Workstation 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.**

---

(Continued on next page)

**Deleting a journal or log**

Use this procedure to delete a script's journal or log.

Step	Action
1	<p>Select <b>Script</b> from the Gentran:Server host main menu.</p> <p><b>System Response</b> Gentran:Server displays the Script Maintenance screen.</p> <pre data-bbox="634 625 1406 978"> 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&gt;Edit F6:Stat F7:Log F8:Exec F9:Quit                     </pre>
2	<p>Select the script name.</p>
3	<p>Press F7 to display the script log or journal.</p> <p><b>System Response</b> If the script is running, Gentran:Server displays the script log (<i>script_name.l</i>). If the script is not running, Gentran:Server displays the script journal (<i>script_name.old</i>).</p> <p><b>SUGGESTION</b> <b>Look at the name of the log to determine whether the script log or the script journal is displayed.</b></p>
4	<p>Press F3 to delete the log or journal.</p> <p><b>System Response</b> Gentran:Server deletes the log or journal and displays the Script Maintenance screen again.</p>

---

# Glossary

---

<b>action line</b>	The line in a script that contains the actual commands you want carried out.
<b>active archive period</b>	The number of days prior to today for which the system maintains active archive entries in the archive index file.
<b>agent</b>	A data manager.
<b>application name map</b>	A record that identifies the records and fields that the data manager extracts from the application transaction to build the document reference number.
<b>arch</b>	The default name of the archive handler.
<b>archive</b>	The process of capturing and storing a copy of a document after a data manager processes it.
<b>archive data file</b>	The file to which the archive handler routes data to be archived. Each data manager has its own archive data file.
<b>archive index file</b>	An index file that contains a detailed record of the archived files for a specific data manager.
<b>archive directory</b>	The directory to which a data manager routes an archive copy of a document it processes. This directory is the archive handler's work directory.

---

<b>archive handler</b>	The special data manager designed to archive data.
<b>audit file</b>	The file that a data manager produces to pass archiving instructions to the archive handler.
<b>auditing facility</b>	The 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.
<b>base initialization file</b>	The default initialization file that contains all the processing parameters for all data manager personality types except the archive data manager.
<b>category</b>	A class or grouping of Trading Partnership records.
<b>child process</b>	A process that is started by and is part of another process. The other process is called the parent process.
<b>cl_arch</b>	The program that purges your archives.
<b>clean_trn</b>	The clean transaction register utility, which is used to delete transaction register entries older than a specified date and time.
<b>cleanlog</b>	The command line tool that enables you to purge log file entries.
<b>comment line</b>	A phrase or sentence in a script that explains the purpose or effect of the line of instructions that follows the comment line.

---

---

<b>configuration record</b>	<p>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:</p> <ul style="list-style-type: none"><li>• Specifies the Trading Partnership code or file name that the data manager is to use to identify data</li><li>• Tells the data manager what to do with the data it has identified.</li></ul>
<b>cron</b>	<p>The UNIX system daemon that starts programs identified in the system's <b>crontab</b> at scheduled times.</p>
<b>crontab</b>	<p>A UNIX system file that contains the files listing all the programs to be run by the <b>cron</b> daemon. Gentran:Server submits entries in the permanent schedule to <b>crontab</b>.</p> <p>database</p> <p>A collection of stored data often shared by different applications.</p>
<b>data manager</b>	<p>A program that periodically scans a directory or queue for data files and then processes the files it finds. Processing can include:</p> <ul style="list-style-type: none"><li>• Routing data</li><li>• Invoking scripts</li><li>• Archiving data</li><li>• Handling data errors.</li></ul>
<b>data manager log</b>	<p>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.</p>
<b>date-time stamp</b>	<p>The label that Gentran:Server attaches to a document to identify the date and time the document was received.</p>
<b>default transaction register</b>	<p>The transaction register that a data manager uses unless you specify a different directory for the transaction register in the data manager's initialization file.</p>

---

---

<b>default value</b>	The value that Gentran:Server uses if you do not specify a different one.
<b>delimiter</b>	Special characters that designate the type of information on a line in the script.
<b>deltrn</b>	The transaction register delete utility, which is used to delete transaction register entries.
<b>document reference number</b>	The unique number that Gentran:Server assigns to each data set or document to track the movement of the data set or document through the system.
<b>document specifier table</b>	A collection of set ID or application name maps that specify the places in a trading partner's document that certain Gentran:Server processes use to construct the document reference number.
<b>document specifier utility</b>	The 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.
<b>EDI_ARCHIST</b>	The environment variable that sets the default active archive period.
<b>error class</b>	An error category set for an error handler and entered into a configuration record. An error class enables GENTRAN:Server to handle the same type of error differently for different Trading Partnerships or file names.
<b>error handler</b>	A device in an initialization file used to specify how you want a particular error handled.
<b>FIFO</b>	A first-in-first-out file that can be read only once before the data is removed

---



---

<b>Foreground Manager</b>	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.
<b>FTP daemon</b>	The background program that controls file transfer protocol. File transfer protocol moves or copies files between computers.
<b>functional acknowledgment (FA)</b>	The standard transaction set used to acknowledge receipt of a transmission.
<b>group</b>	A set of related parameters in an initialization file.
<b>initialization file</b>	The configurable file that sets the data manager's personality and processing parameters.
<b>intelligent agent</b>	An event-driven computer program that can operate without interaction from a person at a computer terminal.
<b>IPC Trigger</b>	The GENTRAN:Server feature that enables you to configure data managers to process files in real-time mode.
<b>key fields</b>	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 distinguish it from other entries.
<b>label line</b>	The line in a script that contains the name that you assign to the action that you want performed.

---

---

**lcl** The 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 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 **lcl** and **xlcl**. 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.

---

**lock** A device in an initialization file used to ensure that two data managers or other processes do not process a file at the same time.

---

**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.

---

<b>longterm</b>	The script that controls long-term archiving.
<b>mail_proc file</b>	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.
<b>mailbag ID</b>	The 6-character, base-32 code that Gentran:Server generates to identify a session in which files were received and data files passed in the session.
<b>map picture</b>	A pictorial representation of the map for the document reference number. Each field in the map is represented with a unique symbol.
<b>meld</b>	The process of merging a pattern with Trading Partnership records to produce configuration records.
<b>meld log</b>	The temporary log file produced when a pattern is merged with Trading Partnership records. Contains messages about the meld operation. The end of the log shows the number of new records created and the number of records rewritten.
<b>mksrvdb</b>	The program or script that creates the database Life Cycle table.
<b>pattern</b>	One or a series of generic configuration records that describe the flow of data in a process.
<b>pattern configuration record</b>	A generic configuration record used in the meld process to generate actual configuration records from selected Trading Partnership codes or file names.

---

---

<b>Permanent Schedule</b>	The Gentran:Server feature that enables you to run scripts on a specified schedule.
<b>personality</b>	The data manager type that determines how the data manager processes data.
<b>PID</b>	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.
<b>process flow</b>	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.
<b>Process Control Manager (PCM) wizard</b>	The Gentran:Server wizard that guides you through the process of creating a two-agent flow.
<b>real-time processing</b>	A system configuration that enables your system to move critical documents through the processing cycle as quickly as possible.
<b>record file layout</b>	The user-defined file layout of an application file.
<b>recover script</b>	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.
<b>rtv_arc</b>	The program that retrieves archived files.
<b>rtv_arc log</b>	The log produced when you retrieve archived files. Contains a description of the entries retrieved, their names, and where they were sent.

---

---

<b>script</b>	A set of commands that controls processes or performs some action.
<b>script directory</b>	The directory that contains all Gentran:Server scripts.
<b>script editor</b>	The default editor that Gentran:Server calls when you create or edit a script.
<b>script journal</b>	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.
<b>script library</b>	A file that lists and describes all the Gentran:Server scripts that have been added to the file.
<b>script log</b>	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.
<b>Script Manager</b>	The Gentran:Server program that directs the script interpreter to execute the commands in a script.
<b>sequence number</b>	Value that <b>lcl</b> 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 <b>lcl</b> program increments the sequence number by one to distinguish the new entry from the existing entry.
<b>set ID map</b>	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.
<b>source agent</b>	The data manager (inbd or appm personality) that begins a PCM inbound process flow.

---

---

<b>status</b>	The running state (active or inactive) of a script or data manager.
<b>token</b>	A generic place holder in a pattern file. During the meld process, GENTRAN:Server replaces the tokens in the pattern with specific values from the Trading Partnership records and other GENTRAN:Server sources. This creates a configuration record for each Trading Partnership you meld with the pattern.
<b>tracker</b>	The Gentrans:Server command line program that enables you to run a statistical report on the translation traffic from Life Cycle data.
<b>Trading Partner record</b>	One of the five records maintained in trading partner files: Trading Partnership record, Interchange Organization record, Group Organization record, and Contact record, and TRADACOMS record.
<b>Trading Partnership</b>	An arrangement with a specific trading partner to exchange information in a specific document type, described by a map file.
<b>Trading Partnership code</b>	A user-defined code that uniquely identifies a Trading Partnership record.
<b>Trading Partnership record</b>	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.
<b>transaction register</b>	An indexed file used to keep track of documents that GENTRAN:Server handles.
<b>translation agent</b>	The translation data manager (xltr personality). For PCM 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 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 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 Gentran:Server program that updates the Life Cycle table with translation audit file and functional acknowledgment information.
-





# Index

## Symbols

\$EDITOR environment variable 9-30, 9-38  
 ./mail\_proc 9-54  
 ./script directory 9-31  
     effect of deleting script from 9-52  
 \_gen\_xltr.scr 3-39, 3-59

## A

action line (in a Gentran:Server script) 9-10  
 Add Queue Entry screen  
     fields and functions 7-24  
     function keys 7-25  
     illustration 7-24  
 ADD\_CONFIG 5-45  
 ADD\_MBAG 5-30  
 ADD\_MBAG\_TO\_DOC\_REF\_NUM 5-47  
 ADD\_TP 5-46  
 administrative login ID 1-6  
 ALLOW\_DUPLICATES 5-53  
 APP Add screen 11-33  
     fields and functions 11-33  
     function keys 11-36  
     illustration 11-33  
 APP Mapping screen  
     fields and functions 11-66  
     function keys 11-68  
     illustration 11-66  
     purpose 11-66  
 APP\_ERROR 5-17  
 application data manager 4-6  
 application description  
     length of file name 1-10  
 Application name map 11-25  
 application name map  
     adding 11-43  
     adding to a document specifier table 11-43  
     changing 11-76  
     deleting from a document specifier table 11-79  
 application-to-application process flow 3-38  
 application-to-standard process flow 3-38  
 arch (default name for archive handler) 8-6  
 Archive Data field  
     role in archiving process 8-9  
 archive directory 2-36  
     where name is set 8-4

archive file  
     definition 8-5  
     name 8-5  
 archive flag 2-34  
 archive handler  
     adding 8-13  
     configuration 8-6  
     configuring 8-13  
     definition 8-6  
     functions 8-6  
     how data flows through 8-8  
     ways to invoke 8-7  
 archive handling parameters (in initialization file) 5-52  
 archive index  
     contents 8-25  
     file 8-5  
     restoring from tape 8-25  
 archive index file  
     contents 8-5  
     name 8-5  
 Archive List screen  
     fields and functions 8-29  
     function keys 8-30  
     illustration 8-29  
 archive period  
     definition 8-16  
     how GENTRAN:Server determines 8-16  
 Archive Query screen  
     fields and functions 8-27  
     function keys 8-28  
     illustration 8-27  
     using to search for archived files 8-31  
 archive retrieval  
     process 8-26  
     program rtv\_arc 8-26  
 archive retrieval log  
     contents 8-34  
     displaying 8-34, 8-35  
     example 8-34  
     name 8-34  
 ARCHIVE\_DIRECTORY 5-52  
 ARCHIVE\_ERROR 5-52  
 ARCHIVE\_PERIOD 5-52  
     using to set data manager's archive period 8-16

- archived data file
    - contents 8-5
    - name 8-2, 8-5
  - archived files
    - managing overview 8-15
    - manually purging 8-22
    - retrieving overview 8-25
    - selecting and retrieving 8-31
  - archiving
    - configuring a data manager to perform 8-4
    - configuring your system 8-10
    - definition 8-4
    - introduction to 8-4
    - process 8-9
    - processed data 8-1
    - task overview 8-10
  - audit file 13-7, 14-7
    - contents 8-4
    - definition 8-4
    - name 8-5
    - role in archiving process 8-9
- B**
- background process 1-8
  - build (reserved word) 9-25
- C**
- categories
    - using in patterns 6-24
  - category
    - definition 6-24
  - CHK\_PREFIX 5-47
  - cl\_arch 8-22
    - command line examples 8-24
    - command line format 8-22
    - functions 8-16
    - invocation screen 8-22
    - purging archived files with 8-22
    - relationship to longterm script 8-15, 8-17
    - running from the command line 8-22
  - clean\_trn utility 12-16
    - running from the command line 12-16
  - CMD\_STR 5-42
  - CMD\_STR parameter 4-29
  - CMD\_STR\_ENV 5-42
  - CMD\_STR\_ENV parameter 4-29
  - command cards 4-28
    - format of 4-28
  - comment delimiter
    - required position in a script line 9-33
  - compare (reserved word) 9-25
  - concurrent processing
    - controlling with file locking 5-73
  - conf.d directory 5-7
  - configuration record
    - configuring for data archiving 8-12
    - copying 6-17
    - created by PCM wizard 3-5
    - creating manually 6-8, 6-13
    - definition 6-5
    - deleting 6-69
    - editing 6-67
    - opening 6-15
    - verifying creation of 6-66
  - configuration records
    - deleting a range of 6-70
    - for host command card data manager 4-29
    - introduction to 6-5
    - number needed 6-5
    - using meld function to generate 6-57
    - ways to create 6-6
  - Configuration screen
    - illustration 6-8
  - copyright screen
    - displaying 1-19
  - cpio 8-17
  - cpio command
    - using to move old archived files 8-15
  - cpre compiler
    - requirements for Sybase database table 13-37
  - cron
    - enabling cron to find Gentran:Server executable files 10-15
  - crontab
    - relationship to Permanent Schedule 10-15
- D**
- data flow administration 1-4
    - exiting 1-14, 1-19
    - starting 1-14
  - data flow administrator main menu 1-12
  - DATA group
    - avoiding blank lines in 9-33
    - description 9-17
    - example 9-18
    - general format 9-17
    - relationship to DPROCS and PROCS groups 9-17
    - uses 9-17
  - data manager
    - ability to recall stopping point 4-8
    - configuration overview 4-10

- configuring to archive data 8-11
- copying 4-19
- definition 2-7
- deleting 4-25
- description and purpose 4-5
- editing process fields 4-22
- effect of processing halts and restarts 4-8
- length of file name 1-10
- name 2-8
- personalities 4-5
- processing actions 4-7
- processing illustration 4-7
- role in a process flow 2-7
- running script for an error 10-7
- selecting for a process flow 2-18
- starting 15-17
- starting with startserver or startserver.sh 15-23
- stopping 15-17
- stopping with stopserver or stopserver.sh 15-20
- terminating without a Foreground Manager request 4-9
- terminating without a request from fmgr 15-17
- types 2-9
- Data Manager Configuration screen
  - function keys 4-17
  - illustration 4-17, 6-8
- Data Manager Control screen 4-12
- data manager initialization file 5-6
  - accessing and editing 5-55
  - archive handling parameters 5-52
  - configuring to archive data 8-11
  - copying 5-58
  - deleting 5-58
  - deleting automatically 4-25
  - duplicates checking parameters 5-53
  - Handlers section 5-19
  - introduction to maintaining 5-54
  - Life Cycle settings 13-40
  - line manager parameters 5-35
  - miscellaneous parameters 5-38
  - Parameters section 5-10
  - personality parameters table 5-23
  - routing parameters 5-29
  - sections in 5-9
  - structure overview 5-8
  - translator parameters 5-36
- data manager initialization files
  - location 5-7
- data manager log file 15-11
  - deleting 15-29
  - deleting automatically 4-25
  - purging entries from 15-27
  - viewing 15-14
- data manager pattern
  - length of file name 1-10
- Data Manager Pattern Configuration screen
  - illustration 6-25
- data manager personalities
  - basic types 4-5
  - templates 4-19
- Data Manager Selection panel
  - displaying with hot key 1-18, 6-34
- data manager trading partner/file name configuration
  - copying 6-15
- data manager's status
  - checking 15-13
- data managers
  - configuring for real-time processing 5-82
- dbaudit.i 14-21
- DDF name map 11-25
  - adding 11-43
  - changing 11-76
- default value
  - definition 5-54
- delimiter string
  - using to enclose command card 4-28
- delimiters
  - role in a Gentran:Server script 9-8
  - standard symbols in a script 9-8
- delivery agent
  - function in an inbound NCPDP process flow 3-91
  - function in an inbound process flow 3-31
  - function in an outbound application flow 3-51, 3-71
  - in an inbound NCPDP process flow 3-94
  - in an inbound process flow 3-35
  - in an outbound application flow 3-38, 3-55, 3-58, 3-75
- Delivery Agent dialog box
  - for inbound NCPDP process flow 3-91
  - for inbound process flow 3-31
  - for outbound application flow 3-51, 3-71
- deltrn utility 12-14
  - arguments 12-15
  - command line format 12-15
  - operating options 12-14
- DEST\_Q\_ERROR 5-16
- destination directory 2-5, 2-11, 2-33
- directory name length 1-9
- directory tree 1-7

- DM\_LOCK 5-38
  - document reference number
    - functions 11-5
    - how GENTRAN:Server sets 11-6
    - in Life Cycle event record 14-5
    - overview 11-5
    - picture 11-48
    - purpose of mapping 11-24
    - role in duplicate checking 11-5
    - tasks in defining 11-12
  - Document Reference Number Specifier screen
    - fields and functions 11-13
    - function keys 11-14
    - illustration 11-13
  - document specifier file 2-34
  - document specifier map
    - changing 11-76
  - document specifier table
    - adding 11-16
    - attaching a Trading Partnership code to 11-51
    - changing a Trading Partnership code attachment 11-58
    - copying 11-22
    - creating overview 11-15
    - deleting 11-55, 11-61
    - displaying 11-19
    - locating by Trading Partnership code 11-20
    - maintainence overview 11-57
    - mapping overview 11-24
    - overview 11-10
    - verifying Trading Partnership code attachments 11-55
    - viewing list of attached Trading Partnership codes 11-55
    - ways to create 11-15
  - download data manager 4-5
    - processing stages 2-35
  - downstream data manager
    - as data manager that reads from a queue 7-4
    - configuring to read from a queue 7-13
    - role in real-time processing 5-77
  - DPROCS group
    - description 9-19
    - example 9-19
    - general format 9-19
  - DUPLICATE 5-16
  - duplicate
    - checking sets for 12-5
    - checks performed on sets within routing method 2-36
    - how data managers deal with 12-5
  - duplicates checking parameters (in initialization file) 5-53
- E**
- E\_FILE\_ACCESS 5-17
  - E\_INIT\_XLTR 5-18
  - E\_ISAM 5-17
  - E\_LINE\_MANAGER 5-18
  - E\_LOCK\_FAIL 5-17, 5-76
  - E\_MEM\_ERROR 5-17
  - E\_RW\_SAME\_QUEUE 5-17
  - E\_WRITE\_ERROR 5-17
  - EDI Add screen
    - fields and functions 11-28
    - function keys 11-32
    - illustration 11-28, 11-40
  - EDI data 3-18, 3-79
  - EDI Mapping screen 11-63
    - fields and functions 11-63
    - function keys 11-65
    - illustration 11-63
  - EDI\_ARCHIST 8-17
    - using in the longterm script 8-15
    - using to set archive period 8-16
  - edi\_env
    - sample file 10-22
  - edi\_env file
    - using to control environment 9-35
  - ediarc 3-45, 3-46, 3-66
    - called by translation script 9-46
  - electronic data interchange (EDI) 1-4
  - ENCLOSING\_CHARACTER 5-49
  - end (reserved word) 9-25
  - envelope
    - called by translation script 9-46
  - ENVIRON group
    - description 9-11
    - example 9-12
    - general format 9-11
  - environment file
    - creating 10-22
    - sample 10-22
    - setting environment variables in 10-22
  - environment variables
    - setting for scripts run on a schedule 10-22
  - error class
    - creating and using 5-71
  - Error Handling dialog box 3-98
  - error handling instructions 3-100
  - error names (in an initialization file) 5-16
  - error notification

- setting Handlers section for 5-63
- EXEC\_ERROR 5-17
- EXEC\_FAIL\_ERROR 5-17

## F

- FA\_ERROR 5-17
- FA\_HANDLER 5-44
- FA\_MAIL 5-44
- FIFO 5-78
- file
  - configuring data manager to lock 5-73
- file data manager 4-6
- file definition
  - length of file name 1-10
- file locking
  - controlling concurrent processing with 5-73
  - process 5-75
- 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 13-21
- Flow Identification dialog box 3-14
- flow summary 3-109
- flow summary reports 3-115
  - printing 3-117
  - types 3-115
- Foreground Manager 4-9
  - ability to send termination message 4-9
  - activity log 4-9
  - as parent process 4-9
  - definition 4-9
- Foreground Manager (fmgr)
  - invoking archive handler from 8-7
- foreground process 1-8
- FTP daemon
  - account needed in /etc/passwd 5-84
  - configuring to notify a data manager 5-84
  - file moving process 5-81
- FULL\_TP\_TEST 5-50
- function keys
  - using to start an action 1-17
- Functional Acknowledgment Due report (Life Cycle) 14-37
- Functional Acknowledgment Exception Report (Life Cycle) 14-35
- functional acknowledgments

- columns updated by xlld 14-24
- fields in Life Cycle table 14-14
- how xlld handles 14-23
- how xlld locates Life Cycle record 14-24
- update logic 14-23
- xlld and 14-14

## G

- Gentran
  - Server script
    - deleting 9-52
- GENTRAN:Server
  - displaying version of 1-19
- GENTRAN:Server script
  - configuring a data manager to run 10-5
  - definition 2-13
  - role in process flow 2-13
  - running for errors 10-7
- Gentran:Server script
  - adding to the Permanent Schedule 10-23
  - adding with another editor 9-41
  - adding with the script editor 9-38
  - changing processing schedule for 10-28
  - copying 9-43
  - copying a schedule 10-25
  - creating overview 9-30
  - DATA group 9-17
  - DPROCS group 9-19
  - editing with the script editor 9-50
  - ENVIRON group 9-11
  - guidelines for writing 9-33
  - introduction to running 10-3
  - LOCKS group 9-13
  - overview of running on a schedule 10-14
  - permissions 9-30
  - PROCS group 9-23
  - removing from Permanent Schedule 10-30
  - restarting automatically 10-32, 10-34
  - RESULTS group 9-27
  - running from another script 10-10
  - running from command line 10-9
  - running from the Script Maintenance screen 10-12
  - selecting an editor 9-30
  - STEPS group 9-20
  - tasks in creating 9-32
  - tips 9-35
  - using UNIX mail scripts with 9-54
  - ways to use 9-7
- group (UNIX) 1-6
- group name (in a Gentran:Server script) 9-10

- guidelines for using 9-33
- groups (in a script) 9-9
  - general format 9-10
  - names and functions 9-9

## H

- Handlers section (in initialization file) 5-19
- host command card data manager 4-6, 4-28
  - initialization file parameters for 4-28
  - processing stages of 4-30
- host operating environment 1-6

## I

- if (reserved word) 9-24
- implementation guide
  - length of file name 1-10
- IN\_OUT 5-53
- inbound data manager 4-5
  - processing stages 2-32
- inbound process flow
  - creating with the PCM wizard 3-18, 3-79
- Informix Life Cycle reports
  - running 14-46
- Informix Life Cycle table 13-10
- initialization file
  - accessing and editing 5-55
  - default 5-6
  - definition 5-6
  - deleting 5-58
  - overview of Parameters section 5-22
  - parameters for Life Cycle activity 13-40
  - path name 5-7
  - sections in 5-9
  - standard error names in 5-16
- input file
  - length of file name 1-10
- INPUT\_FN\_APP\_TP 5-48
- intelligent agent 4-5
- interchange segment 11-31
  - setting Occurrence field to extract contents 11-47
- INVALID\_FN 5-17
- IPC Trigger 5-31
  - overview 5-77
- isql
  - required in path to run Informix Life Cycle reports 14-46

## L

- label name (in a Gentran:Server script) 9-10
  - guidelines for 9-33

- label\_name (in a script) 5-12, 5-19
- LCDestInfo table
  - how xlld updates 14-21
  - ODBC translation audit file and 14-21
- lcl 14-15
  - functions 14-11
  - log file 14-16
  - run log 14-16
  - running from the command line 14-17
  - versions 14-16
- lftran 3-18, 3-79
  - called by translation script 9-46
  - Life Cycle and 14-30
- Life Cycle
  - audit files 14-1
  - auditing facility 14-3
  - configuration process 13-25
  - configuring Gentran:Server system profile for using Oracle database 13-30
  - data manager initialization file settings 13-40
  - event file 14-5
  - event record 14-5
  - how xlld handles update failures 14-26
  - inbound database audit records and 14-30
  - load programs 14-11, 14-13
  - process flow diagram 13-25
  - process, how lcl and xlld are called in 14-13
  - records 14-5
  - reports 14-31
    - running overview 14-44
  - setting up for Informix database table 13-34
  - setting up for Sybase database table 13-37
  - setting up to use Oracle database table 13-28
  - using reconciliation IDs in 13-42
  - wetting processing directory 13-40
- Life Cycle command
  - defining the command 13-41
  - setting when it is run 13-41
- Life Cycle event file 2-34, 2-36
- Life Cycle event record 3-18, 3-38, 3-39, 3-58, 3-59, 3-79
- Life Cycle file
  - setting contents 13-40
- Life Cycle load programs 2-37
- Life Cycle reports
  - FA Due Report 14-37
  - FA Exception report 14-35
  - list of 14-31
  - overview 14-31
  - running Informix 14-46
  - running Translation Traffic report 14-49

- TP List report 14-39
- Translation Activity 14-33
- Translation Traffic report 14-41
- Life Cycle table
  - how xlld updates 14-19
- LIFE\_CYCLE\_CALL\_SWITCH 5-32
  - setting timing of Life Cycle command 13-41
- LIFE\_CYCLE\_DIR 5-31, 14-5
  - setting Life Cycle event file directory 13-40
- LIFE\_CYCLE\_EXEC\_LINE 5-33
  - setting Life Cycle command 13-41
- line manager data manager 4-6
- line manager parameters (in initialization file) 5-35
- linker
  - setting path to database linker 13-29
- list
  - selecting an item from 1-16
- LM\_EXEC\_LINE 5-35
- LM\_GRACE\_SECS 5-35
- LM\_INIT\_MODEL 5-35
- LM\_MODEL\_DIR 5-35
- LM\_RUN\_DIR 5-35
- LM\_TERMINATE 5-35
- lock file names
  - adding to LOCKS directory 9-45
- LOCK\_ATTEMPT 5-41
- LOCKS
  - directory 9-45
  - types of 9-13
- LOCKS directory
  - adding lock file names to 9-45
  - contents 9-14
- LOCKS group
  - description 9-13
  - execution order 9-13
- log
  - script 15-39
- log file
  - data manager's 5-7
  - Screen View navigation keys 15-5
  - screen viewer 15-5
  - viewing a data manager 15-14
- login ID, Gentran:Server administrative 1-6
- long-term archive model 8-18
- longterm script
  - adding to the Permanent Schedule 8-17
  - editing 8-19
  - functions 8-15
  - overview 8-17
  - running transaction register purge commands from 12-14

**M**

- mail notification (of script failure) 9-6
- MAIL\_PREFIX\_CONSTRUCT 5-30
- MAIL\_PREFIX\_NAME 5-30
- mail\_proc directory 5-67
- mail\_proc file 3-100
  - deleting automatically 4-25
- Main Menu (Data Flow Administration) 1-12
- map
  - length of file name 1-10
- map picture
  - displaying 11-48
- mapping table
  - length of file name 1-10
- MBAG\_CONSTRUCT 5-29
- meld log
  - example 6-59
- melding
  - definition 6-21
  - tasks 6-30
  - using to generate configuration records 6-57
- menu
  - selecting an option from 1-14, 1-16
- miscellaneous parameters (in initialization file) 5-38
- mksrvdb program 14-9
- monitoring processes 15-1
- MULTIPLE\_LIFE\_CYCLE\_FILES 5-32
- MULTIPLE\_LIFE\_CYCLE\_FILES 14-6
  - setting Life Cycle file scope with 13-40

**N**

- named pipe 5-78
- NCPDP Add screen 11-40
  - fields and functions 11-40
  - function keys 11-42
  - illustration 11-40
- NCPDP Mapping screen 11-72
  - fields and functions 11-72
- NCPDP-to-application process flow 3-79
- NCPDP-to-standard process flow 3-79
- NO\_CONFIG 5-16
- NO\_TP 5-16
- non-shareable resources
  - definition 9-13
- NOTIFY 5-31
  - role in real-time processing 5-78

**O**

- Occurrence field (on EDI and APP screens) 11-46
  - purpose 11-46

- ODBC.ini 14-21
  - operating environment (of host) 1-6
  - Oracle C compiler
    - requirement for Life Cycle 13-28
  - Oracle Life Cycle table 13-14
  - org.dat/idx
    - file record layouts 13-21
  - outbound application flow
    - agents in a PCM 3-38
    - creating with the PCM wizard 3-38
    - input file names for 3-38
  - outbound data manager
    - processing stages 2-35
  - output file
    - length of file name 1-10
- P**
- parameters
    - data manager personalities and 5-23
  - Parameters section (in initialization file) 5-10
  - PARMS section (in initialization file) 5-10
  - PATH 1-14
  - path name 1-7
  - pattern
    - adding another data manager to 6-45
    - copying 6-37
    - copying a configuration record in 6-51
    - creating a new 6-31
    - definition 6-19
    - deleting 6-54, 6-56
    - editing 6-47
    - using categories in 6-24
    - using to create configuration records 6-19
  - pattern configuration record
    - adding 6-47
    - copying in a pattern 6-51
    - editing 6-54
  - Pattern Configuration screen
    - illustration 6-25
  - PCM process flow
    - agents in an inbound 3-18
    - agents in an inbound NCPDP 3-79
    - agents in outbound 3-38
    - beginning a 3-11
    - creating an inbound 3-18
    - creating an inbound NCPDP 3-79
    - creating an outbound application 3-38
    - creating supporting files for 3-13
    - deleting 3-124
    - deleting Trading Partnerships from 3-122
    - delivery agent for inbound 3-35
    - delivery agent for inbound NCPDP 3-94
    - delivery agent for outbound application 3-55, 3-75
    - editing 3-120
    - naming and describing 3-16
    - processing agent for inbound 3-29
    - processing agent for inbound NCPDP 3-90
    - processing agent for outbound application 3-49, 3-69
    - source agent for inbound 3-23
    - source agent for inbound NCPDP 3-84
    - source agent for outbound application 3-43
    - types 3-8
  - Permanent Schedule
    - adding a Gentran:Server script to 10-23
    - changing a script's schedule 10-28
    - changing a scripts schedule 10-28
    - copying a script's schedule 10-25
    - default environment 10-22
    - deleting a script from 10-30
    - function 10-15
    - need for environment file 10-15
    - overview 10-15
    - overwriting crontab files 10-15
  - Permanent Schedule Maintenance screen
    - fields and functions 10-16
    - function keys 10-18
    - illustration 10-16
  - Permanent Schedule screen
    - entering a schedule on 10-21
    - fields and functions 10-19
    - illustration 10-19
  - permissions (UNIX file) 1-6
  - PERSONALITY 5-42
  - primary control 4-16
    - how GENTRAN:Server grants 4-16
  - pro\*c
    - in Oracle bin directory 13-28
  - Process Control Manager wizard 3-5
    - purpose 3-5
  - process flow
    - adding components to 2-22
    - agents in a PCM outbound application 3-38
    - completing a PCM 3-97
    - components of a PCM 3-7
    - creating an inbound 3-18
    - creating an inbound NCPDP 3-79
    - 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 2-5, 3-7
- deleting a PCM 3-124
- deleting Trading Partnerships from PCM 3-122
- delivery agent in an inbound flow 3-35
- delivery agent in an inbound NCPDP flow 3-94
- delivery agent in an outbound application flow 3-55, 3-75
- designing 2-15
- editing a PCM 3-119, 3-120
- error handling instructions for 3-100
- example inbound 2-11
- example of designing inbound 2-23
- example with script 2-14
- identifying purpose of 2-16
- maintaining PCM 3-119
- naming and describing a PCM 3-16
- processing agent in an inbound flow 3-29
- processing agent in an inbound NCPDP flow 3-90
- processing agent in an outbound application flow 3-49, 3-69
- selecting data managers for 2-18
- source agent in an inbound flow 3-23
- source agent in an inbound NCPDP flow 3-84
- source agent in an outbound application 3-43
- stages of inbound 2-32
- stages of outbound 2-35
- two basic types of 2-5
- types of PCM flows 3-7
- worksheet 2-28
- process flow summary 3-109
- process flow summary reports 3-115
- process ID number (PID) 15-19
- process log 2-34, 2-36, 15-11
- PROCESS\_TA1 5-49
- processing agent
  - function in an inbound NCPDP process flow 3-90
  - function in an inbound process flow 3-29
  - function in an outbound application flow 3-45, 3-65
  - in an inbound NCPDP process flow 3-79, 3-86
  - in an inbound process flow 3-18, 3-25
  - in an outbound application flow 3-38, 3-49, 3-58, 3-69
- Processing Agent dialog box
  - for inbound NCPDP process flow 3-86
  - for inbound process flow 3-25
  - for outbound application flow 3-45, 3-65
- PROCS group
  - description 9-23

- putenv (reserved word) 9-25

## Q

- Q\_FILE\_MISSING 5-16
- Qualifier field (on EDI and APP screens) 11-46
  - purpose 11-46
- queue 2-11
  - adding an entry on the Queue File screen 7-26
  - adding an entry with the `srvr_enq` command 7-26
  - benefits of using 7-5
  - configuring downstream data manager to read from 7-13
  - configuring GENTRAN:Server to use 7-10
  - creating a new 7-11
  - definition 7-4
  - deleting 7-32
  - deleting an entry using the Queue File screen 7-28
  - deleting an entry with the `srvr_deq` command 7-28
  - information found in a 7-4
  - length of file name 1-10
  - names 7-5
  - removing from the Select list 7-31
  - searching for an entry in 7-23
  - selecting from the queue Select screen 7-7
  - viewing entries in 7-22
- Queue File screen
  - deleting an entry from 7-28
  - fields and functions 7-8
  - function keys 7-9
  - illustration 7-8
- queue process 7-6

## R

- real-time processing
  - configuring data managers for 5-82
  - stages in 5-78
- REC\_ID\_MISS 5-47
- reconciliation IDs
  - using alternates for application or download data managers 13-42
- recover script
  - adding to the `inittab` or `rc` initialization file 10-36
  - location 10-33
  - modifying 10-34
  - modifying to include script name 10-34
  - overview 10-33
  - role in restarting scripts 10-33
  - tips for using 10-33

- using to generate mail notices 10-33
  - recover.scr
    - see recover script 10-32
    - using to restart Gentran:Server scripts 10-33
  - relational data base 14-8
  - release (reserved word) 9-25, 9-35
  - REPORT\_SET\_ERRORS 5-33
  - requirements
    - incorporating new 2-22
  - reserved words (in a Gentran:Server script) 9-24
    - build 9-25
    - compare 9-25
    - else 9-24
    - end 9-25
    - if 9-24
    - putenv 9-25
    - release 9-25
    - then 9-24
  - RESOURCE\_GROUP 5-39
  - RESULTS group
    - avoiding blank lines in 9-33
    - choosing steplabels for 9-28
    - description 9-27
    - example 9-28
    - general format 9-27
  - root directory 1-7
  - routing method
    - transaction register entries and 12-4
  - routing parameters (in initialization file) 5-29
  - ROUTING\_ERROR 5-16
  - ROUTING\_METHOD 5-29
  - rtv\_arc.l 8-26
  - rtv\_arc.l log
    - contents 8-34
    - displaying 8-34
    - example 8-34
- S**
- SCAN\_DELAY 5-40
  - schedule
    - changing for a Gentran:Server script 10-28
    - copying a run schedule for a Gentran:Server script 10-25
  - Screen Viewer 15-5
    - function keys 15-5
    - navigation keys 15-5
  - script
    - adding a Gentran:Server 9-38
    - adding a Gentran:Server script to the Permanent Schedule 10-23
    - checking the status of 15-33
    - deleting a Gentran:Server 9-52
    - displaying usage statistics for a Gentran:Server 10-32
    - enabling a data manager to invoke a GENTRAN:Server 10-5
    - GENTRAN
      - Server
        - role in a process flow 2-13
    - GENTRAN:Server 2-13
    - Gentran:Server, description of 9-5
    - guidelines for writing a Gentran:Server 9-33
    - length of file name 1-10
    - processing on a schedule 10-1
    - removing a Gentran:Server script from the Permanent Schedule 10-30
    - running a Gentran:Server script from the command line 10-9
    - running a Gentran:Server script from the Script Maintenance screen 10-12
    - running a GENTRAN:Server script under specific conditions 10-7
    - tasks in creating a Gentran:Server 9-32
    - tips for using a Gentran:Server 9-35
    - using a Gentran:Server script to invoke another script 10-10
    - ways of starting a Gentran:Server 9-8
    - ways to run Gentran:Server 10-3
  - script directory 9-5, 9-31
    - effect of deleting script from 9-52
  - script editor
    - set in \$EDITOR environment variable 9-50
  - script journal 15-32
    - deleting 15-41
    - purging entries from 15-39
    - viewing 15-35
  - script journal file 15-31
  - script library
    - contents 9-31
  - script log 15-31
    - erasing 15-41
    - viewing 15-35
  - script log file 15-31
    - deleting 15-41
    - maintaining overview 15-38
  - Script Maintenance screen
    - fields and functions 9-36
    - function keys 9-37
    - illustration 9-36
    - running a script from 10-12
  - Script Manager 2-37
    - overview 9-6

- script name
    - effect of deleting from the library list 9-52
  - script usage statistics
    - viewing 15-36
  - SCRIPT\_RUN\_SWITCH 5-48
    - setting to control when script is run 10-6
  - scripts
    - monitoring overview 15-30
  - secondary control 4-16
  - Select list
    - removing a queue from 7-31
  - Select screen (for queues)
    - function keys 7-7
    - illustration 7-7
  - SEND\_MAIL\_TP 5-46
  - Set ID map 11-24
    - adding 11-43
    - changing 11-76
    - deleting from a document specifier table 11-79
  - shell
    - definition of 1-8
  - shortcut keys 1-18
  - source agent
    - in an inbound NCPDP process flow 3-79, 3-84
    - in an inbound process flow 3-18, 3-23
    - in an outbound application flow 3-38, 3-43, 3-58
  - source directory 2-5
  - Source Setup dialog box
    - for inbound NCPDP process flow 3-81
    - for inbound process flow 3-20
    - for outbound application flow 3-40, 3-60
  - sql compiler
    - requirements for Informix Life Cycle table 13-34
  - SQL error 13-27
  - SRC\_Q\_ERROR 5-16
  - svr\_deq command
    - arguments 7-29
    - deleting a queue entry with 7-28
    - format 7-28
  - svr\_enq command
    - adding entries to a queue with 7-26
    - arguments 7-27
    - format 7-26
  - standard error names (in an initialization file) 5-16
  - standard-to-application process flow 3-18
  - standard-to-standard process flow 3-18
  - startserver
    - functions 15-23
    - when to use 15-23
  - startserver.sh
    - functions 15-23
  - script recovery 15-23
    - when to use 15-23
  - STAT\_CHECK 5-41
    - setting for downstream data manager 5-83
  - status
    - checking data manager 15-18
    - viewing a data manager's 15-13
  - status field (on Data Manager Control screen) 15-19
  - STEPS group
    - description 9-20
    - example 9-21
    - general format 9-20
    - relationship to PROCS group 9-20
  - stopserver
    - function 15-20
    - when to use 15-20
  - stopserver.sh
    - function 15-20
    - when to use 15-20
  - STRUCT\_ERROR 5-16
  - SUBSTITUTE\_NEWLINE 5-43
  - supporting files
    - creating for a PCM process flow 3-12
  - SWAP\_ID\_FLAG 5-42
  - Sybase Life Cycle table 13-18
- ## T
- TA1 segment
    - what happens when a data manager encounters 2-32
  - then (reserved word) 9-24
  - third-party products 1-8
  - time of day
    - displaying 1-18
  - tokens
    - in pattern configuration records 6-19
    - list of 6-22
    - what they can represent 6-19
  - TP Code modifier
    - defining on Data Manager Pattern Configuration screen 6-26
    - defining on the Data Manager Configuration screen 6-9
  - TP search key 1-20
  - tracker command
    - arguments 14-49
    - command line format 14-49
    - running Translation Traffic report with 14-49
  - tracker program 14-41
  - tracking data (with Life Cycle) 14-1

- Trading Partner List report (Life Cycle) 14-39
  - Trading Partner Records dialog box 3-102
  - Trading Partnership record
    - mail message for auto-created 6-62
    - searching for 1-20
  - Trading Partnership Search options screen 1-21
  - TRADING\_PARTNER\_DIR 5-39
  - TRANS\_REGISTER\_DIR 5-39
    - updating prior to deleting a transaction register 12-20
    - using to set name of transaction register directory 12-3, 12-18
  - transaction register 2-35
    - adding a new 12-18
    - adding an entry to 12-11
    - contents 12-3
    - default 12-3
    - deleting 12-20
    - deleting an entry from 12-13
    - deleting entries in 12-14
    - deleting multiple entries from 12-14
    - description 12-3
    - duplicate entries 12-4
    - how data manager finds 12-18
    - how data managers make entries to 12-4
    - length of file name 1-10
    - purge utilities 12-14
    - purging with the clean\_trn utility 12-16
    - purging with the deltrn utility 12-14
    - searching for an entry in 12-9
    - sets written to 2-34
    - uses 12-3
    - using multiple 12-3
    - working with a single 12-9
    - working with multiple 12-18
  - Transaction Register screen 12-6
    - fields and functions 12-6
    - function keys 12-7
    - illustration 12-6
  - Transaction Register Search screen 12-8
    - fields and functions 12-8
    - function keys 12-8
    - illustration 12-8
  - Translation Activity report (Life Cycle) 14-33
  - translation audit record
    - contents of a database 14-30
  - translation data manager 4-6
  - Translation Options dialog box 3-27, 3-47, 3-67, 3-88
  - translation script 3-18, 3-38, 3-45, 3-58, 3-79
    - creating 9-47
    - definition 9-46
    - Life Cycle activities 14-18
    - naming convention for 9-46
  - Translation Traffic report (Life Cycle) 14-41
  - TRANSLATION\_FAILED 5-16
  - translator 3-18, 3-38, 3-49, 3-69, 14-13
    - called by translation script 9-46
    - relationship to Data Flow Administration 1-5
  - translator parameters (in initialization file) 5-11, 5-36
  - translator program lfran 3-79
  - TRIM\_TERMINATORS 5-34
  - trn.dat/idx
    - default transaction register 12-3
- ## U
- UNIQUE\_FILE\_NAMES 5-31
  - UNIX directory structure 1-7
  - UNIX file permissions 1-6
  - UNIX mail script
    - adding to a Gentran/Server script 9-55
    - deleting 9-57
    - editing 9-55
    - example of 9-54
    - location 9-54
    - overview 9-54
    - using to send messages 5-67
    - using with Gentran:Server scripts 9-54
  - UNIX operating environment 1-6
  - UNIX shell
    - accessing from a GENTRAN:Server screen 1-18
  - UNIX user group 1-6
  - upstream data manager
    - as data manager that writes to a queue 7-4
    - role in real-time processing 5-77
  - USE\_FN\_PRE\_ON\_ERR 5-31
  - USE\_MOD 5-48
  - USE\_RECON\_IDS 5-43
    - setting Life Cycle values 13-42
- ## V
- VAL\_RESOURCE\_GRP 5-39
  - variables
    - defining in a Gentran:Server script 9-11
    - guidelines for using in a Gentran:Server script 9-34
    - referencing in a Gentran:Server script 9-12
  - version (GENTRAN:Server)
    - displaying 1-19

VT100 terminal emulator  
function key emulation 1-17

## W

work directory 2-11  
WORK\_DIRECTORY 5-38  
setting for translation data manager 9-49  
WORK\_TYPE 5-38

## X

XL\_ARGS 5-37  
XL\_LINK\_DIRS 5-36  
XL\_LINK\_FILES 5-36  
XL\_MAKE\_DIRS 5-37  
XL\_MODE 5-36  
XL\_MODEL\_SCR 5-36  
setting for translation data manager 9-49  
using to specify name of translation script 9-46  
XL\_RUN\_DIR 5-36  
setting for translation data manager 9-49  
xlld 14-13  
called by translation script 9-46

command line format 14-28  
Life Cycle columns updated by 14-19  
log file 14-26, 14-27  
overview 14-18  
running from the command line 14-28  
running from translation scripts 14-18  
source of translation and transaction detail 14-21  
types of updates 14-20  
viewing log file 14-27  
ways to call 14-18  
xlld.pc  
checking revision number 13-28  
XML Add screen 11-37  
fields and functions 11-37  
function keys 11-39  
illustration 11-37  
XML Mapping screen  
fields and functions 11-69  
function keys 11-70  
purpose 11-69

