

Gentran:Server[®] for Windows[®]

Application Integration User's Guide

Version 5.2

Sterling Commerce
An IBM Company

Copyright Notice

Gentran:Server for Windows

© Copyright 1995–2008
Sterling Commerce, Inc.
ALL RIGHTS RESERVED

Sterling Commerce Software Trade Secret Notice

THE GENTRAN:SERVER FOR WINDOWS 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. Gentran and Gentran: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").

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

Preface	About This Guide	
	• Introduction	viii
	• What's in This Manual	ix
	• Getting Support	xi
Chapter 1	Application Integration Basics	
	Introducing Mapping	1-2
	• Key Terms and Concepts	1-2
	• How to Use Application and Forms Integration	1-4
	• Inbound and Outbound Translation Process	1-5
	• Application File Format	1-6
	• EDI File Format	1-7
	Introducing the Application Integration Subsystem	1-10
	• Overview	1-10
	• How to Navigate in Application Integration	1-11
	• How to Build a Map with Application Integration	1-12
Chapter 2	Designing your Map	
	Overview	2-3
	• Introduction	2-3
	• Preparation and Analysis	2-4
	Setting Global Defaults	2-6
	• Customizing Your Display	2-6
	• How to Customize Global Display Options	2-7
	• How to Customize Global Colours	2-9
	• How to Customize Global Fonts	2-11
	• How to Customize Global Display of Links	2-13
	• How to Customize the Auto-increment Map Version	2-15
	• How to Make the Two Sides of a Map Equal	2-17
	• How to Set the Default Date Format	2-18
	Creating a Map and Defining Details	2-20
	• How to Create a New Map	2-20
	• How to Define Translation Object Details	2-27
	• How to Load a File Definition	2-34
	• How to Save a File Definition	2-36

•	How to Activate EDI Map Components	2-37
•	How to Use Auto Trim	2-39
•	How to Promote Groups and Repeating Segments	2-41
•	How to Split Groups and Repeating Segments	2-44
•	How to Use Copy, Cut, and Paste	2-48
	Using a Fixed-Format File	2-51
•	Overview	2-51
•	How to Adjust File Format Properties	2-52
•	How to Create the First Record	2-54
•	How to Create Subsequent Records	2-56
•	How to Create a Temporary Record	2-58
•	How to Create a Group	2-62
•	How to Create Fields	2-64
	Using an EDI File	2-68
•	Overview	2-68
•	How to Verify EDI Delimiters	2-69
•	How to Modify Group Properties	2-71
•	How to Modify Segment Properties	2-73
•	How to Use Loop Start and Loop End Segments	2-76
•	How to Modify Composite Properties	2-82
•	How to Modify Element Properties	2-84
	Using a CII File	2-89
•	Overview	2-89
•	How to Modify CII File Properties	2-90
•	How to Create a Group	2-92
•	How to Create a TFD	2-94
	Formatting Your Data	2-98
•	Overview	2-98
•	How to Use the String Type	2-99
•	How to Delete a Syntax Token	2-105
•	How to Use Syntax Tokens	2-106
•	How to Use the Number Type	2-108
•	How to Use the Date/Time Type	2-110
	Completing a Map	2-114
•	How to Create Simple Links	2-114
•	How to Map Binary Segments	2-115
•	How to Set the Auto-register Option	2-118
•	How to Compile a Map	2-120
•	How to Compile Maps Using the Command Line	2-123
•	How to Print a Mapping Report	2-124

Chapter 3 Using Standard Rules

• Introduction	3-3
Using the Select Standard Rule	3-4
• Overview	3-4
• How to Use Information from the Partner Definition	3-9
• How to Use Information from a Cross-reference Table	3-11
• How to Use Information from Location Tables	3-14
• How to Use Information from Lookup Tables	3-16
Using the Update Standard Rule	3-17
• Overview	3-17
• How to Set Up the Document Name and Reference	3-19
Using the System Variable Standard Rule	3-21
• Overview	3-21
• How to Use the System Date and Time	3-22
Using the Use Constant Standard Rule	3-23
• Overview	3-23
• How to Use a Constant in a Map	3-27
• How to Define a Qualifying Relationship	3-28
• How to Define and Edit Literal Constants	3-30
• How to Delete Literal Constants	3-31
• How to Map Literal Constants	3-32
• How to Generate Qualifiers	3-33
Using the Loop Count Standard Rule	3-34
• Overview	3-34
• How to Use the Loop Count Function	3-35
Using the Use Accumulator Standard Rule	3-36
• Overview	3-36
• How to Count Line Items	3-42
• How to Calculate Hash Totals	3-45
• How to Reset and Calculate a Value Total	3-48
Using the Use Code Standard Rule	3-56
• Overview	3-56
• How to Validate Data Against a Code List Table	3-61
• How to Define and Modify a Code List	3-62
• How to Delete a Code List or Code List Entry	3-63
• How to Import a Code List	3-64
• How to Export a Code List	3-65
• How to Load a Code List Table from the Standard	3-66
• How to Copy and Paste Code Lists	3-67
• How to Validate Data Against Code List Tables	3-68
• How to Map Code Item Descriptions	3-69

Chapter 4	Setting Up the Import Process	
	Overview	4-2
	Importing Data	4-3
	How to Modify System Configuration	4-5
	How to Create a System Import Map	4-6
	How to Define the Six-Field Key	4-8
	How to Define the Alternate Key	4-15
	How to Compile the System Import Translation Object	4-20
Chapter 5	Setting Up the Export Process	
	Overview	5-2
	Inbound Process Prior to Exporting Data	5-3
	How to Use Supplementary Envelope Information	5-5
Chapter 6	Using Extended Rules	
	Introducing Extended Rules	6-4
	Overview	6-4
	Declarations and Initialization	6-5
	Statements	6-7
	When Extended Rules are Processed	6-8
	Defining Extended Rules	6-12
	Overview	6-12
	How to Define a Session Rule	6-13
	How to Define a Map Component Rule	6-15
	Extended Rule Syntax	6-17
	Overview	6-17
	Keywords and Commands	6-18
	Operators	6-20
	Symbols	6-21
	Common Statements and Examples	6-25
	Overview	6-25
	Assignment	6-26
	Datetime Expressions	6-27
	Conditional Logic	6-29
	String Conditions and Functions	6-31
	Numerical Functions	6-33
	Raise Compliance Error Function (CERROR)	6-34
	Remove Field Value Function (EMPTY)	6-35
	Existence of Data Function (EXIST)	6-36
	Count Function (COUNT)	6-37
	Delete Function (DELETE)	6-38
	File Pointer Functions (FSEEK, FTELL)	6-39
	Block of Data Functions (READBLOCK, WRITEBLOCK)	6-40

▶ Select Function (SELECT)	6-41
▶ Update Function (UPDATE)	6-42
▶ Insert Function (INSERT)	6-43
▶ User Exit Function (EXEC)	6-44
▶ ActiveX and User Exit Functions	6-45
Alphabetic Language Reference	6-47
▶ Overview	6-47
▶ atoi	6-48
▶ aton	6-49
▶ auditlog	6-50
▶ begin	6-52
▶ break	6-53
▶ cerror	6-54
▶ concat	6-61
▶ continue	6-62
▶ count	6-63
▶ createobject	6-64
▶ date	6-65
▶ delete	6-67
▶ deleteobject	6-68
▶ empty	6-69
▶ end	6-70
▶ exec	6-71
▶ exist	6-72
▶ fseek	6-73
▶ ftell	6-74
▶ get	6-75
▶ getiid	6-76
▶ if then...else	6-77
▶ index	6-78
▶ insert	6-79
▶ left	6-81
▶ len	6-82
▶ messagebox	6-83
▶ mid	6-85
▶ ntoa	6-86
▶ param	6-87
▶ queryobject	6-88
▶ readblock	6-89
▶ readbytes	6-90
▶ right	6-91
▶ select	6-92

▶ set	6-93
▶ strdate	6-94
▶ strstr	6-96
▶ unreadblock	6-97
▶ update	6-98
▶ while...do	6-99
▶ winexec	6-100
▶ writeblock	6-102
▶ writebytes	6-103
▶ Select and Update Available Options	6-104

Appendix A Error Messages

▶ Overview	A-2
▶ Compile Error Messages	A-3
▶ Gentran:Server Error Messages	A-16
▶ Translator Report Error Messages	A-22

Appendix B User Exits

▶ Overview	B-2
▶ Definition of ActiveX Terminology	B-3
▶ Gentran:Server User Exit Overview	B-4
▶ Examples of User Exits	B-7
▶ Examples of Automation Servers	B-12
▶ How to Create a User Exit	B-15

Appendix C Using GentranEx.DLL

▶ Introduction	C-2
Database Access	C-3
▶ Overview	C-3
▶ Implementing Database Access	C-4
Debugging	C-5
▶ Overview	C-5
▶ Implementing Debugging	C-6
Rules Extensions	C-8
▶ Overview	C-8
▶ Implementing Rules Extensions	C-9

Appendix D Using the Translator Command Line Interface

▶ Introduction	D-2
▶ Command Line Syntax	D-3

Glossary

About This Guide

Contents

▶ Introduction	viii
▶ What's in This Manualix
▶ Getting Supportxi

Introduction

Welcome Welcome to Gentran:Server® Windows® Application Integration subsystem, Sterling Commerce's Electronic Data Interchange (EDI) mapping software for the Windows operating system.

Gentran:Server provides you with the tools you need to translate:

- your application files to EDI standard formats for documents you send to your partners (outbound mapping)
- EDI standard formats to your application format for documents that you receive from your partners (inbound mapping)

We believe that you will find this software and the supporting materials easy to use and directly beneficial to your business.

What's in This Manual

Assumptions

This manual assumes that you are familiar with using a PC and with Microsoft® Windows functions, including:

- The terminology used to describe mouse and cursor actions.
- The terminology used to describe Windows-specific attributes, including dialog boxes, icons, windows, and buttons.

This manual also assumes that you are familiar with EDI concepts, EDI standard structure, your internal application format, data mapping concepts, and the Gentran:Server product.

Description of contents

The majority of the Gentran:Server Application Integration documentation is contained in the online Help system. This manual is intended to support the online Help by assisting you in performing various tasks in the Application Integration subsystem. This task-oriented approach is intended to answer any questions you may have about Gentran:Server with step-by-step instructions.

The *Application Integration User's Guide* is organized into chapters. A brief description of the chapter contents follows.

- *Chapter 1, Application Integration Basics*, explains the content, organization, and conventions in this guide, as well as the system requirements. This chapter also describes how to use the extensive online Help system, how to get technical support, and a brief outline of how to build maps using the Application Integration subsystem, and gives you an introduction to mapping and Gentran:Server for Windows.
- *Chapter 2, Designing your Map*, explains how to perform basic functions in the Application Integration subsystem, including creating a new map; using fixed-format, EDI, and CII files; opening, saving, and naming a map; compiling a map; and printing a mapping report. This chapter also explains customizing display options; setting the default date format; selecting which map components to use; cutting, copying, and pasting; setting up the document name and reference; formatting your data; and using field-to-field linking.
- *Chapter 3, Customizing the Properties of Map Components*, explains how to customize the properties of record, segments, looping structure, fields, elements, composite elements, and TFDs. This chapter also explains how to use loop start (LS) and loop end (LE) segments.
- *Chapter 4, Using Standard Rules*, explains how to use the Application Integration standard rules: select, update, constant, use accum, loop count, and use code. This chapter also illustrates examples of when you would use certain standard rules.
- *Chapter 5, Setting Up the Import Process*, explains all the facets of the import process, defines import file restrictions, and describes how to create a system import map.
- *Chapter 6, Setting Up the Export Process*, explains all the export process and describes how to use envelope information.

(Continued on next page)

Description of contents (contd)

- *Chapter 7, Using Extended Rules*, defines what extended rules are and when they are used, and explains the proprietary programming language syntax.
- *Appendix A, Error Messages*, describes Gentran:Server - Application Integration error messages and specifies corrective actions.
- *Appendix B, User Exits*, explains how to use user exits with Gentran:Server.
- *Appendix C, Using GentranEx.DLL*, describes how to use GentranEx.DLL, the Gentran:Server in-process automation server.
- *Appendix D, Using the Translator Command Line Interface*, describes how to use the command line to invoke the services of the TX32.EXE translator to perform translation outside Gentran:Server, including generating a translator report in XML format.
- The *Glossary* contains mapping and Gentran:Server terms and concepts.

Online Help System

The majority of the documentation for Gentran:Server is contained in the online Help system. This includes all the dialog box element definitions, detailed processing information, and all the “how to” information from this manual.

Getting Support

Introduction Sterling Commerce's Gentran:Server software is supported by trained product support personnel who are available to help you with product questions or concerns.

Note

Gentran:Server Customer Support does not support non-Sterling Commerce products (e.g., SQL Server, Oracle, etc.), but can assist you in configuring non-Sterling Commerce products to work with Gentran:Server.

Phone number For assistance, please refer to your *Getting Started Guide* to determine which support phone number you should use.

Before calling support

To help us provide prompt service, we ask that you do the following:

- Attempt to recreate any problem that you encounter and record the exact sequence of events.
- When you call product support, you should be prepared to provide us with the information below.

Information	Description
Identification	Your company name, your name, telephone number and extension, and the case number (if the question refers to a previously reported issue).
System Configuration	The Gentran:Server version (and any service packs installed) and information about the primary Gentran system controller and all machines experiencing problems, including: the Windows operating system version, amount of memory, available disk space, database version, Microsoft Data Access (MDAC) version, and Internet Explorer version. Also, please describe any recent changes in your hardware, software, or the configuration of your system.
System Data Store	Which machines contain folders in the system data store?
Error Messages	Record the exact wording of any error messages you receive and the point in the software where the error occurred, as well as any log files.
Attempted Solutions	Record any steps that you took attempting to resolve the problem and note all the outcomes, and provide an estimate on how many times the problem occurred and whether it can be reproduced.

Accessing the Sterling Commerce Support Web Site

The Sterling Commerce Customer Support Web Site contains valuable information about getting support for Gentran:Server for Windows, including the:

- scope of support services,
- customer support policies,
- call prioritizing,
- customer support phone directory,
- how to create new Support on Demand cases,
- how to check the status of Support on Demand cases, and
- how to add information to Support on Demand cases.

The Customer Support Web Site is constantly updated and all Sterling Commerce customers have access to it. This web site also contains the most recent product updates and is a valuable source of product information.

Reference

Refer to the *Getting Started Guide* for information about how to access the Customer Support website.

Documentation

The Customer Support Web Site contains a documentation library, which has the entire Gentran:Server for Windows documentation set. You can download the product manuals in PDF format from this library at any time.

Application Integration Basics

Contents	Introducing Mapping	1 - 2
	▶ Key Terms and Concepts	1 - 2
	▶ How to Use Application and Forms Integration	1 - 4
	▶ Inbound and Outbound Translation Process	1 - 5
	▶ Application File Format	1 - 6
	▶ EDI File Format	1 - 7
	Introducing the Application Integration Subsystem	1 - 10
	▶ Overview	1 - 10
	▶ How to Navigate in Application Integration	1 - 11
	▶ How to Build a Map with Application Integration	1 - 12

Introducing Mapping

Key Terms and Concepts

Map A map is a set of instructions you define in the Gentran:Server Application Integration subsystem that indicates the corresponding relationship between your application file and the EDI standards (including the Japanese Center for the Information of Industry (CII) standard), and defines how the system should translate data.

Inbound mapping For inbound mapping, the system translates your trading partner's EDI standard formatted business document to your application file format. To translate inbound data, you need to create an export map in Gentran:Server Application Integration (because the system is *exporting to* your application file).

Outbound mapping For outbound mapping, the system translates your application file format to EDI standard formats, so you can send documents to your partners. To translate outbound data, you need to create an import map and a system import map in the Gentran:Server Application Integration subsystem.

Import map

You need to create an import map because the system is *importing from* your application file. The import map defines how to move data from your application file (flat file definition), which may contain multiple documents, to the EDI standard-formatted documents that your partners expect to receive from you.

System import map

You need to create a system import map to determine which trading relationship (established in Partner Editor) corresponds to each document in the application file, so the system knows which import map to use to process the document.

Reference

See *How to Create a System Import Map* on page 4 - 6 for more information.

Source After you create a map in the Application Integration subsystem, this map is the *source* from which you compile a translation object.

Translation object

A translation object is a set of rules that instruct the translator how to convert a file from one format to another. Every function performed during translation requires a translation object. Translation objects are used to control all the functions of Gentran:Server EDI translation. Translation objects can belong to the following two categories: system translation objects or user translation objects.

System translation objects

System translation objects control the creation and separation of interchanges, functional groups, and transaction sets. They are also used to generate and reconcile functional acknowledgements.

User translation objects

User translation objects control data entry, importing, exporting, document turnaround, and creating printed reports. All the required system translation objects are automatically installed with the Gentran:Server system.

How to Use Application and Forms Integration

Introduction

The Gentran:Server Application Integration subsystem enables you to generate import, export, and document turnaround translation objects. The Gentran:Server Forms Integration subsystem enables you to generate data entry and print translation objects.

Registering translation objects

Once you generate a translation object using the Gentran:Server Application Integration subsystem (or the Gentran:Server Forms Integration subsystem), you must register the translation object with Gentran:Server.

Reference

See Using Translation Objects in the *Gentran:Server for Windows User's Guide* for more information.

Partner relationships

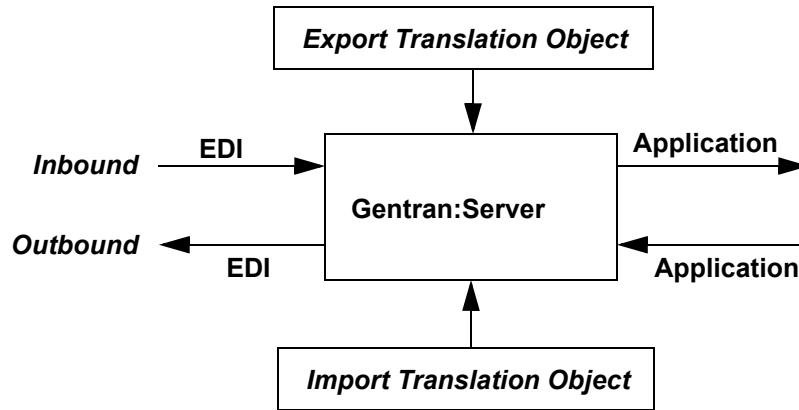
In addition, you must establish correct trading relationships for all trading partners from which you receive or to which you send data. During the process of establishing a trading relationship, you need to specify the name of the appropriate translation object that the translator uses to translate the data.

Reference

See Using Partners in the *Gentran:Server for Windows User's Guide* for more information.

Inbound and Outbound Translation Process

Diagram This illustration depicts the inbound and outbound translation process:



Outbound translation When you create an import map, you need to define to Gentran:Server your application file format (your data that you send to your partners) and the EDI standard format in which your partners expect to receive the documents.

Inbound translation When you create an export map, you need to define to Gentran:Server the EDI standard format in which your partners sends documents to you and your application file format (how the data needs to be formatted for your application to process it).




Application File Format

Introduction

If you are creating an import or export map, you must define your application to the Gentran:Server Application Integration subsystem. In Gentran:Server terminology, your application file is also referred to as a *fixed-format file* or a *positional file*. Your application file must contain all the information that you either need to extract from your partner's document (if the map is inbound) or need to send to your partner (if the map is outbound), so that your system can accurately process the data.

Map components

This table lists the icons that visually represent the map components you use to define your application file.

Icon	Description
	<p>A group is a looping structure that contains related records and/or groups that repeat in sequence until either the group data ends or the maximum number of times that the loop is allowed to repeat is exhausted.</p> <p>If you create a group that is subordinate to another group (a subgroup), this corresponds to a nested looping structure (a loop within a loop).</p>
	<p>A record contains a group of related fields. A record can occur once or can repeat multiple times.</p>
	<p>A field is the smallest piece of information defined in the application file. A field is the application map component that is mapped (linked) to a corresponding EDI element to move data to and from your application file.</p> <p>When a field has a mapping operation performed against it, a red checkmark is displayed over the field icon.</p>

Before you begin

Before you define your application, you should obtain a layout of the necessary records, fields, and groups. Each map component is arranged sequentially in the order that it is most logical for the system to process. Therefore, each level of your application must be created sequentially.

Example

Your application file contains records and groups. The records contain fields and the groups contain records and/or subgroups. This means that you must create records and groups before you create the subordinate fields.

Reference

See *Using a Fixed-Format File* on page 2 - 51 for more information about defining your application file format.

EDI File Format

Introduction

The EDI file must contain all the information that you expect to receive from your partner (if the map is inbound) or need to send to your partner (if the map is outbound), so that your system can accurately process the data.

For non-CII standards, the Gentran:Server Application Integration subsystem generates an EDI file for you, based on the standard (agency), version, transaction set, and release (for the TRADACOMS standard *only*) you selected. The system includes all the groups, segments, composites, and elements that are defined by the standards agency for the version of the document you selected. Please see *How to Create a New Map* on page 2 - 20 for more information about generating an EDI file when you create a new map.

If you are creating an import or export map, you typically need to customize the system-generated EDI file by modifying the properties of the map components, and using specialized Gentran:Server functions to manipulate the EDI file structure.

However, if you want to use a specialized version of an EDI standard that is not available in the Gentran:Server standards database (including the CII standard), you must define the EDI file yourself.


Regardless of whether the system generates the EDI file or you define it, the specific EDI map components that you use depend on the type of map you are creating. This includes the standard, version, and transaction set (document) selected, and which groups, segments, TFDs (for the CII standard), composites, and elements your company requires.




Before you begin

We recommend that you determine which map components you are using before generating or defining an EDI file.

Map components

This table lists the map components that Gentran:Server uses to visually represent the EDI file format.

Icon	Description
	<p>A group is a looping structure that contains related segments and/or groups that repeat in sequence until either the group data ends or the maximum number of times that the loop is allowed to repeat is exhausted.</p> <p>Groups are defined by the EDI standards. A group that is subordinate to another group is a subgroup (this corresponds to a nested looping structure – a loop within a loop).</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Icon	Description
	A segment contains a group of related elements or composite data elements that combine to communicate useful data. Segments are defined by the EDI standards. A segment can occur once or can repeat multiple times.
	A TFD (Transfer Form Data) is used only in the CII standard. A TFD consists of a data tag (data identifier), data element length, and the variable-length data element TFD.
	<p>A composite is a data element that contains two or more component data elements or subelements. Composites are defined by the EDI standards that use them (EDIFACT, TRADACOMS, and certain ANSI X12 standards).</p> <p>A repeating composite is a related group of EDI subelements that have the ability to loop as a whole (occur more than once) within a particular EDI segment. To allow a composite to repeat multiple times within a segment, each occurrence of the composite must be separated by a special delimiter, known as the repeating element delimiter.</p> <p>Example Your delimiters are set as follows:</p> <p>Segment ~ Element * Tag * Sub Element : Repeating Ele ^</p> <p>Then, if a BGM segment contains a repeating composite that occurs 3 times within the data followed by 1 regular element, and the composite contains 2 subelements, the segment is in the following form:</p> <pre>BGM*SubA-1:SubB-1^SubA-2:SubB-2^SubA-3:SubB-3*Field2~</pre> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Icon	Description
■	<p>An element is the smallest piece of information defined by the EDI standards. An individual element can have different meanings depending on the context. Therefore, elements are normally not considered to have useful meaning until they are combined into segments.</p> <p>An element is the EDI map component that is mapped (linked) to a corresponding application field to move data to and from the EDI file.</p> <p>A repeating element is an EDI elements with the ability to loop (occur more than once) within a particular EDI segment. To allow a single element to repeat multiple times within a segment, each element must be separated by a special delimiter, known as the repeating element delimiter. The use of this delimiter prevents the system from mistaking the repeating elements for normal elements.</p> <p>Example Your delimiters are set as follows:</p> <p>Segment ~ Element * Tag * Sub Element : Repeating Ele ^</p> <p>Then, if a BGM segment contains 3 elements and the second element is a repeating element that occurs 4 times, the segment is in the following form:</p> <pre>BGM*Field1*Field2.1^Field2.2^Field2.3^Field2.4*Field3~</pre>

Reference

See *Using an EDI File* on page 2 - 68 and *Using a CII File* on page 2 - 89 for more information about customizing the properties of the map components and using specialized Gentran:Server functions to manipulate the EDI file structure.

Introducing the Application Integration Subsystem

Overview

Introduction

The Gentran:Server Application Integration subsystem enables you to translate your application files to EDI standard formats for documents you send to your partners (outbound mapping), and to translate EDI standard formats to your application format for documents that you receive from your partners (inbound mapping).

How to Navigate in Application Integration

Introduction

The Gentran:Server Application Integration Window allows you to access its functionality in four different ways:

- Click the menu option from the Main Menu Bar.
- Click the appropriate button on the Main Toolbar.
- Click the appropriate part of the map.
- Click a map component with the right mouse button to access a shortcut menu that contains all the functions that are currently available for that map component. The shortcut menus enable you to quickly and easily access available functionality. The content of the shortcut menus varies, depending on the type and level of the selected map component.

Note

When you start Gentran:Server Application Integration, the Main Menu Bar contains a subset of menu items. The full set of menu items is displayed after you create a new map or open (load) an existing map.

Application Integration Window

This table lists the parts of the Gentran:Server - Application Integration Window.

Part	Function
Main menu bar	Contains drop-down menus. Unavailable items are dimmed. Reference Detailed descriptions of each of the commands on these menus are found in the online Help.
Main Toolbar	Enables you to access some of the most common functions in the Application Integration subsystem. Unavailable items are dimmed. Note The Main Toolbar is a dockable toolbar, so you can affix it to any edge of the client window.
Status bar	Displays status information about a selection, command, or process, defines menu items as you highlight each item in the menu, and indicates any current keyboard-initiated modes for typing.

How to Build a Map with Application Integration

Map Building Process

This table defines how to proceed with building a map using the Gentran:Server Application Integration subsystem.

Stage	Description
1	Prepare and analyze <ul style="list-style-type: none"> ▶ Obtain a layout of your application file and determine how it corresponds with the EDI standard you are using. ▶ Determine how you move data to or from each application field.
2	Set global defaults (one time only) <p>The first time you use the Gentran:Server Application Integration subsystem, you should establish the default date format that the system uses.</p> <p>Reference See <i>How to Set the Default Date Format</i> on page 2 - 18 for more information.</p>
3	Create a new map <p>Reference See <i>How to Create a New Map</i> on page 2 - 20 for more information.</p>
4	Activate the appropriate EDI groups, segments, and elements or auto trim based on a sample EDI file <i>or</i> Define your CII EDI file <p>References</p> <ul style="list-style-type: none"> ▶ See <i>How to Activate EDI Map Components</i> on page 2 - 37 for more information. ▶ See <i>How to Use Auto Trim</i> on page 2 - 39 for more information. ▶ See <i>Using a CII File</i> on page 2 - 89 for more information.
5	Define your application <p>Reference See <i>Using a Fixed-Format File</i> on page 2 - 51 for more information.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Stage	Description
6	<p>Map the appropriate data for each application field</p> <p>References</p> <ul style="list-style-type: none"> ▶ See <i>How to Create Simple Links</i> on page 2 - 114 for more information. ▶ See <i>Using Standard Rules</i>, chapter 3, for more information. ▶ See <i>Using Extended Rules</i>, chapter 6, for more information.
7	<p>Compile the translation object</p> <p>Reference</p> <p>See <i>How to Compile a Map</i> on page 2 - 120 for more information about compiling the translation object and translation object naming conventions.</p>
8	<p>Print the mapping report</p> <p>Validate and review the map, and make modifications as needed.</p> <p>Reference</p> <p>See <i>How to Print a Mapping Report</i> on page 2 - 124 for more information.</p>
9	<p>Register the translation object with Gentran:Server</p> <p>Reference</p> <p>See <i>How to Register a New Translation Object in the Gentran:Server User's Guide</i> for more information.</p>
10	<p>Create the appropriate trading relationship</p> <p>Establish the appropriate trading relationship in Gentran:Server for your trading partners.</p> <p>Reference</p> <p>See <i>Using Partners in the Gentran:Server User's Guide</i> for more information about creating trading relationships.</p>
11	<p>Testing the translation object</p> <p>Obtain test data from your partners and process the data. Verify acknowledgement processing (if applicable). Verify communications with your network.</p>

Designing your Map

Contents	Overview	2 - 3
	▶ Introduction.....	2 - 3
	▶ Preparation and Analysis.....	2 - 4
	Setting Global Defaults	2 - 6
	▶ Customizing Your Display.....	2 - 6
	▶ How to Customize Global Display Options.....	2 - 7
	▶ How to Customize Global Colours.....	2 - 9
	▶ How to Customize Global Fonts.....	2 - 11
	▶ How to Customize Global Display of Links.....	2 - 13
	▶ How to Customize the Auto-increment Map Version.....	2 - 15
	▶ How to Make the Two Sides of a Map Equal.....	2 - 17
	▶ How to Set the Default Date Format.....	2 - 18
	Creating a Map and Defining Details	2 - 20
	▶ How to Create a New Map.....	2 - 20
	▶ How to Define Translation Object Details.....	2 - 28
	▶ How to Load a File Definition.....	2 - 35
	▶ How to Save a File Definition.....	2 - 37
	▶ How to Activate EDI Map Components.....	2 - 38
	▶ How to Use Auto Trim.....	2 - 40
	▶ How to Promote Groups and Repeating Segments.....	2 - 42
	▶ How to Split Groups and Repeating Segments.....	2 - 45
	▶ How to Use Copy, Cut, and Paste.....	2 - 49
	Using a Fixed-Format File	2 - 52
	▶ Overview.....	2 - 52
	▶ How to Adjust File Format Properties.....	2 - 53
	▶ How to Create the First Record.....	2 - 55
	▶ How to Create Subsequent Records.....	2 - 57
	▶ How to Create a Temporary Record.....	2 - 59
	▶ How to Create a Group.....	2 - 63

Overview

Introduction

What's in this chapter

This chapter explains some of the basic mapping functions. Some of the mapping functions are performed initially (creating a map), some are performed as you define the map, and some are performed after the map is completed (compiling and printing a mapping report).

Preparation and Analysis

Introduction

The first step in creating a map is the analysis of the mapping requirements. This is the most important step in creating a successful map. If the analysis you perform is complete, you have all the information you need to create the map in an efficient and logical manner. If you omit this critical step and proceed directly to creating the map, it is likely that creating the map is a much longer and arduous task, and the map may be invalid due to oversights and omissions.

Regardless of whether you are translating data inbound or outbound, this table lists the stages of mapping analysis.

Stage	Description
1	Analyzing your application file format
2	Analyzing your partners EDI file format
3	Correlating your application and the EDI file formats

After you complete these steps, you can begin creating the map.

Analyzing your application file format

The first step of mapping analysis is analyzing your application file format, because this is probably the component that is the most familiar to you. Your application file format contains all the information that you need to extract from the purchase order your partner sends you, so that your system can correctly process the purchase order, and your company can fill the order correctly.

You must define your application file format to Gentran:Server. If your company has an existing application file format, you should obtain the record layout from the appropriate person. If you did not have an existing application file format, you need to create one by determining which fields are necessary to process the data correctly, and then grouping the fields logically under records.

Analyzing the EDI file

To analyze the EDI file, you must first determine what your trading partners are sending you. You and your partners need to agree on which standard, version, and transaction set you are using. It is important to know the information that your partners are sending you within the purchase order, and what data is contained in each element used.

It is very helpful to review the EDI standards that you are using before analyzing the EDI file. Obtain an EDI standards manual for each standard and version that you are using. Standards manuals are available from EDI standard agencies.

After discussing with your partners what they are sending, determining which segments and elements your company requires, and reviewing the EDI standard, you can list the map components that you need to make available for use (activate).

Reconciling your application file and the EDI file

To reconcile your application file format with the EDI file, you must identify each application file with its corresponding element in the EDI file and select a method for mapping it. To map information to a field, you use linking, standard rules, extended rules, or a combination of all three.

Linking

Linking (simple mapping) enables you to map a field or element from the input side of the map to a field or element on the output side of the map. The link between two map components (fields) is visually represented with a line connecting the two fields.

Standard rules

Standard rules give you access to mapping operation functions that are more complex than simple linking, but less involved than extended rules.

Extended rules

Extended rules enable you to use a Gentran:Server proprietary programming language to perform virtually any mapping operation you require.

Additionally, you may determine that you need to establish temporary storage (work) areas for the map to handle items such as Ship To and Bill To name and address information, which is extracted using extended rules from a group in the EDI data.

Recommendation

Determine which mapping operations are required on a field-by-field basis for your application file.

After your analysis is complete

After you finish correlating your application file format and EDI file on paper, you can begin creating the map.

Setting Global Defaults

Customizing Your Display

Introduction

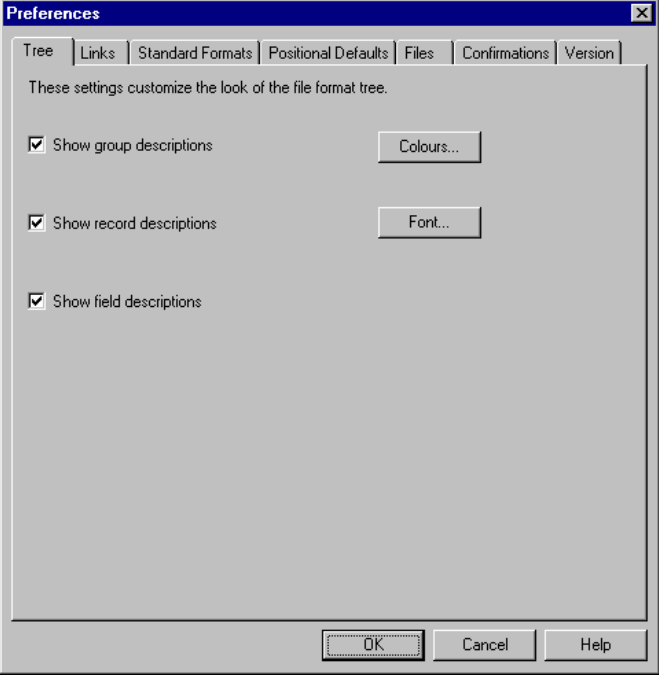
The Gentrans:Server Application Integration subsystem allows you to customize the display of maps in several different ways:

- Customizing global display options
- Customizing colours (global option)
- Customizing fonts (global option)
- Displaying links (global option)
- Customizing the auto-increment map version
- Making the two sides of a map equal

The Preferences dialog box enables you to set global defaults for Gentrans:Server. The map display options can be set or changed at any time.

How to Customize Global Display Options

Procedure Use this procedure to customize global (for all maps) display options.

Step	Action
1	<p>From the Options menu, select Preferences.</p> <p>System response The system displays the Preferences dialog box (Tree tab displayed by default).</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>

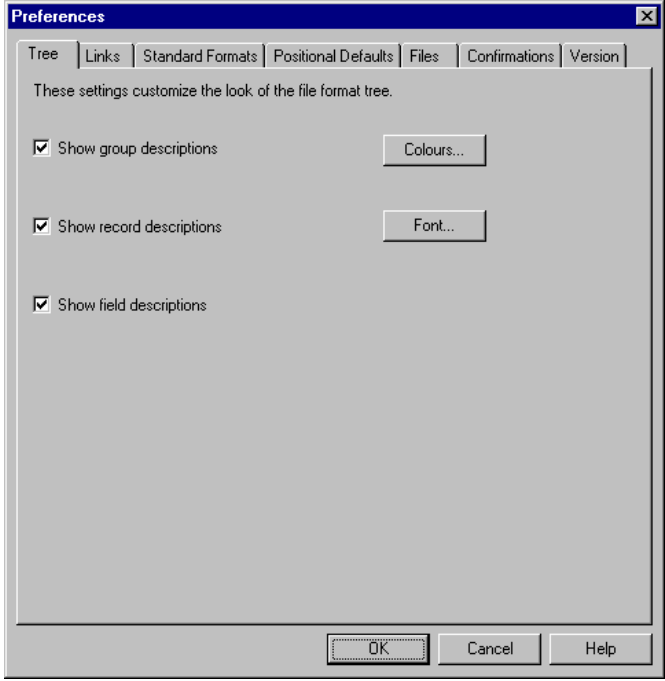


(Contd) Step	Action
2	<p>Do you want to turn on the default display of group, record (segment), and field (element) descriptions?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, click the check box in front of each in the List section of the dialog box (so the system displays a checkmark in the check box). <p>To turn off the default display of group, record (segment), and field (element) descriptions, click the appropriate check box again to remove the checkmark.</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with the next step. <p>Notes Typically, you want to have all the descriptions displayed for reference. However, for a variety of reasons, you may not want the descriptions displayed. Depending on the size of your monitor, it may be easier to view the entire map if the descriptions are not displayed.</p> <p>You may want to experiment with shrinking the size of the font for the map. See <i>How to Customize Global Fonts</i> on page 2 - 11 before you turn off the display of descriptions.</p>
3	<p>Do you want to change the default display colors?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, see <i>How to Customize Global Colours</i> on page 2 - 9 for more information. ▶ If <i>no</i>, continue with the next step.
4	<p>Do you want to change the default display font?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, see <i>How to Customize Global Fonts</i> on page 2 - 11 for more information. ▶ If <i>no</i>, continue with the next step.
5	Click OK to save changes and exit the Preferences dialog box.

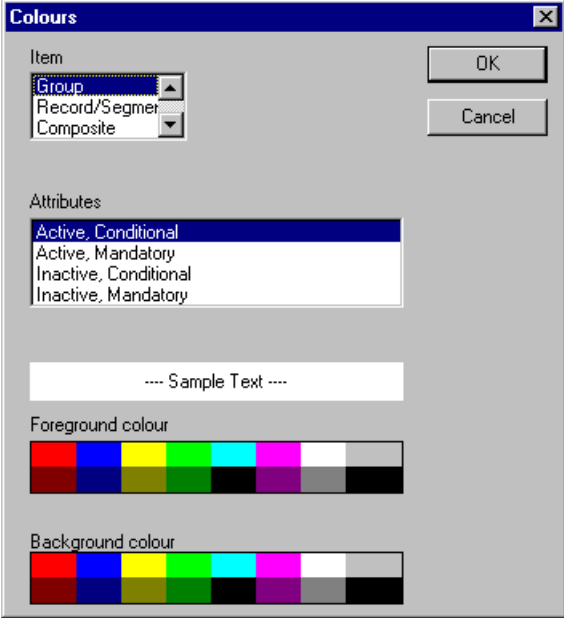
How to Customize Global Colours

Introduction The Colours feature enables you to select foreground and background colours to visually define the various map components. The use of colour is optional.

Procedure Use this procedure to customize colours for all maps.

Step	Action
1	<p>From the Options menu, select Preferences.</p> <p>System response The system displays the Preferences dialog box (Tree tab displayed by default).</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>



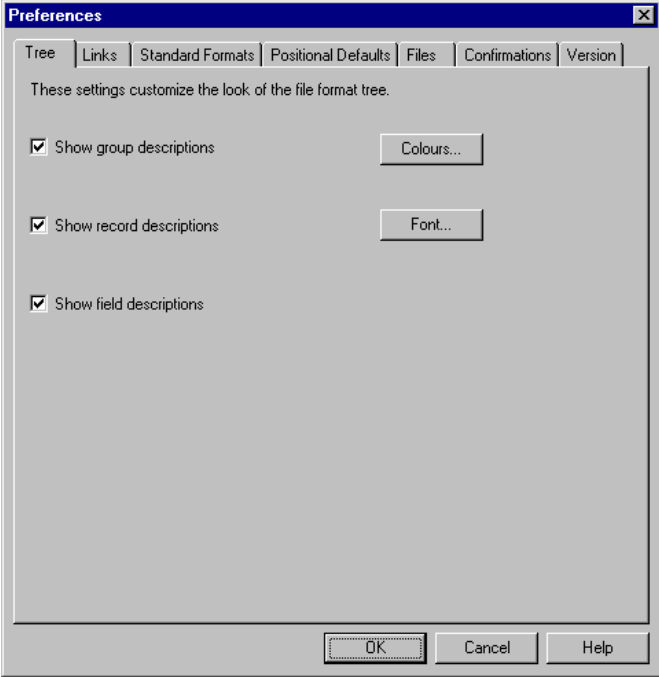
(Contd) Step	Action
2	<p>Click Colours.</p> <p>System response The system displays the Colours dialog box.</p> 
3	From the Item list, select the type of map component (Group, Record/Segment, Composite, Field/Element).
4	<p>Further define the type of map component by selecting whether it is active or inactive and conditional or mandatory from the Attributes list.</p> <p>The valid selections are:</p> <ul style="list-style-type: none"> ▶ Active, Conditional ▶ Active, Mandatory ▶ Inactive, Conditional ▶ Inactive, Mandatory
5	Select the foreground and background colors.
6	Click OK to globally define the selected colours for all maps.

How to Customize Global Fonts

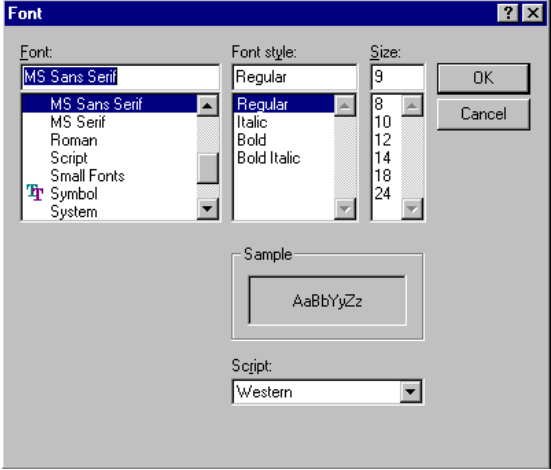
Introduction The Font feature enables you to globally change the font that is used in the display of all maps. The default font that Gentran:Server uses is MS Sans Serif 9 point.

The Font feature enables you to change the font type, style, and point size. This gives you the flexibility to shrink the font if you need to view more of the map on your monitor, enlarge the font, or change the type and style to be more easily readable by you.

Procedure Use this procedure to customize the display font for all maps.

Step	Action
1	<p>From the Options menu, select Preferences.</p> <p>System response The system displays the Preferences dialog box (Tree tab displayed by default).</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>




(Contd) Step	Action
2	<p>Click Font.</p> <p>System response The system displays the Font dialog box.</p> 
3	<p>From the Font box, select the type of font.</p> <p>Note The default is MS Sans Serif.</p>
4	<p>From the Font Style box, select the style.</p> <p>Note The default is Regular.</p>
5	<p>From the Size box, select the point size.</p> <p>Note The default is 9.</p>
6	<p>Click OK to make the global font change to all maps and exit the Font dialog box.</p>

How to Customize Global Display of Links

Introduction Mapping Links are the visual lines that connect each field/element/TFD on the Input side of the map that is mapped to a field/element/TFD on the Output side of the map.

Procedure Use this procedure to customize the global display of mapping links.

Step	Action
1	From the Options menu, select Preferences . System response The system displays the Preferences dialog box (Tree tab displayed by default).
2	Select the Links tab. System response The system displays the linking options.  <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
3	<p>Select the linking option that you want to set as the default for all maps.</p> <ul style="list-style-type: none">▶ Show no links Do not visually display mapping links.▶ Show links to or from the currently selected element Display only the mapping links for the currently selected field (this option enables you to concentrate on the selected field and removes the confusion of viewing many links at once).▶ Show links to or from all visible elements Display all the mapping links.
4	<p>Click the OK button to save changes and exit the Preferences dialog box.</p>

How to Customize the Auto-increment Map Version

Introduction

Gentran:Server supports automatically incrementing the version number of a map based on a global option you can set. Previously, the version number of a map could only be set or incremented manually, on the Application Integration Translation Object Details dialog box.

You can set a global option on the Application Integration Preferences dialog box to specify when map version numbers are incremented. This list describes the available options for automatically incrementing the map version number.

- Never auto-increment.
- Ask whether to increment when saving a map.
- Ask whether to auto-increment when compiling a map.
- Always auto-increment when saving a map.
- Always auto-increment when compiling a map.

Note

The auto-increment function updates the minor version number of the map on the Translation Object Details dialog box, up to 255. If the minor version number exceeds 255, the system updates the minor version number to zero and increases the major version number.

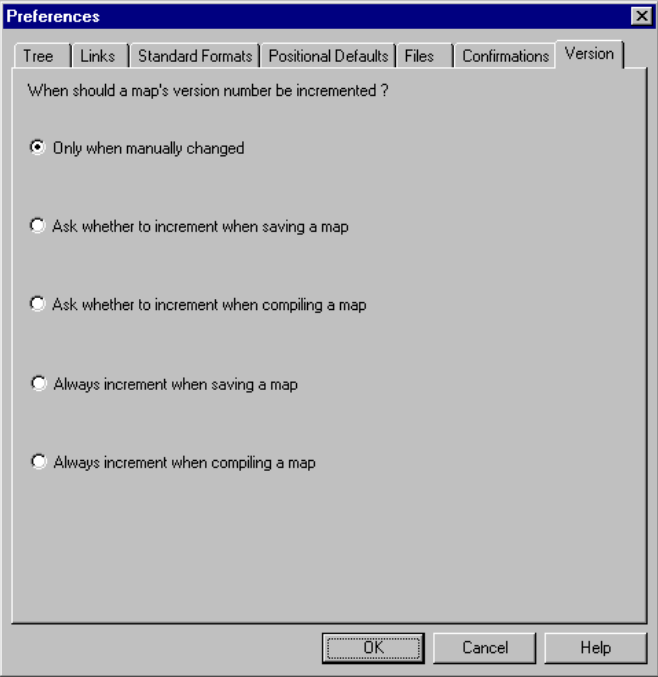
Example

Version number 1.255 will be auto-incremented to 2.0.

Setting auto-increment

Use this procedure to set the auto-increment map version function.

Step	Action
1	<p>From the Options menu, select Preferences.</p> <p>System response The system displays the Preferences dialog box (Tree tab displayed by default).</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
2	<p>Select the Version tab.</p> <p>System response The system displays the auto-increment version options.</p> 
3	<p>Select the appropriate option and click OK.</p> <p>System response The system sets the global auto-increment option you selected.</p>

How to Make the Two Sides of a Map Equal

Introduction

When you open a map, the Input and Output sides of a map are displayed in equal dimensions. After manipulating a map and moving the center bar that divides the display of the Input and Output sides, you may wish to again make the two sides equal.

The Equalize function enables you to reinstate the two sides of the map with focus in equal dimensions. The use of the Equalize function is optional.

Procedure

Use this procedure to make the two sides of a map equal.

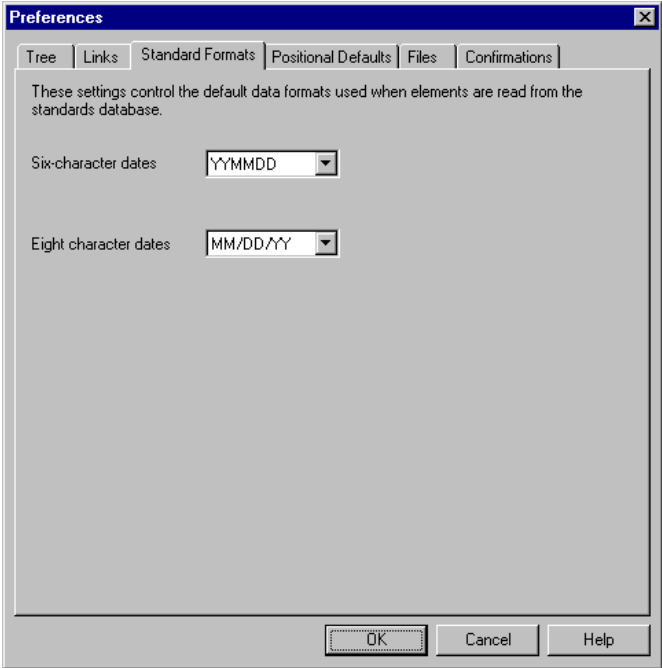
Step	Action
1	From the View menu, select Equalize . Note This moves the center dividing bar between the two sides of the map with focus, so that the two sides of the map are equally proportioned.

How to Set the Default Date Format

Introduction You typically establish the default date format for all date fields *one time only*. This default is used when EDI documents are initially loaded from the standard. This default can be overridden on the Field Properties dialog box.

This global option changes the default date format for all maps. However, the format of the existing date fields do not change; the default is only used for new maps.

Procedure Use this procedure to establish the default date format.

Step	Action
1	From the Options menu, select Preferences . System response The system displays the Preferences dialog box (Tree tab displayed by default).
2	Select the Standard Formats tab. System response The system displays the date format options. 
3	From the Six-character dates and Eight character dates lists, select the appropriate six-character and eight-character default date formats. <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
4	<p>Click OK to accept the default date formats for the six- and eight-character date boxes and exit the Preferences dialog box.</p> <p>Tip To change the order in which the date formats appear in the Six-character dates and Eight character dates lists or to add a new date format to the lists, select Date Formats from the Options menu.</p>

Creating a Map and Defining Details

How to Create a New Map

Introduction You can create four different types of maps: System Import (containing the header information for an Import map), Import (for Outbound Maps), Export (for Inbound Maps), or Turnaround (for EDI to EDI maps). The initial procedures for creating a new map of each of the four types is the same.

Procedure Use this procedure to create a new map.

Step	Action														
1	From the File menu, select New . System response The system displays the New Map Wizard.														
2	Answer the following questions and then click Next . <ul style="list-style-type: none"> What kind of map are you creating? The following table defines the selections: <table border="1"> <thead> <tr> <th>Part</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>Import</td> <td>Used for outbound maps.</td> </tr> <tr> <td>Export</td> <td>Used for inbound maps.</td> </tr> <tr> <td>Interchange break</td> <td>Used in advanced mapping to separate interchanges.</td> </tr> <tr> <td>Functional Acknowledgement Inbound</td> <td>Used in advanced mapping to reconcile functional acknowledgements.</td> </tr> <tr> <td>Functional Acknowledgement Outbound</td> <td>Used in advanced mapping to generate functional acknowledgements.</td> </tr> <tr> <td>System Import Header</td> <td>Used to determine which trading relationship (established in Partner Editor) corresponds to each document in the application file, so the system knows which import translation object to use to process the document.</td> </tr> </tbody> </table>	Part	Function	Import	Used for outbound maps.	Export	Used for inbound maps.	Interchange break	Used in advanced mapping to separate interchanges.	Functional Acknowledgement Inbound	Used in advanced mapping to reconcile functional acknowledgements.	Functional Acknowledgement Outbound	Used in advanced mapping to generate functional acknowledgements.	System Import Header	Used to determine which trading relationship (established in Partner Editor) corresponds to each document in the application file, so the system knows which import translation object to use to process the document.
Part	Function														
Import	Used for outbound maps.														
Export	Used for inbound maps.														
Interchange break	Used in advanced mapping to separate interchanges.														
Functional Acknowledgement Inbound	Used in advanced mapping to reconcile functional acknowledgements.														
Functional Acknowledgement Outbound	Used in advanced mapping to generate functional acknowledgements.														
System Import Header	Used to determine which trading relationship (established in Partner Editor) corresponds to each document in the application file, so the system knows which import translation object to use to process the document.														
(Continued on next page)															

(Contd) Step	Action	
2 (contd)	Turnaround	Used for EDI to EDI maps.
	Transaction build	Used in advanced mapping to build transaction envelopes.
	Transaction break	Used in advanced mapping to separate documents.
	Functional group build	Used in advanced mapping to build functional group envelopes.
	Functional group break	Used in advanced mapping to separate functional groups.
	Interchange build	Used in advanced mapping to build interchange envelopes.
	Direct Map	<p>Warning This map function is not used by any Gentran:Server for Windows component or by any Sterling Commerce product that may connect to Gentran:Server for Windows. This function will be removed in a future release.</p> <p>Recommendation Any existing maps that are configured with the Direct Map type should be switched to another appropriate type.</p>
<ul style="list-style-type: none"> ▶ <i>What is the name of the map?</i> Type the unique name of the map. The system adds the .MAP extension. ▶ <i>What is your name?</i> Type your name if it differs from the user name prompted by the system. <p>System response The system displays the New Map Wizard - Input Format dialog box.</p> <p>Note You need to complete the format of the Input side of the map. This is the format of the data that is translated by the Gentran:Server system.</p> <p style="text-align: right;">(Continued on next page)</p>		



(Contd) Step	Action
3	<p>For the input side of the map: do you want to create a new data format using a syntax that you define?</p> <ul style="list-style-type: none"> <p>▶ If <i>yes</i>, select one of the following input format options and continue with step 4:</p> <p>Delimited EDI (Electronic Data Interchange file)</p> <p>Positional (VDA, GENCOD, application files, etc.)</p> <p>CII (Japanese standard)</p> <p>CII Positional (for CII Build/Break maps)</p> <p>Note For Inbound maps (Export), the Input Format Type is “CII” or “EDI”. For Outbound maps (System Import or Import), the Input Format Type is “Positional.” For Turnaround maps, the Input Format Type is “EDI” or “CII.”</p> <p>▶ If <i>no</i> (you want to load the data format from a saved definition), select the Load the data format from a saved definition option, and type the path and file name of the saved definition (or click Browse to display the Open File Definition dialog box). Continue with step 6.</p>
4	<p>Did you select “Delimited EDI” or “CII” and want to customize the format?</p> <ul style="list-style-type: none"> <p>▶ If <i>yes</i>, click Customize and continue with the next step.</p> <p>System response The system displays the New Delimited EDI Wizard or New CII Wizard dialog box.</p> <p>Note The Delimited EDI and CII wizards enable you to create your format from the standards database.</p> <p>▶ If <i>no</i>, click Next and continue with step 7.</p> <p>System response The system displays the New Map Wizard - Output Format dialog box.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action	
5	Follow the steps for the appropriate dialog box and then continue with step 7.	
	IF the dialog box displayed is...	THEN follow this procedure...
	New Delimited EDI Wizard	<ul style="list-style-type: none"> a. Click Next. b. Select the ODBC data source that contains the standards database. c. Select the standards agency, version, transaction set, and release (for TRADACOMS only) and click Next. d. Click Finish.
	New CII Wizard	<ul style="list-style-type: none"> a. Click Next. b. Select the ODBC data source that contains the standards database. c. If you want to use Japanese descriptions, select the check box (the default is English descriptions). d. Click Next. e. Select the standards agency, version, transaction set, and release (for TRADACOMS only) and click Next. f. Select the appropriate Multi-detail header and click Next. g. Click Finish.
	<p>System response The system displays the New Map Wizard - Output Format dialog box.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>	



(Contd) Step	Action
6	<p>Did you choose to load the data format from a saved definition and click Browse to display the Open File Definition dialog box?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, type the file name and click Open to load the selected file format definition, and then continue with step 7. <p>Note You can now select either a .DDF or .IFD file.</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with step 7. <p>Note If the DDF is invalid, the system displays a message box explaining the problem and terminates the import.</p>
7	<p>For the output side of the map: do you want to create a new data format using a syntax that you define?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select one of the following output format options and continue with step 8: Delimited EDI (Electronic Data Interchange file) Positional (VDA, GENCOD, application files, etc.) CII (Japanese standard) CII Positional (for CII Build/Break maps) <p>Note For Inbound maps (Export), the Output Format Type is “Positional.” For Outbound maps (System Import or Import), the Output Format Type is “CII” or “EDI.” For Turnaround maps, the Output Format Type is “EDI” or “CII.”</p> <ul style="list-style-type: none"> ▶ If <i>no</i> (you want to load the data format from a saved definition), select the Load the data format from a saved definition option and type the path and file name of the saved definition (or click Browse to display the Open File Definition dialog box). Continue with step 9. <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action	
8	<p>Did you select “Delimited EDI” or “CII” and want to customize the format?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, click Customize and continue with the next step. <p>System response The system displays the New Delimited EDI Wizard or New CII Wizard dialog box.</p> <p>Note The Delimited EDI and CII wizards enable you to create your format from the standards database.</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, click Finish to load the standards information you selected and create the new map (this may take a few seconds). <p>System response The system displays the new map in the Application Integration Window.</p>	
9	Follow the steps for the appropriate dialog box and then continue with step 11.	
	IF the dialog box displayed is...	THEN follow this procedure...
	New Delimited EDI Wizard	<ul style="list-style-type: none"> a. Click Next. b. Select the ODBC data source that contains the standards database. c. Select the standards agency, version, transaction set, and release (for TRADACOMS only) and click Next. d. Click Finish.
(Continued on next page)		



(Contd) Step	Action	
9 (contd)	New CII Wizard	a. Click Next . b. Select the ODBC data source that contains the standards database. c. If you want to use Japanese descriptions, select the check box (the default is English descriptions). d. Click Next . e. Select the standards agency, version, transaction set, and release (for TRADACOMS only) and click Next . f. Select the appropriate Multi-detail header and click Next . g. Click Finish .
System response The system displays the New Map Wizard dialog box.		
10	Did you choose to load the data format from a saved definition and click Browse to display the Open File Definition dialog box? <ul style="list-style-type: none"> ▶ If <i>yes</i>, type the file name and click Open to load the selected file format definition, and then continue with step 11. Note You can now select either a .DDF or .IFD file. <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with step 11. 	
11	Click Finish to load the standards information you selected and create the new map (this may take a few seconds). System response The system displays the new map in the Application Integration Window. Note After you finish creating and saving a new map, you need to define the Input and Output sides of the map. The steps you take are different, depending on whether the map is an Import, System Import, Export, or Turnaround map.	

How to Define Translation Object Details

Introduction

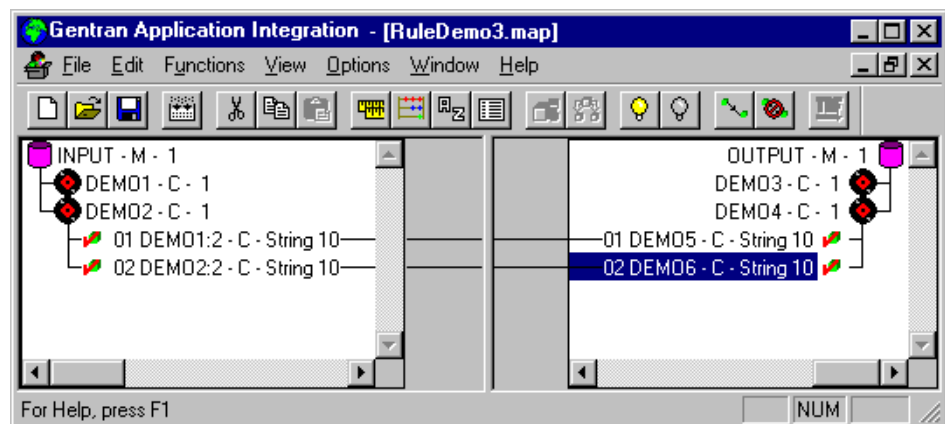
The Translation Object Details dialog box enables you to edit the details of the translation object, including the description and version information. This dialog box enables you to instruct the translator to use the pad character and alignment settings of each string field when reading a positional or CII positional file, to determine how to trim pad characters from the string data.

Rule execution prior to version 3.0

The Translation Object Details dialog box also enables you to specify whether you want the translator to execute standard and extended rules as it did for Gentran:Server prior to version 3.0. The standard behavior for Gentran:Server prior to version 3.0 is as follows: if an output record/segment is linked to an input field in a record/segment that does not contain data but is at the same hierarchical level as an input record/segment that does contain data, the standard and extended rules on the output field will be executed even though there is no data present.

Example

In the example diagram below, the extended rule on the DEMO5 field on the output side of the map is executed even though there is no data for the field to which it is linked (DEMO1:2 on the input side of the map). The rule is executed because there is data present for a field in the DEMO1 record on the input side of the map, and that record is at the same level as DEMO2.



Note

For any map that was compiled with Gentran:Server prior to version 3.0, the system emulates the previous behavior by automatically selecting the “Gentran:Server for Windows 2.x Compatible Rule Execution” check box on the *Translation Object Details dialog box* on page 2 - 29. You can modify this default if you wish, but you would then need to re-map any instances where you want standard or extended rules to run on an output record/segment that does not contain fields that will be populated with data.

Rule execution in version 3.0 and above

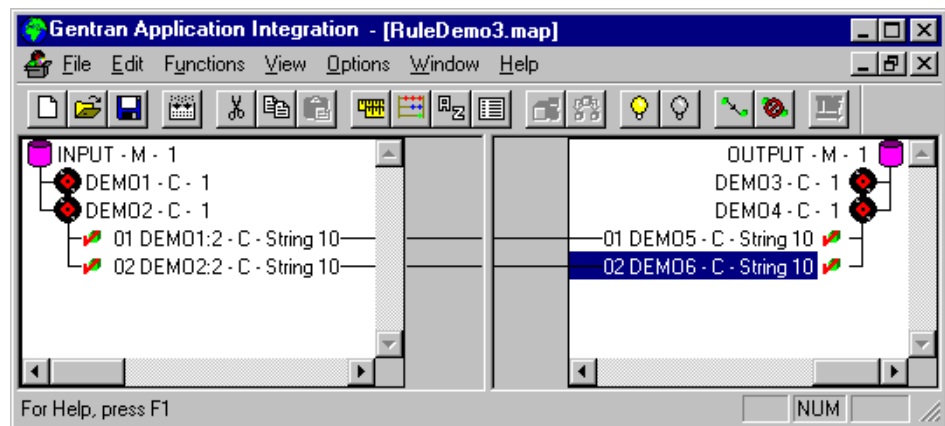
In version 3.0, the standard rule execution behavior is as follows: standard and extended rules for a record/segment on the output side of the map are only executed if any of its links populate the output map component with data.

Note

For any map created with Gentran:Server version 3.x or greater, the system default is to clear the “Gentran:Server for Windows 2.x Compatible Rule Execution” check box on the Translation Object Details dialog box.

Example

In the example diagram below, the extended rule on the DEMO5 field on the output side of the map is *not* executed because there is no data for the field to which it is linked (DEMO1:2 on the input side of the map). The rule is *not* executed even though there is data present for a field in the DEMO1 record on the input side of the map, and that record is at the same level as DEMO2.

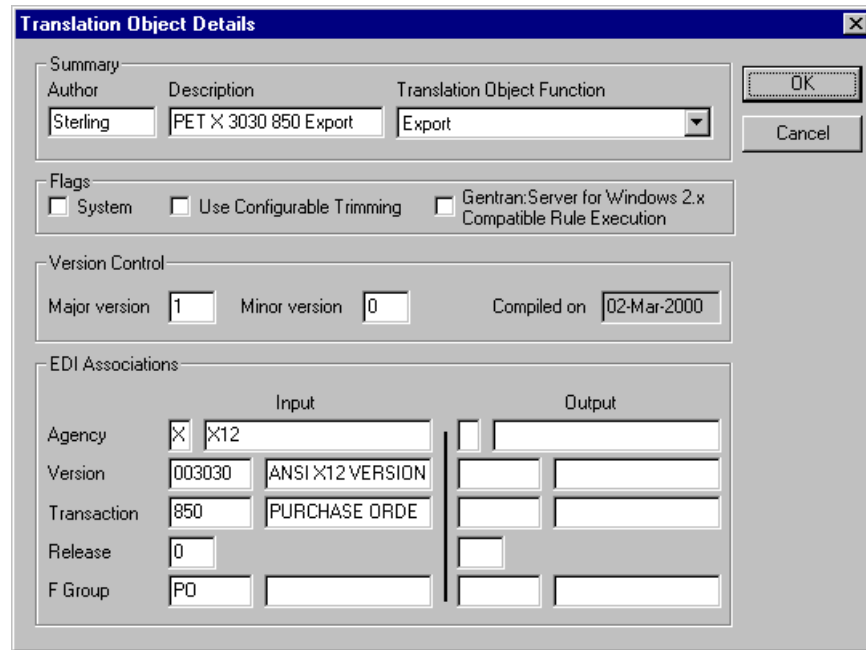


Recommendation

We recommend that you do not select this check box for version 3.x and greater, as this differs from the prescribed behavior.

Translation Object Details dialog box

This diagram illustrates the Translation Object Details dialog box.



Parts and functions

The following table lists the parts of the Translation Object Details dialog box and their functions.

Part	Function
Author	Identifies the person or department that created the map. Note This box is mandatory.
Description	Contains a unique description of the map. Note This box is mandatory. This description is used by the system to identify the map. We strongly recommend that you use the following identifying characteristics of the map: the partner that this map is used for, the standard, the version, the type of transaction this map uses, and the direction of the map (i.e., Import, Export, etc.). Example “MWT X 3030 850 Export” is the description of a map used with partner MWT, for an ANSI X.12 version 003030 Purchase Order (850), that is sent Inbound (Export). (Continued on next page)



Part	Function
Translation Object Function	<p>Verify the type of map from the list. Valid values are:</p> <ul style="list-style-type: none"> ▶ Import - choose this type for outbound maps (application to EDI) ▶ Export - choose this type for inbound maps (EDI to application) ▶ Turnaround - choose this type for EDI to EDI maps ▶ Transaction build (this type used in advanced mapping to build transaction envelopes) ▶ Transaction break (this type used in advanced mapping to separate documents) ▶ Functional group build (this type used in advanced mapping to build functional group envelopes) ▶ Functional group break (this type used in advanced mapping to separate functional groups) ▶ Interchange build (this type used in advanced mapping to build interchange envelopes) ▶ Interchange break (this type used in advanced mapping to separate interchanges) ▶ F/A Inbound (this type used in advanced mapping to reconcile functional acknowledgements) ▶ F/A Outbound (this type used in advanced mapping to generate functional acknowledgements) <p>Note This box is mandatory.</p>
System	<p>Designates this map as a system translation object (one that will be used internally by the system).</p> <p>Warning You should only select this check box if you are certain that you want to define this translation object as a system translation object. Once a system translation object is registered with Gentran:Server, it cannot be deleted.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

Part	Function
Use Configurable Trimming	<p>Instructs the translator to use the pad character and alignment settings of each string field when reading a positional or CII positional file, to determine how to trim pad characters from the string data.</p> <p>Example If the alignment setting indicates that the data is aligned in the left of the field, then pad characters are trimmed from the right of the field. If the alignment setting indicates that the data is aligned at the right of the field, then pad characters are trimmed from the left of the field. Finally, if the alignment setting indicates that the field data is aligned in the center of the field, then pad characters are trimmed from both the left and right of the field.</p> <p>If this option is cleared (which is the default) pad characters are always trimmed from both the left and right of each positional string field, which is the current system behavior.</p> <p>Example Use configurable trimming to preserve either trailing or leading spaces in your application data.</p>
Gentran:Server for Windows 2.x Compatible Rule Execution	<p>Enables you to specify how the translator executes standard and extended rules on the output side of a map.</p> <p>Reference See <i>Rule execution prior to version 3.0</i> on page 2 - 27 and <i>Rule execution in version 3.0 and above</i> on page 2 - 28 for more information.</p>
Major version/ Minor version	<p>Enables you to designate different versions of a translation object.</p> <p>Note The valid values for each box is “0” - “255”. These two boxes are available for your use only. They are not used by the system.</p> <p>Example If you type “5” in the Major box and “45” in the Minor box, the Translation Object Version Number is “5.45”.</p>
Compiled on	<p>Displays the date when the translation object was compiled.</p> <p>Note It will be blank if the translation object has not been compiled yet.</p> <p style="text-align: right;">(Continued on next page)</p>



Part	Function
EDI Associations section	If you selected “Delimited EDI” for the input and/or output sides of the map, this section contains the EDI information. Warning You should change these boxes only if you want to change the EDI agency, version, transaction, release, or functional group of an existing map. This enables you to copy and alter an existing map.
OK	Saves changes and exits the dialog box.
Cancel	Exits the dialog box without saving changes.

Procedure Use this procedure to specify translation object details.

Step	Action	
1	From the Edit menu, select Details . System response The system displays the <i>Translation Object Details dialog box</i> on page 2 - 29.	
2	Do you want to change the map description? <ul style="list-style-type: none"> ▶ If <i>yes</i>, type the new description in the Description box and continue with the next step. ▶ If <i>no</i>, continue with the next step. 	
3	Do you want to change the compatible rule execution specification? <ul style="list-style-type: none"> ▶ If <i>yes</i>, use the following table to determine your action and continue with the next step. 	
	IF you want the system to execute rules based on this behavior...	THEN the Gentran:Server for Windows 2.x Compatible Rule Execution check box should be...
	prior to version 3.0,	selected.
	version 3.0 and above,	cleared.
	<ul style="list-style-type: none"> ▶ If <i>no</i>, continue with the next step. 	
(Continued on next page)		

(Contd) Step	Action
4	Do you want to change the map version? <ul style="list-style-type: none">▶ If <i>yes</i>, type the appropriate version numbers in the Major and Minor boxes and continue with the next step.▶ If <i>no</i>, continue with the next step.
5	Click OK . System response The system saves your changes and exits the Translation Object Details dialog box.

How to Load a File Definition

Introduction Gentran:Server enables you to load an individual file format definition that you previously saved. This feature provides you with a quick way to build either side of your map.

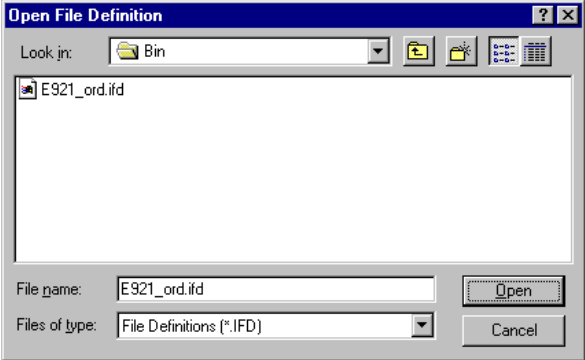
Reference

See *How to Save a File Definition* on page 2 - 36 for more information.

Note

Loading a file definition replaces the selected side of the map. Please be certain that is your intent before performing this task.

Procedure Use this procedure to load a file format definition.

Step	Action
1	<p>Right-click the File Format icon (either the INPUT or OUTPUT side of the map) and select Open File Definition from the shortcut menu.</p> <p>Note If you already used Gentran:Server to create that side of the map, you are prompted with a message that warns you that the existing file format will be replaced. Click Yes to continue.</p> <p>System response The system displays the Open File Definition dialog box.</p> 
2	<p>Does the Look in list reflect the drive\folder where Gentran:Server is installed (the default is GENSRVNT)?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, continue with step 3. ▶ If <i>no</i>, select the appropriate folder and continue with step 3.
3	<p>From the Files of type list, select the type of file that you want to display.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
4	From the list, select the file definition that you want to load into the system or type the file name in the Filename box.
5	Click Open . System response The system loads the selected file format definition.

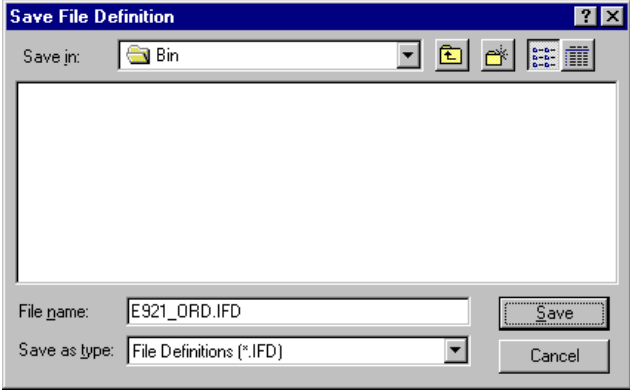
How to Save a File Definition

Introduction Gentran:Server enables you to save an individual file format definition so that you can use it as a guide in future maps. This provides you with a quick way to build either side of your map.

Reference

See *How to Load a File Definition* on page 2 - 34 for more information.

Procedure Use this procedure to save an individual file format definition.

Step	Action
1	<p>Right-click the File Format icon (either the INPUT or OUTPUT side of the map) and select Save File Definition from the shortcut menu.</p> <p>System response The system displays the Save File Definition dialog box.</p> 
2	<p>Does the Save in list reflect the drive\folder where Gentran:Server is installed (the default is GENSRVNT)?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, continue with step 3. ▶ If <i>no</i>, select the appropriate folder and continue with step 3.
3	<p>Type the file name in the Filename box (one to eight characters) or from the list, select the file definition that you want to save.</p>
4	<p>Click Save.</p> <p>System response The system saves the file format definition.</p>

How to Activate EDI Map Components

Introduction

When Gentran:Server generates the EDI sides of the map, the system includes all the groups, segments, composites, and elements that are defined by the standard agency for the version of the document you selected. The system activates all the groups, segments, composites, and elements that are defined as “mandatory” (must be present) by the standard. The system does not enable you to deactivate the mandatory groups, segments, composites, and elements.

When translating data, the system does not process groups, segments, composites, and elements (or records and fields) that are not activated. Therefore, you must activate the groups, segments, composites, and elements that are not defined as mandatory by the standard, but that you have determined that you need to use in mapping.

Tip

As an alternative to activation, you can use the Auto Trim feature if you have a sample EDI file defined.

Reference

See *How to Use Auto Trim* on page 2 - 39 for more information.

Procedure

Use this procedure to activate groups, segments, composites, and elements.

Step	Action
1	From the Functions menu, select Activate .
2	Open (double-click) each group that contains segments or groups that need to be activated. Note As an alternative to opening each map component (steps 2, 5, and 7), you can select Expand All from the View menu to open every map component. You can also select a map component and select Expand Branch from the View menu to open that map component.
3	Select each inactive group that you need to use. This activates the groups. Note If you accidentally click a group, segment, composite, or element that you did not mean to activate, right-click the map component and select Deactivate from the shortcut menu.
4	Select each inactive segment that you need to use. Note This activates the segments.

(Continued on next page)

(Contd) Step	Action
5	Open (double-click) each segment that contains composites or elements that need to be activated.
6	Select each inactive composite that you need to use. Note This activates the composites.
7	Open (double-click) each composite that contains elements that need to be activated.
8	Select each inactive element that you need to use. Note This activates the elements.
9	Once you have activated all the necessary groups, segments, and elements, select Activate from the Functions menu to turn activation mode off.

How to Use Auto Trim

Introduction

Instead of activating map components manually, you can use the Auto Trim feature to modify the EDI side of the map according to a sample EDI file that you select (you must have previously created this EDI file).

The Auto Trim feature examines the EDI file that you specify and then activates and deactivates map components on the EDI side of the map, so that the two files match. The sample EDI file must be the same standard, version, and transaction set (message) as the map for Auto Trim to match map components.

Caveats

To use Auto Trim, you must define the Segment and Element Delimiters. If you do not supply the Tag Delimiter, the system substitutes the Segment Delimiter. If you want to use composite elements, you must specify the Sub Element Delimiter. If you want to use release characters, you must specify the Release Character.

Reference

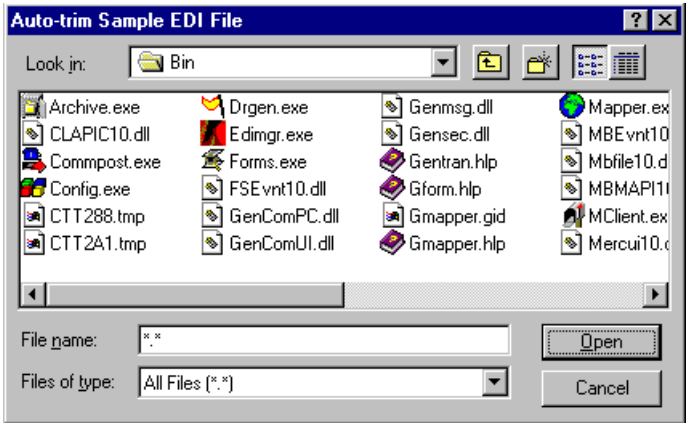
See *How to Verify EDI Delimiters* on page 2 - 69 for more information.

Warning

The sample EDI file must be compliant and must not contain envelope segments.

Procedure

Use the following procedure to use Auto Trim:

Step	Action
1	<ul style="list-style-type: none"> ▶ Right-click the EDI File icon on the EDI side of the map and ▶ select Auto Trim from the shortcut menu. <p>System response The system displays the Auto-trim Sample EDI File dialog box.</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
2	From the Look in list, select the drive\folder where the EDI file is stored. Note The default is GENSRVNT\BIN.
3	If necessary, select the type of file that you want to display from the Files of type list.
4	From the list, select the sample EDI file that you want to load into the system.
5	Click Open to begin the auto trim process. Note When auto trim is complete, the system displays a message stating whether or not auto trim was successful. Click OK to acknowledge the message.

How to Promote Groups and Repeating Segments

Introduction

The Promote function extracts one iteration (instance) of a group or repeating segment. This enables you to map unique data from your application file, and/or enter a specialized definition. Gentran:Server specifies that only one-to-one (no loops) or many-to-many (loop) mapping relationships are valid.

Note

The Promote function is only available if a group or repeating segment is selected.

Promote vs. copy vs. split

You can use the Copy and Paste functions (and change the number in the maximum usage box) to accomplish the same task. However, Promote is a specialized function that guarantees the integrity of your EDI structure. Depending on the circumstances, you may use the Split or Copy/Cut and Paste functions instead.

References

- ▶ See *How to Split Groups and Repeating Segments* on page 2 - 44 for more information.
- ▶ See *How to Use Copy, Cut, and Paste* on page 2 - 48 for more information.

Procedure

Use this procedure to promote a group or repeating segment.

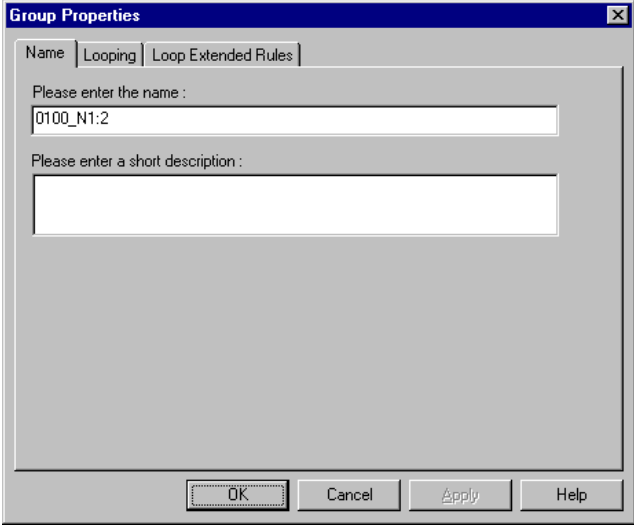
Step	Action
1	Select the group or repeating segment from which you want to extract one iteration
2	Click Promote on the Main Toolbar. System response This extracts one iteration (instance) of the looping structure.

Outbound Example

In this example, your application file contains a Ship To record that groups all the shipping information and a Bill To record that groups all the billing information. Your Ship To and Bill To records both occur only once in the application. The EDI file contains a N1 group (for ANSI) or a NAD group (for EDIFACT) that repeats 200 times. If the Entity Identifier Code of the N1/NAD is “BT,” that means the N1/NAD group contains billing information. If the Entity Identifier Code is “ST,” the N1/NAD group contains shipping information.

Example Procedure

This procedure is an example of how to map unique data to your application file.

Step	Action
1	Activate the N1/NAD loop and the appropriate segments and elements, if you have not already done so. If you already activated the N1/NAD group, just highlight it. Reference See <i>How to Activate EDI Map Components</i> on page 2 - 37 for more information.
2	Click Promote on the Main Toolbar to extract a single instance of the N1/NAD group.
3	From the Edit menu, select Properties . System response The system displays the Group Properties dialog box (the Name tab is displayed by default). 
4	In the description box, type a specialized definition to identify this group as containing billing information. <div style="text-align: right; color: green;">(Continued on next page)</div>

(Contd) Step	Action
5	<p>Click OK.</p> <p>System response The system saves the change and exits the Group Properties dialog box.</p>
6	<ul style="list-style-type: none"> ▶ Right-click the original N1/NAD group (that now repeats 199 times) again and ▶ from the Edit menu, select Properties. <p>System response The system displays the Group Properties dialog box (the Name tab is displayed by default)</p>
7	<p>In the description box, type a specialized definition to identify this group as containing billing information.</p>
8	<p>Click OK.</p> <p>System response The system saves the change and exits the Group Properties dialog box.</p>
9	<ul style="list-style-type: none"> ▶ Click the Deactivate button on the Main Toolbar and then ▶ click the original N1/NAD group (that now repeats 198 times), so the system does not process it.
10	<p>Now you can link the elements in the single occurrence of the N1/NAD that contains billing information to the fields in your Bill To application record.</p> <p>And, you can link the elements in the single occurrence of the N1/NAD that contains shipping information to the fields in your Ship To application record.</p>

How to Split Groups and Repeating Segments

Introduction

The Split function enables you to split (break) a group or repeating segment into two loops. You typically use this function when you need more than one instance of the same map component that still occurs multiple times.

Split vs. copy and paste

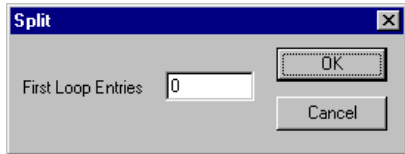
You can use the Copy and Paste functions (and change the number in the maximum usage box) to accomplish the same task. However, Split is a specialized function that guarantees the integrity of your EDI structure.

Note

The Split function is only available if a group or repeating segment is selected.

Procedure

Use this procedure to split a group or repeating segment.

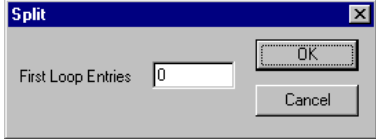
Step	Action
1	Highlight the group or repeating segment from which you want to extract one iteration.
2	<p>Click Split on the Main Toolbar.</p> <p>System response The system displays the Split dialog box.</p> 
3	<p>In the First Loop Entries box, type the number that indicates the sequential number of iterations where you want the group or repeating segment split.</p> <p>Example If the X loop repeats a maximum number of 5 times, and you type “2” in the First Loop Entries box, the resulting split generates one X loop that repeats a maximum of 2 times and a second X loop that repeats a maximum of 3 times.</p> <p>Note The number you enter in the First Loop Entries box must be greater than zero and less than the maximum number of iterations of the loop.</p>
4	<p>Click OK.</p> <p>System response Exits the Split dialog box.</p> <p style="text-align: right;">(Continued on next page)</p>

Outbound Example

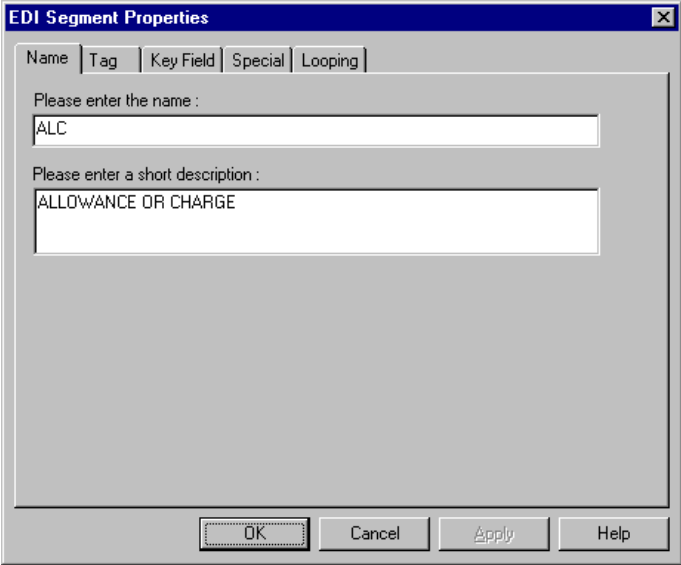
In this example, your application file contains an Allowance record that groups all the allowance information and a Charge record that groups all the charge information. Your Allowance and Charge records can repeat up to five times in the application file. The EDI file contains a ALC repeating segment that repeats 10 times. If you split the ALC segment in two, you can map allowance information directly to your Allowance application record and map charge information directly to your Charge application record.

Example procedure

This procedure is an example of how to split the ALC segment.

Step	Action
1	Activate the ALC segment and the appropriate elements, if you have not already done so. If you already activated the ALC segment, just highlight it. Reference See <i>How to Activate EDI Map Components</i> on page 2 - 37 for more information.
2	Click Split on the Main Toolbar. System response The system displays the Split dialog box. 
3	In the First Loop Entries box, type “5.” Note This is the number of sequential iterations where you want the repeating segment split.
4	Click OK . System response Exits the Split dialog box. <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
5	<p>From the Edit menu, select Properties.</p> <p>System response The system displays the EDI Segment Properties dialog box (Name tab).</p> 
6	In the description box, type a specialized definition to identify this segment as containing allowance information.
7	<p>Click OK.</p> <p>System response The system saves the change and exits the EDI Segment Properties dialog box.</p>
8	Highlight the original ALC segment (that now repeats 5 times).
9	<p>From the Edit menu, select Properties.</p> <p>System response The system displays the EDI Segment Properties dialog box (Name tab).</p>
10	In the description box, type a specialized definition to identify this segment as containing allowance information.
11	<p>Click OK.</p> <p>System response The system saves the change and exits the EDI Segment Properties dialog box.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
12	<p>Now you can link the elements in the ALC segment that contains allowance information to the fields in your Allowance application record.</p> <p>And, you can link the elements in the ALC segment that contains charge information to the fields in your Charge application record.</p>

How to Use Copy, Cut, and Paste

Introduction

The Copy, Cut, and Paste functions are typically used to move EDI information in the map. You may need to use these functions if you are using an EDI standard differently, or if you need to create nested looping structures.

You can cut or copy a single map component (loop, segment, composite, element, record, or field) and paste it in another location in the map. Copied map components retain all the information of the original map component. If the copied map component contains subordinate map components (like a segment contains subordinate elements), the subordinate map components are also copied.

You can also cut, copy, and paste a map component from one map to another.

Procedure

Use this procedure to cut, copy, and paste a map component.

Step	Action
1	Select the map component that you want to cut or copy.
2	From the Edit menu, select Copy . Note If you are pasting the map component in another map, open that map, if it is not already open.
3	Select the map component that you want the cut or copied selection pasted <i>after</i> . Note If you cut or copied an EDI component, you can only paste that component into the EDI side of a map. Conversely, if you cut or copied a positional component, you can only paste that component into the positional side of a map.
4	Click Paste on the Main Toolbar to paste the contents of the Clipboard. System response If the map component that you selected is a group, Gentran:Server prompts you to specify whether you want the contents of the Clipboard pasted as a child (subordinate) of the group or pasted at the same level as the group. Select the appropriate option and click OK .

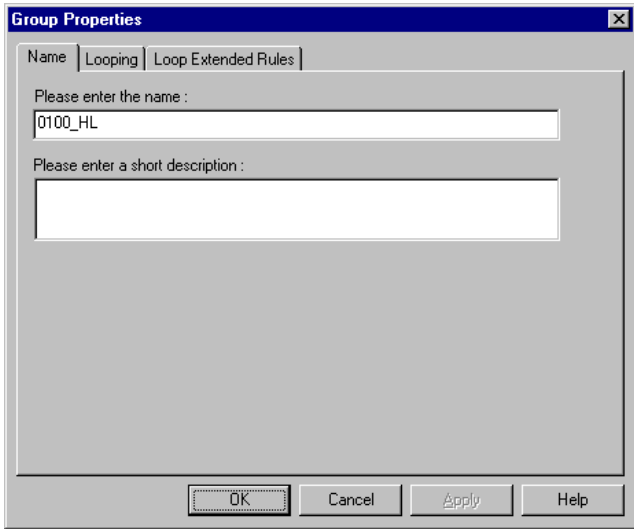
Example

In this example, you are defining an ANSI X12 Ship Notice/Manifest (856). When the system generates the 856 document, it contains one HL group. The HL group identifies dependencies between and the content of hierarchically-related groups of segments.

However, for your EDI file, you need to define additional levels of detail information. The details that you need to define include shipment, order, and line item information. In order to modify the structure of your EDI file to accommodate these requirements, you need to use the Copy and Paste functions to create a nested looping structure for the HL group.

Example procedure

This procedure is an example of how to create a nested looping structure.

Step	Action
1	<ul style="list-style-type: none"> ▶ Highlight the HL group and ▶ click Copy on the Main Toolbar.
2	<p>From the Edit menu, select Properties.</p> <p>System response The system displays the Group Properties dialog box.</p> 
3	<p>In the second box on the Name tab, type a specialized definition to identify this group as containing shipment information.</p>
4	<p>Click OK.</p> <p>System response Saves the change and exits the Group Properties dialog box.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
5	<ul style="list-style-type: none"> ▶ Select the map component in the HL that you want the nested HL group (order information) pasted <i>after</i> and ▶ click Paste. <p>System response The second HL group is added as a nested loop to the original HL loop.</p>
6	<p>From the Edit menu, select Properties.</p> <p>System response Displays the Group Properties dialog box for the second HL group.</p>
7	<p>In the second box on the Name tab, type a specialized definition to identify this group as containing shipment information.</p>
8	<p>Click OK.</p> <p>System response Saves the change and exits the Group Properties dialog box.</p>
9	<p>Double-click the second HL group (order information) to open it.</p>
10	<ul style="list-style-type: none"> ▶ Select the map component in the HL that you want the nested HL group (item information) pasted <i>after</i>, and ▶ click Paste. <p>System response The third HL group (item information) is added as a nested loop of the second HL group.</p>
11	<p>From the Edit menu, select Properties.</p> <p>System response Displays the Group Properties dialog box for the third HL group.</p>
12	<p>In the second box on the Name tab, type a specialized definition to identify this group as containing shipment information.</p>
13	<p>Click OK.</p> <p>System response Saves the change and exits the Group Properties dialog box.</p>
14	<p>Now you can activate the appropriate groups, segments, and elements in the new HL nested looping structure.</p> <p>Reference See <i>How to Activate EDI Map Components</i> on page 2 - 37 for more information.</p>

Using a Fixed-Format File

Overview

Introduction

If a side of your map is positional (fixed-format), you must either define your application to Gentran:Server or load a previously-saved file definition. Your application file must contain all the information that you need to extract from your partner's document (if the map is inbound) or send to your partner (if the map is outbound).

Before you begin

Before you define your application, you should obtain a layout of the necessary records, fields, and groups. The records contain related fields and the groups contain related records. For example, your application contains records and groups. The records contain fields and the groups contain records, subgroups, or both. This means that you must create records and groups before you create the subordinate fields.

Each map component is arranged sequentially in the order that it is most logical for the system to process. Therefore, each level of your application must be created sequentially.

Create Sub and Insert functions

You will use two different Gentran:Server functions to create the necessary group and records—*Create Sub* and *Insert*. This table explains when you use each of these functions.

IF you want to create a group or record...	THEN right-click the map component above and select...
at the same level (equal),	<ul style="list-style-type: none"> ▶ Insert from the shortcut menu, and ▶ select Group or Record from the submenu.
that is subordinate to the selected map component,	<ul style="list-style-type: none"> ▶ Create Sub from the shortcut menu, and ▶ select Group or Record from the submenu.

How to Adjust File Format Properties

Changing record delimiters

To set the record delimiters:

Step	Action
1	In Application Integration, open the positional file.
2	Right-click the file format icon and select Properties from the shortcut menu. System response The Positional File Properties dialog box is displayed.
3	In the Record tab, change the Record Delimiters 1 and 2 as needed.
4	Click OK . You must click OK, even if the delimiter fields are empty, to reset the delimiters.
5	Save the map.
6	Compile the map. System response The end-of-line delimiters are changed

Changing decimal points

If you need to specify that each decimal point in your data is defined and generated as a something other than the default period (.), you can do so on the Decimal Point tab.

To change the default decimal point setting:

Step	Action
1	In Application Integration, open the positional file.
2	Right-click the file format icon and select Properties from the shortcut menu. System response The Positional File Properties dialog box is displayed.
3	In the Decimal Point tab, select the Define Decimal Point check box.
4	In the Decimal Point Character box, type the value you want Sterling Integrator to use as a decimal point—for example, a comma (,). This resets the decimal point default to use what you specified instead of a period. (Continued on next page)

(Contd) Step	Action
5	Click OK . You must click OK, even if the delimiter fields are empty, to reset the delimiters.
6	Save the map.
7	Compile the map. System response The decimal point setting is now correctly set to use your specification.

How to Create the First Record

Introduction

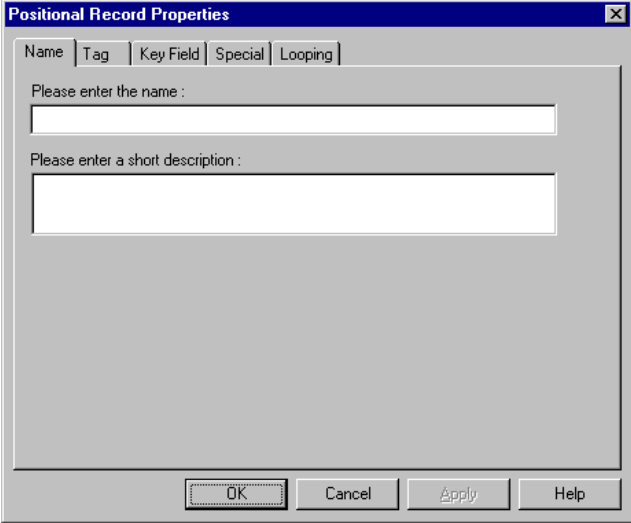
The first record in your application is generally the header record. The first record is subordinate to the icon in the application side of the map. This shows how the records (and groups) are contained within it.

Tip

We recommend that you include a header record in each application file. The header record is mandatory and repeats only once (not a looping structure).

Procedure

Use this procedure to create the first application record.

Step	Action
1	<ul style="list-style-type: none"> ▶ Click the Positional File icon to highlight it (OUTPUT for an inbound map and INPUT for an outbound map). ▶ From the Edit menu select Create Sub. ▶ Select Record from the submenu. <p>System Response The system displays the Positional Record Properties dialog box.</p> 
2	In the first box on the Name tab, type the record name.
3	<p>In the second box on the Name tab, type a description of the record.</p> <p>Note This box is used to provide a brief explanation of the record that allows you to differentiate it from similar records.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
4	Select the Tag tab to access tag information.
5	<p>In the Tag box, type the record identification code <TAG>. The record tag for each record enables the system to recognize that record and then determine the mapping requirements.</p> <p>Example A record is recognized by the system as:</p> <p style="padding-left: 40px;"><TAG>[Field_1][Field_2]</p> <p>Recommendation Define the record tag on the Positional Record Properties dialog box, instead of defining fields with the purpose of explicitly containing the record tag.</p>
6	In the Position box, type the starting column position of the tag in the data record.
7	Select the Special tab only if you want to specify floating (the record does not have a fixed position in the file) or wildcard (generally only used with build or break maps) options. Otherwise, accept the defaults.
8	Select the Looping tab <i>only</i> if the record is a looping structure. Otherwise, accept the defaults and continue with step 11.
9	<p>Is the record a looping structure?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, on the Looping tab in the Min Usage box, type the minimum amount of times the record must repeat. <p>Recommendation Type “1” to make the header record mandatory.</p> <p>Note If the Min Usage box contains a “0” (zero), the record is “conditional.” If the Min Usage box contains a “1” or greater, the record is “mandatory.”</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with step 11.
10	<p>Is the record a looping structure?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, on the Looping tab in the Max Usage box, type the maximum amount of times it can repeat. <p>Recommendation For the header record, we recommend that you accept the default of “1” so it remains a record that only occurs once (not a looping structure).</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with step 11.
11	Click OK to create the record.



How to Create Subsequent Records

Introduction

After you create the first record, you can define subsequent records at the same level. You can also define groups, if your application includes looping structures. You may need to define a loop if your application requires a group of related records (and/or subgroups) that repeat in sequence.

Reference

See *How to Create a Group* on page 2 - 62 for more information.

Procedure

Use this procedure to create additional application records.

Step	Action
1	<ul style="list-style-type: none"> ▶ Highlight the map component which precedes the record you are creating in the file layout. ▶ From the Edit menu select Insert or Create Sub, depending on the level. ▶ Select Record from the submenu. <p>System Response The system displays the Positional Record Properties dialog box.</p>
2	In the first box on the Name tab, type the record name.
3	In the second box on the Name tab, type a description of the record. Note This box is used to provide a brief explanation of the record that allows you to differentiate it from similar records.
4	Select the Tag tab to access tag information.
5	In the Tag box, type the record identification code <TAG>. The record tag for each record enables the system to recognize that record and then determine the mapping requirements. Example A record is recognized by the system as: <TAG>[Field_1][Field_2] Recommendation Define the record tag on the Positional Record Properties dialog box, instead of defining fields with the purpose of explicitly containing the record tag.
6	In the Position box, type the starting column position of the tag in the data record.

(Continued on next page)

(Contd) Step	Action
7	Select the Special tab only if you want to specify floating (the record does not have a fixed position in the file) or wildcard (generally only used with build or break maps) options. Otherwise, accept the defaults.
8	Select the Looping tab <i>only</i> if the record is a looping structure. Otherwise, accept the defaults and continue with step 11.
9	<p>Is the record a looping structure?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, on the Looping tab in the Min Usage box, type the minimum amount of times the record must repeat. <p>Recommendation Type “1” to make the header record mandatory.</p> <p>Note If the Min Usage box contains a “0” (zero), the record is “conditional.” If the Min Usage box contains a “1” or greater, the record is “mandatory.”</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with step 11.
10	<p>Click OK to create the record.</p> <p>Note Use this procedure to create any additional records.</p>

How to Create a Temporary Record

Introduction

You may need to use temporary records and fields when you cannot use a simple link or if you need to extract only specific occurrences of a record from your data file. A simple link allows you to join data from the Input and Output sides of the map in either a one-to-one relationship (map components that both do not repeat) or a many-to-many relationship (map components that repeat the same number of times).

Reference

See *How to Create Simple Links* on page 2 - 114 for more information.

When to use temporary records and fields

If link a single iteration of a map component on the Input side of a map to a map component on the Output side that repeats multiple times, it causes an infinite loop when the map is translated. And, if you link a map component on the Input side that repeats multiple times to a map component on the Output side that does not repeat, the data from the Input side never populates the Output side of the map. In either of these instances, you should use temporary records and fields.

These are other common reasons why you would want to incorporate temporary records and field into your map:

- The Input hierarchical level in a map does not match the Output hierarchical level.
- You only want to populate an Output field with data if a specified qualifier is used or if specific criteria is met.

You may need to use extended rules in addition to creating temporary records and fields.

Reference

See *Using Extended Rules*, chapter 6, for more information.

Where to use temporary records and fields

You can add temporary records and fields at any hierarchical level in a map. However, when you use temporary records and fields, you must locate them immediately after the map component that contains the necessary data.

The following caveats apply to temporary records:

- You must use a record tag that you would never receive in your Input file, and the recommended default is \$\$\$.
 - If you are creating a temporary record for an XML file, use the tag **XXX**, because \$\$\$ is not permitted as a tag in XML.
 - The system does not run standard or extended rules on temporary records.
-

Example To map the shipping information from the Input side of your map (from the N1 Group that repeats a maximum of 200 times) to the ShipTo record in your application file format (that does not repeat), you need to create a temporary storage record and fields on the EDI side of the map that do not repeat. Then, you use an extended rule to extract the shipping and billing information from the N1 group and move it to the appropriate temporary storage elements. Finally, you map the shipping and billing information directly from the temporary storage elements to your application fields.

Step 1 First, you need to create a temporary storage record and fields on the EDI side of the map that do not repeat. The temporary record is located outside the N1 Group at the same hierarchical level as it, and has a maximum repeat number of “1” (i.e., it does not repeat). Then you will create the appropriate temporary fields.

Reference

See *How to Create Fields* on page 2 - 64 for more information.

This table lists the temporary storage record that you are creating.

Name	Description	Tag	Min Usage	Max Usage
ShipToDet	Ship To Details	\$\$\$	0	1

Example procedure

Use this procedure to create the ShipToDet temporary storage record.

Step	Action
1	Highlight the N1 group. The two temporary storage segments are located after the N1 group, at the same level.
2	From the Edit menu, select Insert . From the submenu, select Segment (you select Segment instead of Record because you are creating the temporary record on the EDI side of the map). System response The EDI Segment Properties dialog box is displayed.
3	In the Name box, type ShipToDet .
4	In the Description box, type Ship To Details .
5	Select the Tag tab.

(Continued on next page)



(Contd) Step	Action
6	In the Tag box, type \$\$\$. Note The system does not read a segment with a value of “\$\$\$” in the Tag box. Therefore, it does not flag this temporary storage segment as an error during compliance checking.
7	Click OK to create the ShipToDet temporary storage segment.
8	Create the appropriate temporary fields. Example <ul style="list-style-type: none"> ▶ SHIPTONAME ▶ SHIPTOADDR1 ▶ SHIPTOADDR2 ▶ SHIPTOCITY ▶ SHIPTOSTATE ▶ SHIPTOPCODE Note See <i>How to Create Fields</i> on page 2 - 64 for more information.

Step 2 Create an ON_END extended rule for the N1 Group that assigns the necessary information to the temporary fields.

Example procedure

Use this procedure to create the extended rule.

Step	Action
1	Right-click the N1 group to access the shortcut menu.
2	From the shortcut menu, select Extended Rules to display the Group Properties dialog box.
3	Select the On End option. This specifies that the rule is executed when the loop terminates. The system loads an occurrence of the N1 group (containing the billing or shipping information), and then executes this rule. <div style="text-align: right; color: green;">(Continued on next page)</div>

(Contd) Step	Action
4	<p>In the Editor list, type the following:</p> <pre> IF #0098 = "BT" THEN BEGIN \$850.#BILLTONAME = #0093; \$850.#BILLTOADDR1 = #0166; \$850.#BILLTOADDR2 = #0166:2; \$850.#BILLTOCITY = #0019; \$850.#BILLTOSTATE = #0156; \$850.#BILLTOPCODE = #0116; END IF #0098 = "ST" THEN BEGIN \$850.#SHIPTONAME = #0093; \$850.#SHIPTOADDR1 = #0166; \$850.#SHIPTOADDR2 = #0166:2; \$850.#SHIPTOCITY = #0019; \$850.#SHIPTOSTATE = #0156; \$850.#SHIPTOPCODE = #0116; END </pre> <p>Note If a segment/record or element/field occurs more than once in a map, it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where 'n' is the number of occurrences in the map.</p>
5	<p>Click Compile to validate the syntax of the extended rule. Every rule in the map is compiled when you compile the translation object, after you complete the map. However, the system allows you to compile each rule individually, so that you can verify the accuracy of the rule after you create it.</p> <p>Note This compiles the rule interactively, and allows you to correct any errors that are generated. Any errors or warnings generated in the compilation process are displayed in the Errors list.</p>
6	<p>Click OK to add the extended rule to the N1 group.</p>

Step 3 Link the temporary fields on the Input side of the map with the corresponding Output fields.



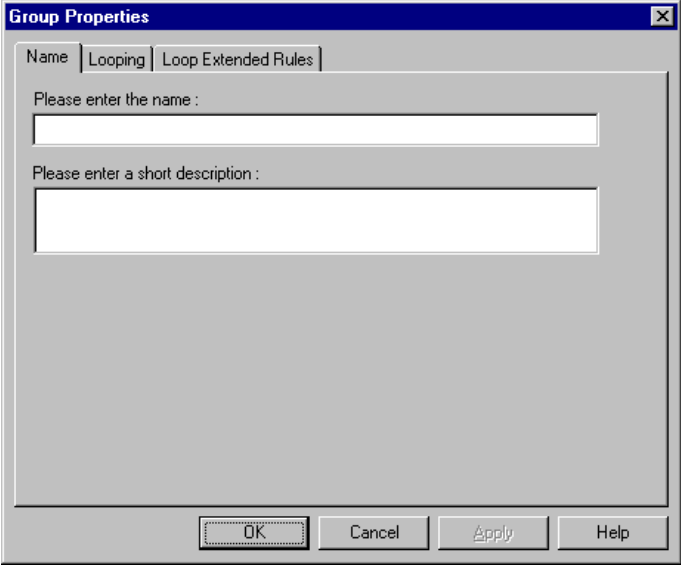
How to Create a Group

Introduction

A group contains related records, groups, or both that repeat in sequence until either the data ends or the maximum number of times that the loop is allowed to repeat is exhausted. If you create a group that is subordinate to another group, this corresponds to a nested looping structure (a loop within a loop).

Procedure

Use this procedure to create a group.

Step	Action
1	<ul style="list-style-type: none"> ▶ Highlight the map component which precedes the group you are creating in the file layout. ▶ From the Edit menu select Insert or Create Sub, depending on the level. ▶ Select Group from the submenu. <p>System Response The system displays the Group Properties dialog box (Name tab displayed by default).</p> 
2	<p>In the first box on the Name tab, type the group name.</p> <p>Note Do not use spaces or dashes (-) in the group name. You can use the underscore () to separate words.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
3	<p>In the second box on the Name tab, type the loop description.</p> <p>Note This box is used to provide a brief explanation of the loop.</p>
4	<p>Select the Looping tab to access looping options.</p>
5	<p>In the Minimum Usage box, type the minimum number of times the loop must be repeated.</p> <p>Note For a conditional loop, the minimum usage should always be “0” (zero). For a mandatory loop, the minimum usage should be “1” or greater.</p>
6	<p>In the Maximum Usage box, type the maximum number of times the loop can be repeated.</p>
7	<p>Is this a single iteration group?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select Promote records to parent if you want to specify that the subordinate records and groups should be extracted and located in the parent group when the group is compiled. ▶ If <i>no</i>, continue with the next step.
8	<p>Do you want to specify an extended rule for this group?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select the Loop Extended Rules tab. <p>Reference See <i>Defining Extended Rules</i> on page 6 - 12 for more information.</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with the next step.
9	<p>Click OK to create the group.</p>

How to Create Fields

Introduction

Each record you create contains a group of logically-related application fields. These fields define the structure and content of the data to the system.

The easiest way to add application fields to a record is to use the Positional Field Editor. Generally, you create the fields for the first record in the application file, and then proceed with each sequential record.

Recommendation

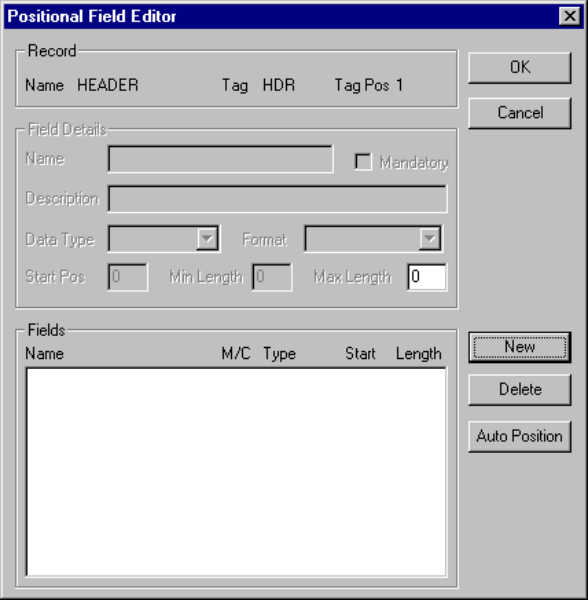
Do not define fields with the purpose of using the field to explicitly contain the record tag. This is because the system takes the tag that you define in the record into account when the automatic sequencing (Auto Position) feature is used. We recommend that you define the record tag on the Positional Record Properties dialog box.

Reference

See *How to Create Subsequent Records* on page 2 - 56 for more information about the Positional Record Properties dialog box.

Procedure

Use this procedure to create the application fields for a record.

Step	Action
1	<p>Right-click the application record and select Edit Fields from the shortcut menu.</p> <p>System response The system displays the Positional Field Editor dialog box.</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
2	<p>Is the field you are creating the first field in the record?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, click New and continue with the next step. ▶ If <i>no</i>, highlight the field that precedes the field you are creating in the record layout and click New. <p>System response The system displays a highlight bar in the Fields section where the new field is positioned.</p> <p>Note You need to fill in the field values in the Field Details section.</p>
3	<p>In the Name box, type the field name.</p> <p>Notes</p> <ul style="list-style-type: none"> ▶ Each application field must have a unique name. It is useful to tag the end of the fields that occur in multiple records with a suffix that identifies the record that contains it. ▶ Do not use spaces or dashes (-) in the field name. You can use the underscore (_) to separate words.
4	<p>Do you want to designate the field as “mandatory” (must be present)?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select the Mandatory check box. ▶ If <i>no</i>, continue with the next step.
5	<p>In the Description box, type a description of the field.</p> <p>Note The description is used to provide a brief explanation of the field that allows you to differentiate it from similar fields.</p>
6	<p>From the Data Type list, select the type of the field.</p> <p>Note Valid values are:</p> <ul style="list-style-type: none"> ▶ String = alpha element ▶ Number = numeric or real element ▶ Date/Time = date or time element <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
7	<p>From the Format list, select how the field is formatted.</p> <p>Note The choices for this field depend on the type of field you selected from the Data Type list. If you choose Number or Date/Time in the Data Type box, you can select the data format from the Format list. If you selected String from the Data Type box, you should type a syntax token to denote that this field must be formatted as the specified syntax token dictates (the default syntax token is “X”).</p> <p>When you installed Gentran:Server, you assigned a default format to the string fields. This format serves as the basis for character validation. Most U.S. users use a default format that corresponds to ASCII characters (for example, the X syntax token). Most users of Asian or European languages and encoded character sets should use the Free Format (0x01-0xFF).</p> <p>Reference See <i>How to Use Syntax Tokens</i> on page 2 - 106 for more information.</p>
8	<p>Do you want to indicate the exact position of the field in the record?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, type the starting position of the field in the Start Pos box. ▶ If <i>no</i>, continue with the next step. <p>Note You want to specify field start positions if, for example, you are only using a few fields but you want them positioned exactly in the record. The alternative to specifying the start position of each field, is to add the fields sequentially in the record and then use the Auto Position function.</p> <p>Reference See step 13 below for more information about the Auto Position function.</p>
9	<p>In the Min Length box, type the minimum number of characters that is acceptable for this field.</p>
10	<p>In the Max Length box, type the maximum length of the field.</p>
11	<p>Click New.</p> <p>System response The system adds the field and creates a new field with blank values ready for you to identify, which is positioned below the field.</p>
12	<p>Create the rest of the fields according to your record layout.</p> <p>Note Click Delete to stop adding fields.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
13	<p>After adding the last field, do you want the system to automatically position the fields in the record?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, click Auto Position. <p>Note This automatically calculates the start position in the record of each field based on the criteria that each field is positioned directly after the previous field and is of the length specified in the Max Length box. Click Yes to acknowledge the warning message that fields are sequenced in order.</p> <p>Warning It is only valid to use the Auto Position function if you define a record tag, and if you define every field in the record in the sequence that each field occurs.</p> <p>Reference See <i>How to Create Subsequent Records</i> on page 2 - 56 for more information.</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with the next step.
14	<p>Are you completely finished adding fields to the record?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, click Close. ▶ If <i>no</i>, repeat steps 2 through 13. <p>Note Use this procedure to add fields to the other records you defined.</p> <p>For inbound maps, you may want to use a literal constant to move a value into an application field that has no corresponding EDI data (you cannot link the field directly to an EDI element).</p> <p>Reference See <i>How to Map Literal Constants</i> on page 3 - 32 for more information about mapping a constant to a field.</p>

Using an EDI File

Overview

Introduction

Gentran:Server generates an EDI file for you, based on the standard (agency), version, transaction set, and release you select. The system includes all the groups, segments, composites, and elements that are defined by the standards agency for the version of the document you select.

Reference

See *How to Create a New Map* on page 2 - 20 for more information about generating an EDI file when you create a new map.

You can modify the system-generated EDI file by modifying the properties of the map components, and using the promote, split, copy, cut, and paste functions.

However, if you want to use a specialized version of an EDI standard that is not available in the Gentran:Server standards database, it may be appropriate for you to either load an EDI file definition or define the EDI file yourself.

Reference

See *Using a CII File* on page 2 - 89 for more information about defining a CII file.

Regardless of whether the system generates the EDI file or you load or define it, the specific EDI map components that you use depends on the type of map you are creating. This includes the standard, version, and transaction set (document) selected, and which groups, segments, composites, and elements your company requires. We recommend that you determine which map components you are using before generating or defining an EDI file.

How to Verify EDI Delimiters

Introduction

If you are using an EDI standard that contains composite elements or subelements, you must verify that Gentran:Server is specifying the correct delimiters. Delimiters are flags that you define to the system as separating specific EDI components.

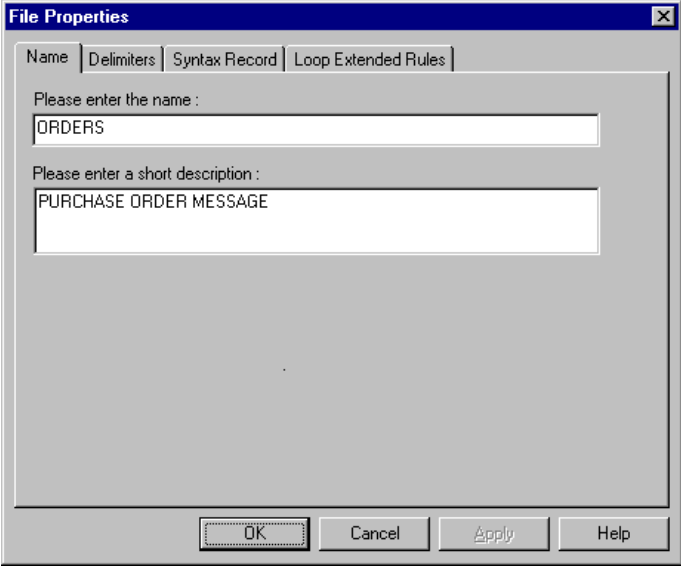
Delimiters are necessary for all variable field-length standards, because the data is compressed (and the leading zeroes and trailing blanks are removed). Since the fields vary in length, the system needs a flag to determine where one element ends and another begins. For example, an element delimiter marks the beginning of a new element.

Recommendation

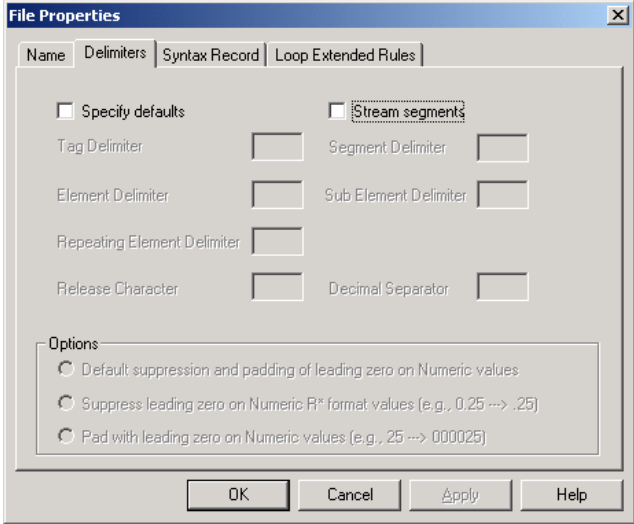
Although verifying EDI delimiters in Gentran:Server is mandatory *only* if you are using a standard with composite elements or subelements, we recommend that you perform this task regardless of which standard you use.

Procedure

Use this procedure to verify EDI delimiters.

Step	Action
1	<p>Right-click the EDI file icon and select Properties from the shortcut menu.</p> <p>System response The system displays the File Properties dialog box.</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
2	<p>Select the Delimiters tab to access delimiter options.</p> 
3	Verify that the Specify defaults check box is selected.
4	<p>Verify the required delimiters for the EDI standard you are using.</p> <p>Note If the delimiters differ from the defaults specified, type either the character or the hexadecimal value in the correct box.</p>
5	<p>Are you using an X12, EDIFACT, or TRADACOMS standard on the output side of a map and you want to suppress leading zeros for the EDI elements that use an R-format numeric (including decimal values less than one)?</p> <ul style="list-style-type: none"> ▶ If Yes, select the Suppress leading zero on Numeric R* format values (e.g., 0.25--> .25) option. ▶ If No, continue with Step 6.
6	<p>Are you using an X12, EDIFACT, or TRADACOMS standard on the output side of a map and you want to pad numeric values (regardless of format) for the EDI elements with leading zeros?</p> <ul style="list-style-type: none"> ▶ If Yes, select the Pad with leading zero on Numeric values (e.g., .25--> 000025) option. <p>Note When you select this option, Numeric values (regardless of format) are padded with leading zeros out to the maximum length of the field.</p> <ul style="list-style-type: none"> ▶ If No, continue with Step 7.
7	Click OK to exit the File Properties dialog box.

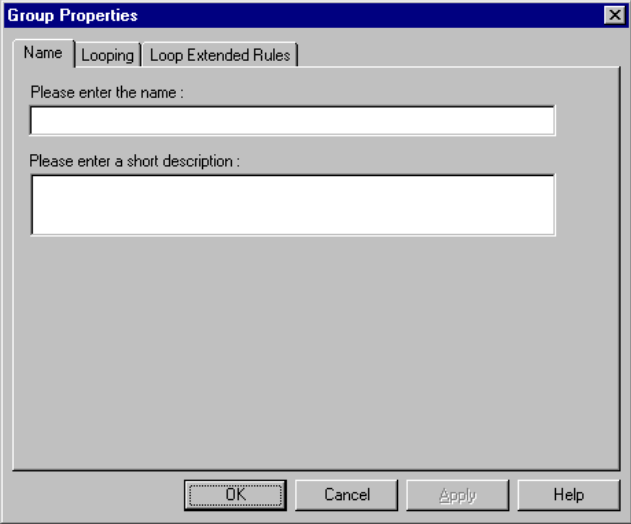
How to Modify Group Properties

Introduction

A group contains related segments and/or groups that repeat in sequence until either the group data ends or the maximum number of times that the loop is allowed to repeat is exhausted. If you create a group that is subordinate to another group, this corresponds to a nested looping structure (a loop within a loop).

Procedure

Use this procedure to modify the properties of a group.

Step	Action
1	<p>Right-click the group you want to modify and select Properties from the shortcut menu.</p> <p>System Response The system displays the Group Properties dialog box (Name tab displayed by default).</p> 
2	<p>If you want to modify the group name, in the first box on the Name tab, type the new name.</p> <p>Note Do not use spaces or dashes (-) in the group name. You can use the underscore () to separate words.</p>
3	<p>If you want to modify the group description, in the second box on the Name tab, type the new description.</p>
4	<p>Select the Looping tab to access looping options.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>



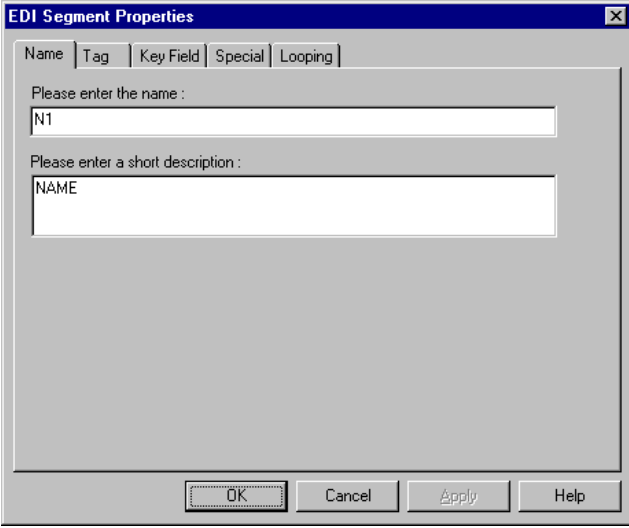
(Contd) Step	Action
5	<p>In the Minimum Usage box, type the minimum number of times the loop must be repeated.</p> <p>Note For a conditional loop, the minimum usage should always be “0” (zero). For a mandatory loop, the minimum usage should be “1” or greater.</p>
6	<p>In the Maximum Usage box, type the maximum number of times the loop can be repeated.</p>
7	<p>Is this a single iteration group?</p> <ul style="list-style-type: none"> • If <i>yes</i>, select Promote records to parent if you want to specify that the subordinate records and groups should be extracted and located in the parent group when the group is compiled. • If <i>no</i>, continue with the next step.
8	<p>Do you want to specify an extended rule for this group?</p> <ul style="list-style-type: none"> • If <i>yes</i>, select the Loop Extended Rules tab. <p>Reference See <i>Defining Extended Rules</i> on page 6 - 12 for more information.</p> <ul style="list-style-type: none"> • If <i>no</i>, continue with the next step.
9	<p>Click OK to save changes to the group.</p>

How to Modify Segment Properties

Introduction You can modify the properties of any segment, including the minimum and maximum number of times the segment can repeat, whether the segment is mandatory or conditional, and whether it is a loop start or loop end segment.

Reference
See *How to Use Loop Start and Loop End Segments* on page 2 - 76 for more information.

Procedure Use this procedure to modify the properties of a segment.

Step	Action
1	<p>Right-click the segment you want to modify and select Properties from the shortcut menu</p> <p>System Response The system displays the EDI Segment Properties dialog box.</p> 
2	<p>If you need to modify the name of the segment, type the segment name in the first box on the Name tab.</p> <p>Note If a segment occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where “n” is the number of the occurrence in the map.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
3	<p>If you want to modify the description of the segment, type a new description in the second box on the Name tab.</p> <p>Note This box is used to provide a brief explanation of the segment that allows you to differentiate it from similar segments.</p>
4	<p>Select the Tag tab only if you want to access tag information.</p>
5	<p>If you want to modify the segment tag, type the segment identification code <TAG> in the Tag box.</p> <p>Note The segment tag for each segment enables the system to recognize that segment.</p> <p>Example A record is recognized by the system as:</p> <pre><TAG>[Delimiter]<DATA>[Delimiter] <DATA>.....<DATA>[Segment Terminator]</pre>
6	<p>Select the Key Field tab only if you want to use a key field.</p> <p>Note The key field function enables you to specify a second qualification in selecting a segment (the segment name is the first qualification). Data must be provided in the order designated through the use of key fields.</p>
7	<p>Select the Special tab only if you want to specify floating (the segment does not have a fixed position in the file) or wildcard (generally only used with build or break maps) options, or if you want to flag this segment as containing binary data. Otherwise, accept the defaults.</p> <p>Note If you select “Binary,” you must define an element of data-type “Bin Length” and another element of data-type “Bin Data.” The “Bin Length” element must precede the “Bin Data” element.</p> <p>Reference See <i>How to Modify Element Properties</i> on page 2 - 84 for more information about defining an element data-type.</p>
8	<p>Is the segment a looping structure?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select the Looping tab. ▶ If <i>no</i>, accept the defaults and continue with step 12. <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
9	<p>In the Min Usage box, type the minimum amount of times the segment must repeat.</p> <p>Note If the Min Usage box contains a “0” (zero), the segment is “conditional.” If the Min Usage box contains a “1” or greater, the segment is “mandatory.”</p>
10	<p>Do you want the segment to repeat (loop)?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, type the maximum amount of times it can repeat in the Max Usage box. ▶ If <i>no</i>, continue with the next step.
11	<p>Do you want to indicate that the segment is the beginning of a loop (Loop Start) or the end of a loop (Loop End)?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select one of the following options. Loop Start (this segment marks the beginning of the loop) Loop End (this segment marks the end of the loop) ▶ If <i>no</i>, accept the default of Normal for the Normal/Loop Start/Loop End options, which indicates that this segment is in the loop but is not the beginning or ending segment
12	Click OK to save changes to the segment.

How to Use Loop Start and Loop End Segments

Introduction

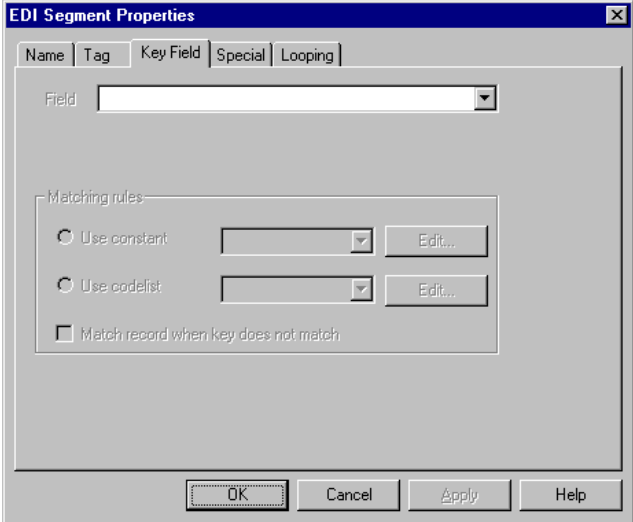
Certain EDI standards use Loop Start (LS) and Loop End (LE) segments. LS and LE segments differentiate between two or more loops of the same type. If the transaction contains LS and LE segments and depending on whether your map is inbound or outbound, you need to define the LS and LE segments for the loops you are using in the map in one of two different ways.

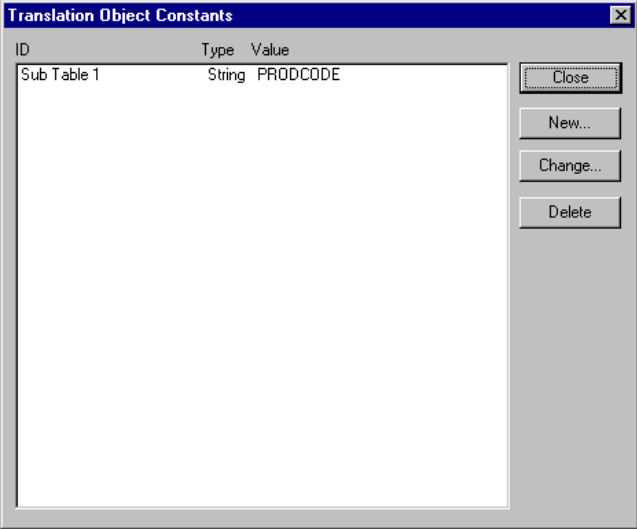
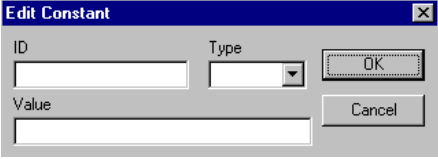
Tip

You need to communicate with your trading partner to determine which loop identification code your partner sends you (for an inbound map), or expects to receive (for an outbound map).

Procedure: defining a LS segment inbound

Use this procedure to define an LS segment for an inbound map.

Step	Action
1	<p>Right-click the LS segment and select Properties from the shortcut menu.</p> <p>System response The system displays the EDI Segment Properties dialog box.</p>
2	<ul style="list-style-type: none"> ▶ Select the Looping tab to access the loop options and ▶ verify that the Loop Start option is selected.
3	<p>Select the Key Field tab.</p> <p>System response The system displays the key field options.</p>  <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
4	<p>From the Field list, select the “Loop Identifier Code.”</p> <p>Note This list contains all the elements that are contained in the segment and gives you the ability to define the Loop Start segment definition, by specifying that the loop identifier code must have the value you specify in the Matching rules section.</p>
5	<p>Select the Use constant option.</p> <p>Note The Matching rules section enables you to access all the literal constants and code lists currently defined for this map.</p>
6	<p>Click Edit (at the right of the Use constant list).</p> <p>System response The system displays the Map Constants dialog box.</p> 
7	<p>Click New.</p> <p>System response The system displays the Edit Constant dialog box.</p> 
8	<p>In the ID box, type the literal constant identifier.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
9	From the Type list, select “String.” This is the category of this literal constant.
10	In the Value box, type the value of the literal constant. Note For an inbound map, this is the loop identifier code that you expect to receive from your trading partner.
11	Click OK to add the constant to the system.
12	Select the constant you created from the list. Note The system matches the selected constant against the loop identifier code.
13	Click OK to exit the EDI Segment Properties dialog box.

**Procedure:
defining a LE
segment inbound**

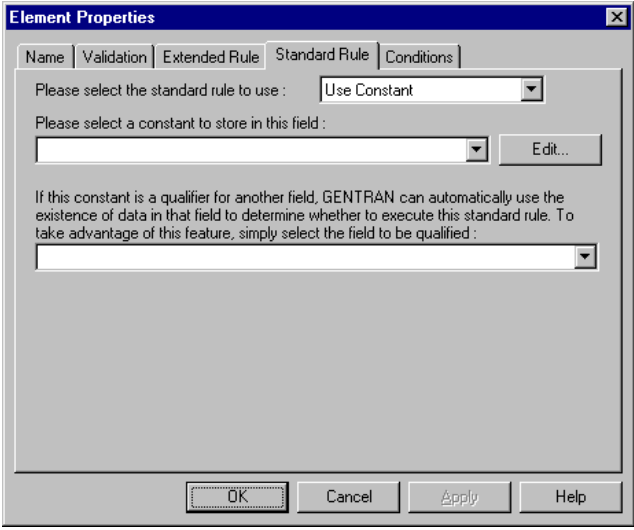
Use this procedure to define an LE segment for an inbound map.

Step	Action
1	Right-click the LE segment and select Properties from the shortcut menu. System response The system displays the EDI Segment Properties dialog box.
2	<ul style="list-style-type: none"> ▶ Select the Looping tab to access the loop options and ▶ verify that the Loop End option is selected.
3	Select the Key Field tab. System response The system displays the key field options.
4	From the Field list, select the “Loop Identifier Code.” Note This list contains all the elements that are contained in the segment and gives you the ability to define the Loop End segment definition, by specifying that the loop identifier code must have the value you specify in the Matching rules section. (Continued on next page)

(Contd) Step	Action
5	<p>Select the constant you created for the LS segment from the list.</p> <p>Notes</p> <ul style="list-style-type: none"> ▶ The system matches the selected constant against the loop identifier code. ▶ The Matching rules section enables you to access all the literal constants and code lists currently defined for this map.
6	Click OK to exit the EDI Segment Properties dialog box.

**Procedure:
defining a LS
segment outbound**

Use this procedure to define an LS segment for an outbound map.

Step	Action
1	<p>Double-click the Loop Identifier Code element in the LS segment.</p> <p>System response The system displays the Element Properties dialog box.</p>
2	Select the Standard Rule tab to access the standard rule options.
3	<p>From the standard rule list, select “Use Constant.”</p> <p>System response The system displays the constant options.</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
4	Click Edit . System response The system displays the Map Constants dialog box.
5	Click New . System response The system displays the Edit Constant dialog box.
6	In the ID box, type the literal constant identifier.
7	From the Type list, select “String.” This is the category of this literal constant.
8	In the Value box, type the value of the literal constant. For an outbound map, this is the loop identifier code that your partner is expecting to receive from you.
9	Click OK to add the constant to the system.
10	Click Close to exit the Map Constants dialog box.
11	From the constant list, select the constant that you created. Note The system matches the selected constant against the loop identifier code.
12	Click OK to exit the Element Properties dialog box.

**Procedure:
defining a LE
segment outbound**

Use this procedure to define an LE segment for an outbound map.

Step	Action
1	Double-click the Loop Identifier Code element in the LE segment. System response The system displays the Element Properties dialog box.
2	Select the Standard Rule tab to access the standard rule options.
3	From the standard rule list, select “Use Constant.” System response The system displays the constant options. (Continued on next page)

(Contd) Step	Action
4	From the constant list, select the constant that you created. Note The system matches the selected constant against the loop identifier code.
5	Click OK to exit the Element Properties dialog box.

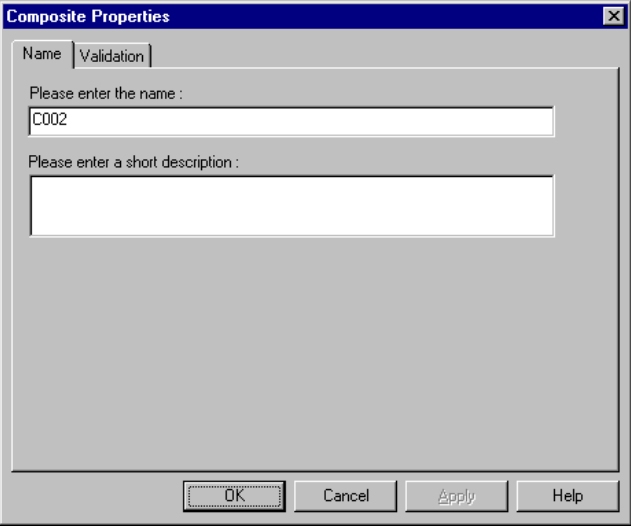
How to Modify Composite Properties

Introduction A composite is a data element that contains two or more component data elements or subelements. You can modify the composite name and description, and whether or not it is mandatory.

Note

Composite elements are only used for EDI maps.

Procedure Use this procedure to modify the properties of a composite.

Step	Action
1	<p>Right-click the composite you want to modify and select Properties from the shortcut menu.</p> <p>System response The system displays the Composite Properties dialog box.</p> 
2	<p>If you want to change the composite name, type the data element identification number in the first box on the Name tab.</p> <p>Note If a composite occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where “n” is the number of the occurrence in the map.</p>
3	<p>If you want to change the composite description, type the new description in the second box on the Name tab.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
4	Select the Validation tab if you want to specify that the composite is mandatory.
5	Select the check box on the Validation tab to indicate whether this composite must be present.
6	Click OK to save the changes to the composite.

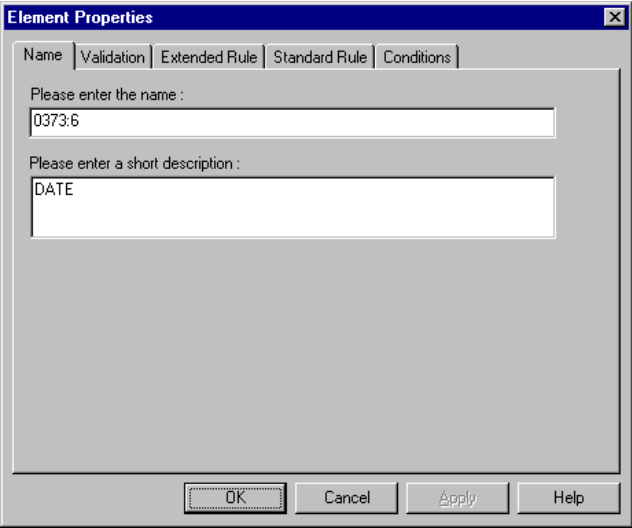
How to Modify Element Properties

Introduction

Each segment or composite contains a group of logically-related elements. These elements define the structure of the EDI data that your system needs to process the document.

Procedure: modifying element properties

Use this procedure to modify the properties of an element.

Step	Action
1	<p>Double-click the element you want to modify.</p> <p>System response The system displays the Element Properties dialog box.</p> 
2	<p>To change the name of the element, type the new name in the first box on the Name tab. The name is typically the element sequence number.</p> <p>Note If an element occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where “n” is the number of the occurrence in the map.</p>
3	<p>To change the description of the element, type the new description in the second box on the Name tab.</p> <p>Note The description is used to provide a brief explanation of the element that allows you to differentiate it from similar elements.</p>
4	<p>Select the Validation tab to access validation options.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
5	If you want to designate the element as “mandatory” (must be present), select the first check box on the Validation tab.
6	<p>To change the minimum length of the element, type the minimum length in the Minimum box.</p> <p>Note If the data is less than the minimum length, a compliance error is generated when translation occurs.</p>
7	To change the maximum length of the element, type the maximum length in the Maximum box.
8	<p>From the Data Type list, select the type of the element.</p> <p>Note Valid values are:</p> <ul style="list-style-type: none"> ▶ String = alpha element ▶ Number = numeric or real element ▶ Date/Time = date or time element ▶ Bin Data = binary data (only available if you select “Binary” on the Special tab of the EDI Segment Properties dialog box) ▶ Bin Length = length of binary data (only available if you select “Binary” on the Special tab of the EDI Segment Properties dialog box) <p>Reference See <i>How to Modify Segment Properties</i> on page 2 - 73 for more information about selecting “Binary” on the Special tab.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
9	<p>From the Format list, select how the data element is formatted.</p> <p>Note The choices for this field depend on the type of field you selected from the Data Type list. If you choose Number or Date/Time in the Data Type box, you can select the data format from the Format list. If you selected String from the Data Type box, you should type a syntax token to denote that this field must be formatted as the specified syntax token dictates.</p> <p>When you installed Gentran:Server, you assigned a default format to the string fields. This format serves as the basis for character validation. Most U.S. users use a default format that corresponds to ASCII characters (for example, the X syntax token). Most users of Asian or European languages and encoded character sets should use the Free Format (0x01-0xFF).</p> <p>Reference See <i>How to Use Syntax Tokens</i> on page 2 - 106 for more information.</p>
10	Click OK to save the changes to the element.

**Procedure:
defining and
modifying
relational
conditions**

You can use relational conditions to connect fields together for syntax or compliance reasons. For example, Field A is invalid unless Field B is present. Therefore, if you set up a condition that pairs Fields A and B, the system generates a compliance error if one of those fields is not present.

You can also view the conditional relationships between elements, as provided by the standard.

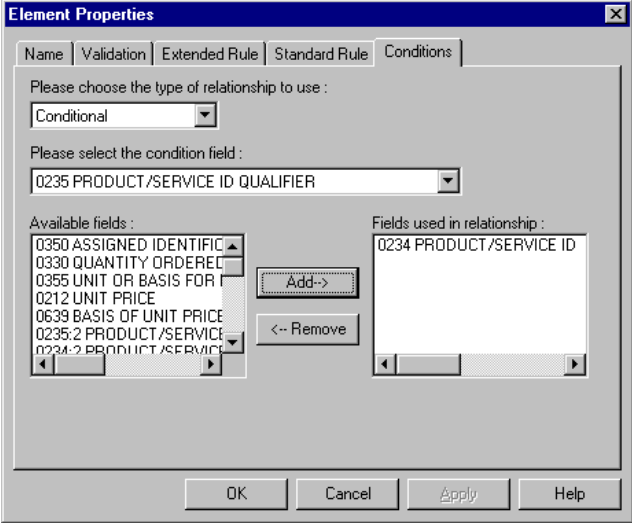
Warning

The system enables you to edit EDI conditional relationships, but if you do, you generates a compliance error. We recommend that you view EDI conditions only.

Procedure

Use this procedure to define, modify, and view field and element relational conditions.

Step	Action
1	<p>Double-click the field or element you want to modify.</p> <p>System response The system displays the Field (or Element) Properties dialog box.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
2	<p>Select the Conditions tab.</p> <p>System response The system displays the relational condition options.</p> 
3	<p>Select the field connection condition from the type of relationship list.</p> <p>Note Valid values are:</p> <ul style="list-style-type: none"> ▶ Paired/Multiple – if any of the specified fields are present then all fields must be present ▶ Required – at least one of the specified fields must be present ▶ Exclusion – no more than one of the specified fields may be present ▶ Conditional – if the first Condition field is present, the rest of the fields must also be present ▶ List Conditional – if the first Condition field is present, at least one of the specified fields must also be present
4	<p>Select the first field from the condition field list.</p> <p>Notes</p> <ul style="list-style-type: none"> ▶ This is the field on which the conditional relationship hinges if you chose “Conditional” or “List Conditional” from the type of relationship list. ▶ The condition field list is only active if you chose either “Conditional” or “List Conditional” from the type of relationship list. <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
5	<p>From the Available fields list, select a field or fields and click Add.</p> <p>System response The system moves the fields to the Fields used in relationship list, to include the fields as a part of the conditional relationship.</p> <p>Notes</p> <ul style="list-style-type: none">▶ The Available fields list contains all the fields in the translation object that are valid to be used in a condition at this point.▶ The Fields used in relationship list contains the fields that you selected to be a part of the conditional relationship.▶ To remove the fields from the conditional relationship, select a field or fields from the Fields used in relationship list and click Remove to move the fields back to the Available fields list.
6	Click OK to add the conditional relationship to the field.

Using a CII File

Overview

Introduction

If a side of your map is CII, you must either define it to Gentran:Server or load a previously-saved file definition. A CII message consists of one message header, one TFD area, and one message trailer

Before you define the CII file, you should determine which groups and TFDs you will need to use. The groups contain related groups and TFDs. This means that you must create groups before you create the subordinate subgroups and TFDs.

Each map component is arranged sequentially in the order that it is most logical for the system to process. Therefore, each level of your application must be created sequentially.

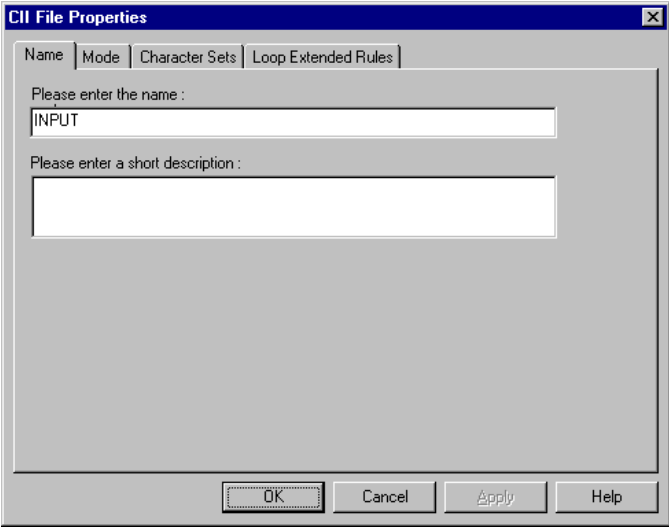
How to Modify CII File Properties

Introduction

The CII File Properties dialog box enables you to modify setting for the CII or CII-Positional sides of a map. This dialog box enables you toggle the use of dividing mode (a method of formatting CII data into fixed-length 251-byte records), non-transparent mode (encodes characters in the CII data so that they do not cause problems with some communications file-transfer protocols). This dialog box also enables you to specify single- or double-byte character sets to use in data translation.

Procedure

Use this procedure to modify CII File Properties.

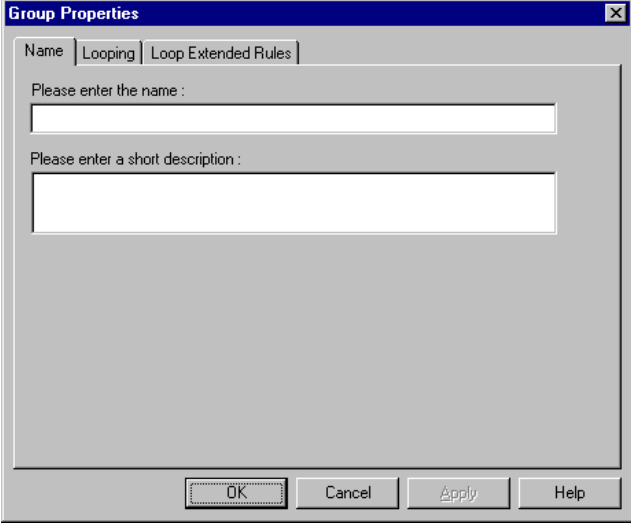
Step	Action
1	<p>Right-click the CII file icon and select Properties from the shortcut menu.</p> <p>System response The system displays the CII File Properties dialog box.</p>  <p>Note The system displays the Name tab by default. On this tab, you can modify the name and description of the CII file.</p>
2	Select the Mode tab to access mode options.
3	<p>Will you be communicating with a transmission system that is unable to process variable length records?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select the dividing mode (a method of formatting CII data into the plural number of 251-byte fixed records) On option. ▶ If <i>no</i>, continue with the next step. <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
4	<p>Do you want to use non-transparent mode to encode characters in the CII data so they do not cause problems with some communications file-transfer protocol communications control characters?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select the Non-Transparent Mode check box. ▶ If <i>no</i>, the default is cleared, which means you are using transparent mode.
5	Select the Character Sets tab to access character set options.
6	<p>Do you want to use single- or double-byte character translation?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select the appropriate option from the 8 bit Character Set and 16 Bit Character Set lists. ▶ If <i>no</i>, continue with the next step
7	<p>Do want to specify an extended rule for this group?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, click the Loop Extended Rules tab. <p>Reference See <i>Defining Extended Rules</i> on page 6 - 12 for more information.</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with the next step.
8	<p>Click OK.</p> <p>System response The system saves your modifications and exits the CII File Properties dialog box.</p>

How to Create a Group

Introduction A group contains related TFDs, groups, or both that repeat in sequence until either the data ends or the maximum number of times that the loop is allowed to repeat is exhausted. If you create a group that is subordinate to another group, this corresponds to a nested looping structure (a loop within a loop).

Procedure Use this procedure to create a group.

Step	Action
1	<ul style="list-style-type: none"> ▶ Highlight the map component which precedes the group you are creating in the file layout. ▶ From the Edit menu select Insert or Create Sub, depending on the level. ▶ Select Group from the submenu. <p>System Response The system displays the Group Properties dialog box (Name tab displayed by default).</p> 
2	<p>In the first box on the Name tab, type the group name.</p> <p>Note Do not use spaces or dashes (-) in the group name. You can use the underscore (_) to separate words.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
3	<p>In the second box on the Name tab, type the loop description.</p> <p>Note This box is used to provide a brief explanation of the loop.</p>
4	<p>Select the Looping tab to access looping options.</p>
5	<p>In the Minimum Usage box, type the minimum number of times the loop must be repeated.</p> <p>Note For a conditional loop, the minimum usage should always be “0” (zero). For a mandatory loop, the minimum usage should be “1” or greater.</p>
6	<p>In the Maximum Usage box, type the maximum number of times the loop can be repeated.</p>
7	<p>Is this a single iteration group?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select Promote records to parent if you want to specify that the subordinate records and groups should be extracted and located in the parent group when the group is compiled. ▶ If <i>no</i>, continue with the next step.
8	<p>Do you want to specify an extended rule for this group?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, select the Loop Extended Rules tab. <p>Reference See <i>Defining Extended Rules</i> on page 6 - 12 for more information.</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with the next step.
9	<p>Click OK to create the group.</p>

How to Create a TFD

Introduction

The first TFD in your CII file is subordinate to the icon on the CII of the map. This graphic representation shows how the TFDs (and groups) are contained within the CII file.

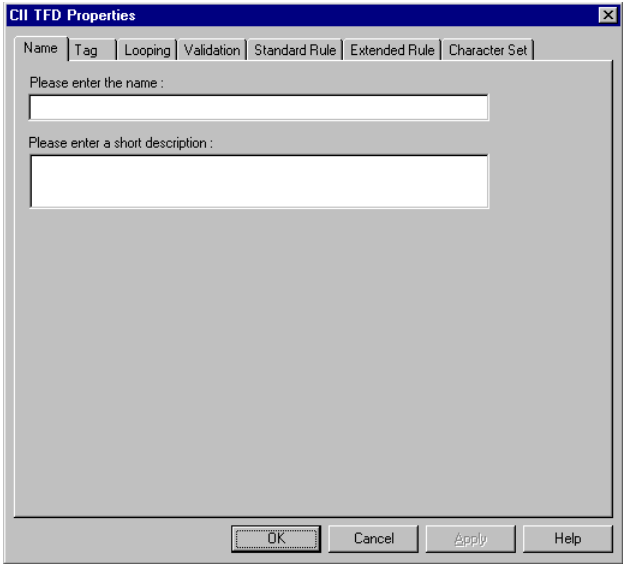
After you create the first TFD, you can define subsequent TFDs at the same level. You can also define groups, if your application includes looping structures. You may need to define a group if your application requires a group of related TFDs (and/or subgroups) that repeat in sequence. Please see *How to Create a Group* on page 2 - 62 for more information.

Note

When you create a TFD, the default data type is “String,” so the system displays the Character Set tab. If you change the TFD data type to “Number” or “Date/Type,” the system removes the Character Set tab.

Procedure

Use this procedure to create a TFD.

Step	Action
1	<p>Are you creating the first TFD in your file?</p> <ul style="list-style-type: none"> • If <i>yes</i>, select the CII File icon (INPUT for an inbound map and OUTPUT for an outbound map), and select Create Sub from the Edit menu. Then, select TFD from the submenu. • If <i>no</i> (you are creating a subsequent TFD), select the TFD which precedes the TFD you are creating in the file layout. Then, select Insert from the Edit menu and select TFD from the submenu. <p>System response The system displays the CII TFD Properties dialog box.</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
2	<p>In the first box on the Name tab, type the TFD name.</p> <p>Warning Do not use spaces or dashes (-) for the TFD name. You can use the underscore () to separate words.</p>
3	<p>In the second box on the Name tab, type a description of the TFD.</p> <p>Note This box is used to provide a brief explanation of the TFD that allows you to differentiate it from similar TFDs.</p>
4	<p>Select the Tag tab to access tag information.</p> <p>Note The data tag is an identifier that expresses attributes of a data element.</p>
5	<p>Select either the Hex or Decimal option, depending on how you want to enter the data tag.</p> <p>Note The default is Hex.</p>
6	<p>If you want to use an extended range of values for the tag, select the Extended Mode check box.</p> <p>Notes</p> <ul style="list-style-type: none"> • If you select the Extended Mode check box, you can enter up to hex 7FFF (decimal 524287) in the Tag box. In compressed mode, you can only use up to hex EF (decimal 239). • The default is cleared, which means that the data tag will be expressed in compressed mode. • In Extended Mode, the range that you can use for the TFD tag is restricted to decimal 61440 - 65535 and hex F000 - FFFF.
7	<p>In the Tag box, type the identification tag. You must enter a tag in this box.</p>
8	<p>Select the Looping tab to access the loop options.</p>
9	<p>Accept the default of Normal, unless you want to indicate that the segment is the beginning of a loop (Loop Start), the end of a loop (Loop End), or that the loop repeats (Loop Repeat).</p> <p>Note If you select Loop Start, the system displays the Key Field tab.</p>
10	<p>Select the Validation tab to access validation options.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
11	If you want to designate the TFD as “mandatory” (must be present), select the first check box on the Validation tab.
12	<p>To change the minimum length of the TFD, type the minimum length in the Minimum box.</p> <p>Note If the data is less than the minimum length, a compliance error is generated when translation occurs.</p>
13	To change the maximum length of the TFD, type the maximum length in the Maximum box.
14	<p>From the Data Type list, select the type of the TFD.</p> <p>Note Valid values are:</p> <ul style="list-style-type: none"> ▶ String = alphabetic ▶ Number = numeric or real ▶ Date/Time = date or time
15	<p>From the Format list, select how the TFD is formatted.</p> <p>Note The choices for this field depend on the type of field you selected from the Data Type list. If you choose Number or Date/Time in the Data Type box, you can select the data format from the Format list. If you selected String from the Data Type box, you should type a syntax token to denote that this field must be formatted as the specified syntax token dictates (the default syntax token is “X”).</p> <p>When you installed Gentran:Server, you assigned a default format to the string fields. This format serves as the basis for character validation. Most U.S. users use a default format that corresponds to ASCII characters (for example, the X syntax token). Most users of Asian or European languages and encoded character sets should use the Free Format (0x01-0xFF).</p> <p>Reference See <i>How to Use Syntax Tokens</i> on page 2 - 106 for more information.</p>
16	<p>If the TFD data-type is Number or String and you want the data read/written in binary format:</p> <ul style="list-style-type: none"> ▶ Select the raw bytes check box, ▶ From the Width list (Number format only), select how many bytes should be used to represent the number, ▶ Select either the little-endian or big-endian option to specify in which order the bytes should be read. <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
17	If the Character Set tab is available, it enables you to set the single-double-byte character translation for this string. The default character set is 8-bit. Select the appropriate option from the 8 bit Character Set and 16 Bit Character Set lists.
18	Click OK to create the TFD.

Formatting Your Data

Overview

Introduction

When you define or modify a field, element, or TFD, you must specify the type and format. The options that are available for the format of the field, element, or TFD depend on which type you select (string, number, or Date/Time).

How to Use the String Type

Introduction

A string-type field or element contains one or more printable characters. If you specify that a field or element is a string type, you must designate the format by specifying a syntax token.

Syntax tokens enable you to designate a “token” that defines ranges of characters and numbers that are allowed to be used for a string-type element or field. You can then use the syntax tokens in the Format field of the Field Properties dialog box. This enables you to define the type of characters to be used while compliance checking each element/field (e.g., alphanumeric within a certain range or numeric within a certain range).

Note

When you set up a token, it applies only to that map. You may need to set one up for each map that you create.

Character ranges

Gentran:Server uses the ANSI Character Set when determining the start-end range.

For Western European languages, consult the ANSI character chart (1252 Windows Latin 1). This chart also displays ranges so that you can enter appropriate ranges for the characters in your language. If no chart is available, use the following guidelines:

- To include all the accented characters in the major languages of Western Europe, add the following ranges:

Start	End
0xC0	0xD6
0xD8	0xF6
0xF8	0xFC

- Scandinavian users should also add the following in order to include Æ and œ.

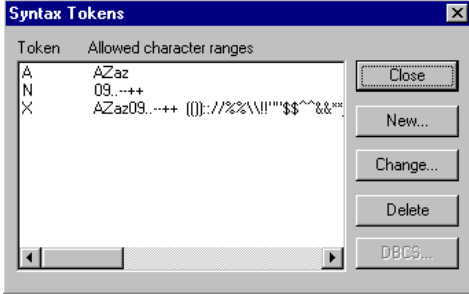

Start	End
0x8C	0x8C
0x9C	0x9C

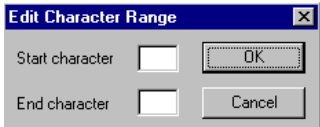
- To include the euro symbol (€), use the following.

Start	End
0x80	0x80

**Procedure:
creating or editing
a syntax token
(Western
European
languages)**

Use this procedure to create a syntax token.

Step	Action
1	<p>From the Edit menu, select Syntax Tokens.</p> <p>System response The system displays the Syntax Tokens dialog box.</p>  <p>Notes</p> <ul style="list-style-type: none"> ▶ The Token column contains the value designated as the syntax token for each existing syntax token. The Token contains a range of characters that, when applied to an element, dictate the way that element must be formatted. If the element is not formatted as specified, the system generates a compliance error. ▶ The Allowed character ranges column contains the ranges of characters that are permitted for each existing syntax token. Each range consists of a pair of characters that define the start and end characters. ▶ DBCS is only available if you are executing a double-byte version of Windows 2000.
2	<p>Are you creating a new syntax token?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, click New. ▶ If <i>no</i>, select the token you want to edit and click Edit. <p>System response The system displays the Edit Syntax Token dialog box.</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
3	<p>In the Token box, type the unique one-character alphanumeric value that the system recognizes as containing the allowed range of characters you designate.</p> <p>Notes</p> <ul style="list-style-type: none"> ▶ The Token can only be one unique character, upper- or lower-case alphabetic or numeric “1” - “9”. ▶ The Character Ranges list contains the character range or ranges that you define for this token. You can define more than one character range for each token. For example, you can define the token “A” as allowing the character range “A” through “Z” <i>and</i> the character range “a” through “z”. This indicates that token “A” only allows upper-case <i>and</i> lower-case alphabets.
4	<p>Do you want to create a new character range?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, click New. <p>System response The system displays the Edit Character Range dialog box.</p>  <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with step 8.
5	<p>In the Start character box, type the character that begins the allowed token range.</p> <p>Example For example, if the character range you want to define is “B” through “D,” type “B” in the Start character box.</p> <p>Notes</p> <ul style="list-style-type: none"> ▶ If you type a character, such as é, that is not accepted, you need to enter it in hex code. To enter hex characters, “0(zero)x” or “0X,” followed by the hex code. For example, the hex equivalent of é is 0xE9. ▶ The Start character and End character can only be one (1) character, upper- or lower-case alphabetic or numeric “1” - “9.”
6	<p>In the End Character box, type the character that terminates the allowed token range. Use the guidelines from step 5.</p>
7	<p>Click OK to return to the Edit Syntax Tokens dialog box.</p>
8	<p>To add additional character ranges to the syntax token, repeat steps 4 - 7 as many times as necessary.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
9	Click OK to save the syntax token and return to the Syntax Tokens dialog box.
10	Click Close to exit the Syntax Tokens dialog box.

Procedure.
creating or editing
a syntax token
(Far Eastern
languages)

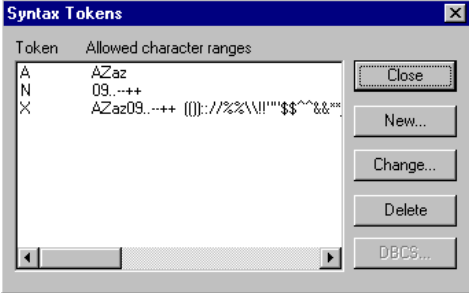
The DBCS syntax token function enables you to create a map that accepts double-byte characters. If you are running Gentran:Server on a Chinese, Japanese, or Korean version of Windows/Windows 2000 or Windows 98/Me, the **DBCS** button on the Syntax Tokens dialog box is available.

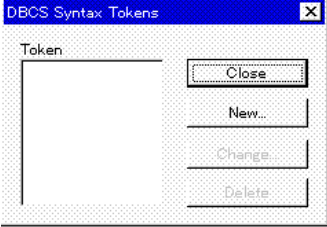
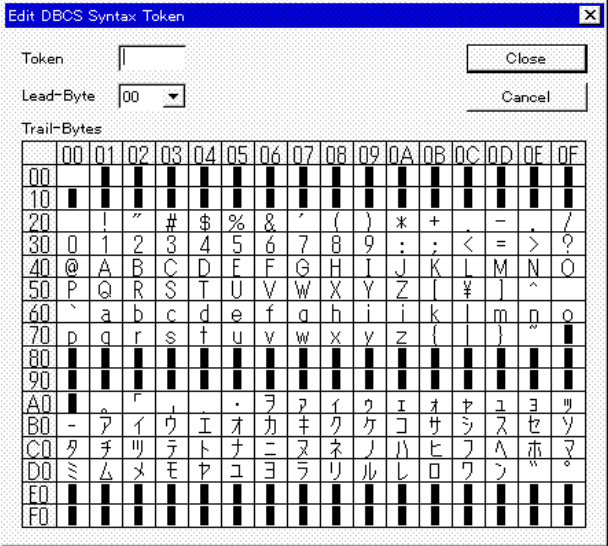
DBCS tokens are displayed only in the DBCS Syntax Tokens dialog box, not in the list in the Syntax Tokens dialog box.

Note

When you set up a DBCS token, it applies only to that map. You may need to set one up for each map that you create.

Use this procedure to create a DBCS syntax token.

Step	Action
1	<p>From the Edit menu, select Syntax Tokens.</p> <p>System response The system displays the Syntax Tokens dialog box.</p>  <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
2	<p>Click DCBS.</p> <p>System response The system displays the DCBS Syntax Token dialog box.</p> 
3	<p>Click New.</p> <p>System response The system displays the Edit DCBS Syntax Token dialog box.</p>  <p>Note Double-byte characters are composed of a lead byte and a trail byte. In the above example, the character A is code point 0041.</p>
4	<p>In the Token box, type the unique one-character alphanumeric value that the system recognizes as containing the allowed range of characters you designate.</p>
5	<p>Do you want to include all characters in the Trail-Bytes section of the dialog box?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, continue with step 7. ▶ If <i>no</i>, continue with step 6. <p style="text-align: right;">(Continued on next page)</p>



(Contd) Step	Action
6	<ul style="list-style-type: none">▶ Use the Lead-Byte list to view other characters in the code page on your system.▶ To exclude individual characters or groups of characters from the token, select them (grey them out).
7	Click Close to save the syntax token and return to the Syntax Tokens dialog box.
8	Click Close to exit the Syntax Tokens dialog box.

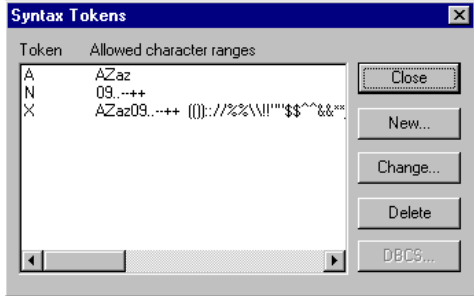
How to Delete a Syntax Token

Introduction

You can delete a syntax token from the system. You can also delete a specific character range from a syntax token.

**Procedure:
deleting a syntax
token**

Use this procedure to delete a syntax token.

Step	Action
1	<p>From the Edit menu, select Syntax Tokens.</p> <p>System response The system displays the Syntax Tokens dialog box.</p> 
2	<p>Select the token that you want to delete and click Delete.</p> <p>System response The selected entry is deleted without warning.</p>

**Procedure:
deleting a
character range**

Use this procedure to delete a character range from a syntax token.

Step	Action
1	<p>From the Edit menu, select Syntax Tokens.</p> <p>System response The system displays the Syntax Tokens dialog box.</p>
2	<p>Select a syntax token and click Change.</p> <p>System response The system displays the Edit Syntax Tokens dialog box.</p>
3	<p>Select the character range that you want to delete and click Delete.</p>



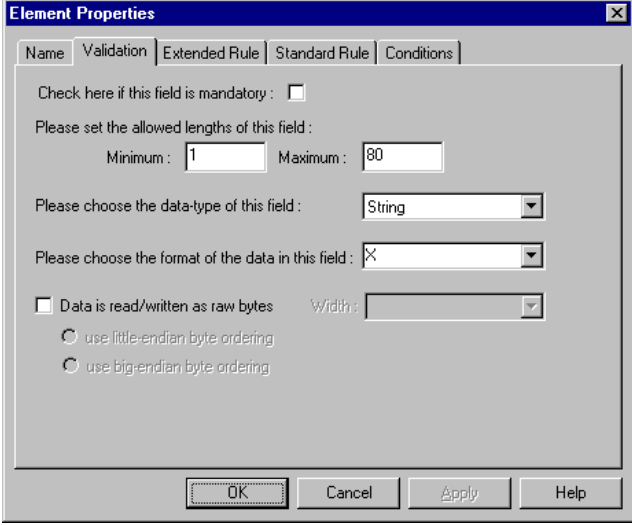
How to Use Syntax Tokens

Introduction

You can use syntax tokens in the Format box of the Field (or Element or CII Record) Properties dialog box. This enables you to designate a “token” that defines ranges of characters and/or numbers that are allowed to be used for a string-type field, element, or TFD. The use of a syntax token enables you to define what characters should be used while compliance checking this field, element, or TFD (i.e., alphanumeric within a certain range, numeric within a certain range, etc.).

Procedure

Use this procedure to use syntax tokens.

Step	Action
1	<p>Double-click an existing field, element, or TFD, or create a new one.</p> <p>Reference See <i>Using a Fixed-Format File</i> on page 2 - 51, <i>Using an EDI File</i> on page 2 - 68, or <i>Using a CII File</i> on page 2 - 89 for more information about creating fields, elements, or TFDs.</p> <p>System response The system displays the Element (or Field or CII Record) Properties dialog box.</p>
2	<p>Select the Validation tab.</p> <p>System response The system displays the validation options.</p>  <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
3	<p>From the data-type list, select String.</p> <p>Note This indicates that the field, element, or TFD contains characters.</p>
4	<p>From the data format list, select free format or a predefined syntax token to denote that this field, element, or TFD must be formatted as the specified syntax token dictates.</p> <p>Notes</p> <ul style="list-style-type: none"> ▶ When you installed Gentran:Server, you assigned a default format to the string fields. This format serves as the basis for character validation. Most U.S. users use a default format that corresponds to ASCII characters (for example, the X syntax token). Most users of Asian or European languages and encoded character sets should use the Free Format (0x01-0xFF). ▶ Free Format indicates that any characters are acceptable in the field. The translator does not check the characters for compliance. <p>Reference See <i>Procedure: creating or editing a syntax token (Western European languages)</i> on page 2 - 100 for more information.</p>
5	<p>Click OK to exit the Element (or Field or CII Record) Properties dialog box.</p>



How to Use the Number Type

Introduction

A number type field, element, or TFD contains either an implied decimal or real number that can be mathematically manipulated. If you specify that a field, element, or TFD is a number type, you must designate the format by specifying a format of either “N” (implied decimal) or “R” (real) and the number of decimal places.

Real or numeric

An N-formatted number has an implied decimal point (for example, 2.01 formatted as N2 is 201). An R-formatted number has an explicit decimal point and truncates trailing zeros (for example, 2.123 formatted as R2 is 2.12 and 3.10 formatted as R2 is 3.1). Whether you use the N or R format depends on the requirements of the document. Regardless of whether you use the N or R format, you must also indicate the number of decimal places in the field.

This lists the available real and numeric options.

- R0 (this field/element/TFD has an explicit decimal point and no decimal places)
- R1 (this field/element/TFD has an explicit decimal point and up to one decimal place)
- R2 (this field/element/TFD has an explicit decimal point and up to two decimal places)
- R3 (this field/element/TFD has an explicit decimal point and up to three decimal places)
- R4 (this field/element/TFD has an explicit decimal point and up to four decimal places)
- R5 (this field/element/TFD has an explicit decimal point and up to five decimal places)
- R6 (this field/element/TFD has an explicit decimal point and up to six decimal places)
- R7 (this field/element/TFD has an explicit decimal point and up to seven decimal places)
- R8 (this field/element/TFD has an explicit decimal point and up to eight decimal places)
- R9 (this field/element/TFD has an explicit decimal point and up to nine decimal places)
- N0 (this field/element/TFD has an implied decimal point and no decimal places)
- N1 (this field/element/TFD has an implied decimal point and up to one decimal place)
- N2 (this field/element/TFD has an implied decimal point and up to two decimal places)
- N3 (this field/element/TFD has an implied decimal point and up to three decimal places)
- N4 (this field/element/TFD has an implied decimal point and up to four decimal places)

(Continued on next page)

Real or numeric (contd)

- N5 (this field/element/TFD has an implied decimal point and up to five decimal places)
- N6 (this field/element/TFD has an implied decimal point and up to six decimal places)
- N7 (this field/element/TFD has an implied decimal point and up to seven decimal places)
- N8 (this field/element/TFD has an implied decimal point and up to eight decimal places)
- N9 (this field/element/TFD has an implied decimal point and up to nine decimal places)

Note

If you select an implicit decimal (“N” format) for a field and the data in that field has less than the specified number of decimal places, the translator pads the data with zeroes to the left so that it still interprets the data within the specified format. For example, if you specify a format of “N3” for a field, and the data in that field is “1”, the translator interprets the data as “.001”.

Procedure

Use this procedure to use the number type.

Step	Action
1	Double-click an existing field, element, or TFD, or create a new one. Reference See <i>Using a Fixed-Format File</i> on page 2 - 51, <i>Using an EDI File</i> on page 2 - 68, or <i>Using a CII File</i> on page 2 - 89 for more information about creating fields, elements, or TFDs. System response The system displays the Element (or Field or CII Record) Properties dialog box.
2	Select the Validation tab. System response The system displays the validation options.
3	From the data-type list, select Number. Note This indicates that the field, element, or TFD is a numeric or real number that can be mathematically manipulated.
4	From the data format list, select the appropriate real or numeric option.
5	Click OK to exit the Element (or Field or CII Record) Properties dialog box.



How to Use the Date/Time Type

Introduction

A Date/Time type field, element, or TFD contains a date or time. If you specify that a field, element, or TFD is a Date/Time type, you must specify exactly how the date or time must be formatted.

Note

Additional date and time formats can be created and added to the Map Editor.

Date/time formats

This table lists the valid date and time formats.

Format	Description
YYMMDD	Two-digit year, two-digit month, two-digit day
MMDDYY	Two-digit month, two-digit day, last two digits of year (example: 121599)
YYYYMMDD	Four-digit year, two-digit month, two-digit day (example: 19991215)
DDMMYYYY	Two-digit day, two-digit month, four-digit year (example: 15121999)
MMDDYYYY	Two-digit month, two-digit day, four-digit year (example: 12151999)
DDMMYY	Two-digit day, two-digit month, last two digits of year (example: 151299)
YYMMMDD	Last two digits of year, three-letter abbreviation of the month, two-digit day (example: 99JAN02)
DDMMYY	Two-digit day, three-letter abbreviation of the month, last two digits of year (example: 02JAN99)
MMMDDYY	Three-letter abbreviation of the month, two-digit day, last two digits of year (example: JAN0299)
YYYYMMMDD	Four-digit year, three-letter abbreviation of the month, two-digit day (example: 2003JUL04)
DDMMYYYY	Two-digit day, three-letter abbreviation of the month, four-digit year (example: 04JUL2003)
MMMDDYYYY	Three-letter abbreviation of the month, two-digit day, four-digit year (example: JUL042003)

(Continued on next page)

(Contd) Format	Description
YYDDD	Last two digits of year, three-digit Julian day (example: 99349 for the 349th day of 1999)
DDDY	Three-digit Julian day, last two digits of year (example: 34999)
YYYYDDD	Four-digit year, three-digit Julian day (example: 1999349)
DDYYYY	Three-digit Julian day, four-digit year (example: 3491999)
YY/MM/DD	Last two digits of year, separator, two-digit month, separator, two-digit day (example: 99/12/05)
DD/MM/YY	Two-digit day, separator, two-digit month, separator, last two digits of year (example: 05/12/99)
MM/DD/YY	Two-digit month, separator, two-digit day, separator, last two digits of year (example: 12/15/99)
YYYY/MM/DD	Four-digit year, separator, two-digit month, separator, two-digit day (example: 1999/12/15)
DD/MM/YYYY	Two-digit day, separator, two-digit month, separator, four-digit year (example: 15/12/1999)
MM/DD/YYYY	Two-digit month, separator, two-digit day, separator, four-digit year (example: 12/15/1999)
YY/MMM/DD	Two-digit year, separator, three-letter abbreviation of the month, separator, two-digit day (example: 99/JUL/20)
DD/MMM/YY	Two-digit day, separator, three-letter abbreviation of the month, separator, two-digit year (example: 20/JUL/99)
MMM/DD/YY	Three-letter abbreviation of the month, separator, two-digit day, separator, two-digit year (example: JUL/20/99)
YYYY/MMM/DD	Four-digit year, separator, three-letter abbreviation of the month, separator, two-digit day (example: 2003/JUL/25)
DD/MMM/YYYY	Two-digit day, separator, three-letter abbreviation of the month, separator, four-digit year (example: 25/JUL/2003)
MMM/DD/YYYY	Three-letter abbreviation of the month, separator, two-digit day, separator, four-digit year (example: JUL/25/2003)
YY/DDD	Last two digits of year, separator, three-digit Julian day (example: 99/349)
DDD/YY	Three-digit Julian day, separator, last two digits of year (example: 349/99)

(Continued on next page)

(Contd) Format	Description
YYYY/DDD	Four-digit year, separator, three-digit Julian day (example: 1999/349)
DDD/YYYY	Three-digit Julian day, separator, four-digit year (example: 349/1999)
MONTH	Month (example: December)
DAY	Day of the week (example: Friday)
HHMM	Two-digit hour, two-digit minutes (example: 0330 for 30 minutes past 3 o'clock)
HHMMSS	Two-digit hour, two-digit minutes, two-digit seconds (example: 033045 for 30 minutes and 45 seconds past 3 o'clock)
HH:MM	Two-digit hour, separator, two-digit minutes (example: 03:30)
HH:MM:SS	Two-digit hour, separator, two-digit minutes, separator, two-digit seconds (example: 03:30:45)
ISO-8601	YYYYMMDDTHHMMSS.mmmZ format: Four-digit year, two-digit month, two-digit day, T (time) indicator, two-digit hour, two-digit minutes, two-digit seconds in Universal Time (also called Zulu Time or Greenwich Mean Time), Z (Zulu time) indicator (example: 20031209T123000.000Z)
YYYYMMDDZ	ISO-8601 date format: Four-digit year, two-digit month, two-digit day, Z (Zulu time) indicator (example: 20031209Z)
MM/DD/YY HH:MM:SS	Two-digit month, separator, two-digit day, separator, last two digits of year, two-digit hour, separator, two-digit minutes, separator, two-digit seconds (example: 12/15/99 03:30:45)
YYMMDD HHMMSS	Last two digits of year, two-digit month, two-digit day, two-digit hour, two-digit minutes, two-digit seconds (example: 991025 033045)
YYYY-MM-DDTHH:MM:SS	Four-digit year, separator, two-digit month, separator, two-digit day, T represents a blank separator, two-digit hour, separator, two-digit minutes, separator, two-digit seconds (example: 2002-02-02 03:30:45)
YYYY-MM-DD	Four-digit year, separator, two-digit month, separator, two-digit day (example: 2002-02-02)
YYYY-MM	Four-digit year, separator, two-digit month (example: 2002-02)
YYYY	Four-digit year (example: 2002)

(Continued on next page)

(Contd) Format	Description
--MM-DD	Two dashes, two-digit month, separator, two-digit day (example: -12-02)
---DD	Three dashes, two-digit day (example: ---02)

Procedure

Use this procedure to use the Date/Time type.

Step	Action
1	Double-click an existing field, element, or TFD, or create a new one. Reference See <i>Using a Fixed-Format File</i> on page 2 - 51, <i>Using an EDI File</i> on page 2 - 68, or <i>Using a CII File</i> on page 2 - 89 for more information about creating fields, elements, or TFDs. System response The system displays the Element (or Field or CII Record) Properties dialog box.
2	Select the Validation tab. System response The system displays the validation options.
3	From the data-type list, select Date/Time. Note This indicates that the field, element, or TFD is a date or time.
4	From the data format list, select the appropriate date or time option.
5	Click OK to exit the Element (or Field or CII Record) Properties dialog box.

Completing a Map

How to Create Simple Links

Introduction

The Link function enables you to map a field, element, or TFD from the Input side of the map to a field, element, or TFD on the Output side of the map. The link between two map components (hereafter referred to as fields) is visually represented with a line connecting the two fields.


If you need to link two input fields to the same output field because of conditions established in your map, you must use an extended rule.

Note

You must have at least one direct link (using the Link function) to every record or segment on the Output side of the map, so the translator can create the record/segment.

Procedure

Use this procedure to link two fields.

Step	Action
1	From the Functions menu, select Link .
2	<p>Select a field on the Input side of the map (or TFD if the Input side is CII).</p> <p>System response The cursor changes to a “Link Arrow”:</p> 
3	<p>Select the field on the Output side of the map (or TFD if the Output side is CII) that the data from the field selected on the Input side of the map is mapped to.</p> <p>System response A line connects the two fields to visually represent the link.</p> <p>Note If a line does not appear when the two fields are linked, check the Preferences dialog box.</p> <p>Reference See <i>How to Customize Global Display of Links</i> on page 2 - 13 for more information. On the Links tab of the Preferences dialog box, you should specify “Show all links possible.”</p>

How to Map Binary Segments

Introduction

Certain EDI standards use binary segments and Gentran:Server enables you to create and configure map components to process binary data.

When you load a EDI standard that contains binary segments from the Standards CD supplied with Gentran:Server, your system automatically creates the required elements. However, if you need to manually create binary segments, use the procedures in this topic.

Example

When you load the ANSI X12 275 transaction set into the Output side of a map, you will notice that it contains a binary data segment with two elements—one for the length of the binary data and one for binary data. The default data type for the length element is “Number” and the maximum length is set to “15.” The default data type for the binary data element is “String” and the default length is “760.”

Required segments and elements

To handle binary data received for an inbound EDI segment, you must create or configure regular segments and elements on the Input side of your map.

To generate an EDI segment containing binary data, you must create a binary data segment on the Output side of your map. Then, you must create two elements in the binary segment for these data types:

- Bin_Len, which is for the length (in characters) of the binary segment
- Bin_Data, which is for the binary data

This table summarizes the input and output elements of your map for binary data

Input Data Type	Link Status	Output Data Type
Number	Not Linked	Bin_Len
String	Linked	Bin_Data

Setting up the Input side

Use this procedure to create the components on the Input side of the map.

Step	Action
1	On the Input side of the map, create or select a segment for the data.
2	<p>Right-click the component and select Properties from the shortcut menu.</p> <p>System Response The system displays the Properties dialog box for the component.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
3	Set the component properties using the dialog box tabs. Note You must complete the Name tab and Looping tab.
4	Add a new element or field to the component you just created.
5	Right-click the element or field and select Properties from the shortcut menu. System Response The system displays the Properties dialog box for the component.
6	Set the component properties using the dialog box tabs. Note On the Validation tab, you must specify Number as the data type.
7	Add a second new element or field to the segment or record.
8	Right-click the component and select Properties from the shortcut menu.
9	Set the component properties using the dialog box tabs. Note On the Validation tab, you must specify String as the data type.

Setting up the Output side

Use this procedure to create the components on the Output side of the map.

Step	Action
1	On the Output side of the map, create or select a segment or record for the binary data.
2	Right-click the component and select Properties from the shortcut menu. System Response The system displays the Properties dialog box for the component.
3	Set the component properties using the dialog box tabs. Note On the Special tab, click the Binary check box.
4	Add a new element or field to the binary component or select the existing element or field. <i>(Continued on next page)</i>

Step	Action
5	<p>Right-click the component and select Properties from the shortcut menu.</p> <p>System Response The system displays the Properties dialog box.</p>
6	<p>Set the component properties using the dialog box tabs.</p> <p>Note On the Validation tab, you must specify Bin_Len as the data type.</p>
7	<p>Add a second new element or field to the segment or select the existing element or field.</p>
8	<p>Right-click the component and select Properties from the shortcut menu.</p>
9	<p>Set the component properties using the dialog box tabs.</p> <p>Note On the Validation tab, you must specify Bin_data as the data type.</p>
10	<p>Link the components on the input side of the map to the components on the output side of the map.</p> <p>Important You must link the string field on the input side of the map (the field created to contain the file name of the binary data file) to the bin_data field on the output side of the map.</p>
11	<p>Complete the map and compile it to create the translation object.</p>

How to Set the Auto-register Option

Introduction The auto-register option allows you to automatically register a translation object when you compile it.

Procedure Use this procedure to set the auto-register option.

Step	Action
1	From the Options menu, select Preferences . System response The system displays the Preferences dialog box.
2	Select the Files tab. System response The system displays the Files preferences. <div data-bbox="654 905 1411 1640" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> </div>
3	Select the Automatically register template after compilation box. (Continued on next page)

(Contd) Step	Action
4	Enter (or Browser and select) the path name for the location of the maps. Warning If you are on a client in a distributed environment, the path name for the map location must be a network path on the primary server. If not, you will receive an error when you compile.
5	Click OK to exit the dialog box.

How to Compile a Map

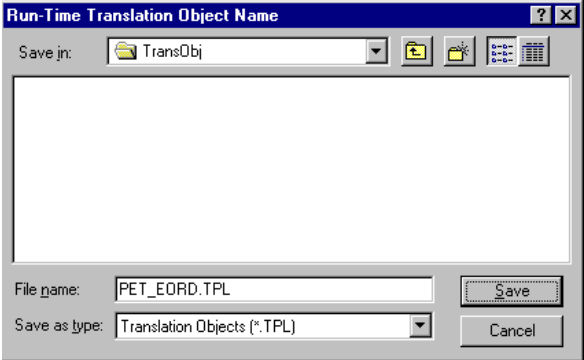
Introduction

The Compile function compiles the map and creates a translation object. The map that you create using Gentran:Server is a *source map*. When that source map is compiled, the result is a *compiled translation object*.

This translation object needs to be registered with the Gentran:Server system before you can use it. You use the Compile function after the map is completed and saved.

Procedure

Use this procedure to compile a map and generate a translation object.

Step	Action
1	<p>From the File menu, select Compile.</p> <p>System response The system displays the Run-Time Translation Object Name dialog box.</p>  <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
2	<p>In the File name box, type the name of the translation object.</p> <p>Note The name of the translation object is 1-8 characters. Use the default .TPL file extension.</p> <p>Recommendation Name the translation object (.TPL file) the same file name (one to eight characters long) as you named the map (.MAP file). Preserving the same file name (with different file extensions) means that the relationship between the source map and the compiled translation object remains evident.</p> <p>Example If the source map name is MWT_850.MAP, the compiled translation object name is MWT_850.TPL.</p> <p>Warning Be very careful not to overlay the source map with the compiled translation object. Use the .TPL file extension to distinguish the translation object.</p>
3	<p>From the Save in list, change the drive\folder where the compiled translation object is stored, if necessary.</p> <p>Warning Do <i>not</i> store the compiled translation object in the GENSRVNT\RegTransObj folder. This folder is reserved for storing a copy of each translation object you register with Gentran:Server.</p>
4	<p>Click Save.</p> <p>System response The system compiles the map and generate a translation object. The system displays the Compile Error dialog box.</p>
5	<p>Verify that no errors occurred and click OK to exit the dialog box.</p> <p>Note The date that the translation object was compiled on is automatically loaded into the “Compiled on” box on the Translation Object Details dialog box.</p>
6	<p>From the File menu, select Save to save the source map with the Compiled on date.</p> <p>System response If you are using the auto-register option, the map will automatically be registered after it compiles and a confirmation dialog box will display. If you are not using the auto-register option, continue with Step 7.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
7	<p>The translation object needs to be registered with the <i>Gentran:Server</i> system before you can use it.</p> <p>Reference See How to Register a New Translation Object in the <i>Gentran:Server User's Guide</i> for more information.</p>

How to Compile Maps Using the Command Line

Introduction The Gentran:Server Application Integration subsystem (MAPPER.EXE) enables you to automatically compile a single map (or maps located in a specified directory) from the command line.

Note

The compiled translation objects are written to same directory as the map source files.

Syntax The command line syntax (from the GENSRVNT\Bin subdirectory) is as follows:

```
mapper.exe -c [mapSourceFile]
```

Note

The [mapSourceFile] is either a single map or a map file name that contains a wildcard character (*). Using the wildcard character causes MAPPER.EXE to compile all maps in the designated directory with that file name pattern. MAPPER.EXE does not recurse through subdirectories.

Example 1 `mapper.exe -c c:\GENSRVNT\tutorial*.map`

Note

In this example, all maps in the tutorial directory are compiled.

Example 2 `mapper.exe -c c:\GENSRVNT\tutorial\pet*.map`

Note

In this example, all maps with names beginning with “pet” in the tutorial directory are compiled.

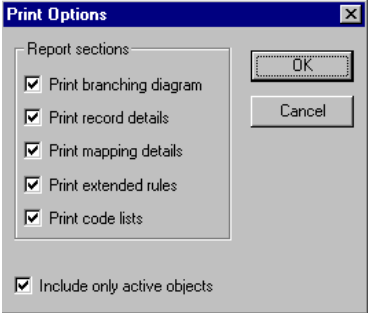
How to Print a Mapping Report

Introduction The Print function enables you to print a mapping report for the current map.

Note

You can print a mapping report to a file by selecting Print to File from the File menu. Change the defaults on the Print Options dialog box and click **OK** to display the Print to File dialog box. Type the name of the file to which you want the report printed and click **OK**.

Use this procedure to print a mapping report.

Step	Action
1	<p>From the File menu, select Print.</p> <p>System response The system displays the Print Options dialog box.</p> 
2	<ul style="list-style-type: none"> ▶ Select the reports that you want by clicking the appropriate check boxes in the Report section, and ▶ click OK <p>System response The system displays the Print dialog box.</p>
3	<p>Set the appropriate options and click OK if you do <i>not</i> need to change Setup options.</p> <p>System response The system prints the mapping report.</p>

1.

Using Standard Rules

Contents	<ul style="list-style-type: none"> ▶ Introduction 3 - 3
	Using the Select Standard Rule 3 - 4
	<ul style="list-style-type: none"> ▶ Overview 3 - 4 ▶ How to Use Information from the Partner Definition 3 - 9 ▶ How to Use Information from a Cross-reference Table 3 - 11 ▶ How to Use Information from Location Tables 3 - 14 ▶ How to Use Information from Lookup Tables 3 - 16
	Using the Update Standard Rule 3 - 17
	<ul style="list-style-type: none"> ▶ Overview 3 - 17 ▶ How to Set Up the Document Name and Reference 3 - 19
	Using the System Variable Standard Rule 3 - 21
	<ul style="list-style-type: none"> ▶ Overview 3 - 21 ▶ How to Use the System Date and Time 3 - 22
	Using the Use Constant Standard Rule 3 - 23
	<ul style="list-style-type: none"> ▶ Overview 3 - 23 ▶ How to Use a Constant in a Map 3 - 27 ▶ How to Define a Qualifying Relationship 3 - 28 ▶ How to Define and Edit Literal Constants 3 - 30 ▶ How to Delete Literal Constants 3 - 31 ▶ How to Map Literal Constants 3 - 32 ▶ How to Generate Qualifiers 3 - 33
	Using the Loop Count Standard Rule 3 - 34
	<ul style="list-style-type: none"> ▶ Overview 3 - 34 ▶ How to Use the Loop Count Function 3 - 35
	Using the Use Accumulator Standard Rule 3 - 36
	<ul style="list-style-type: none"> ▶ Overview 3 - 36 ▶ How to Count Line Items 3 - 42 ▶ How to Calculate Hash Totals 3 - 45 ▶ How to Reset and Calculate a Value Total 3 - 48

Using the Use Code Standard Rule	3 - 56
▶ Overview	3 - 56
▶ How to Validate Data Against a Code List Table	3 - 61
▶ How to Define and Modify a Code List	3 - 62
▶ How to Delete a Code List or Code List Entry	3 - 63
▶ How to Import a Code List	3 - 64
▶ How to Export a Code List	3 - 65
▶ How to Load a Code List Table from the Standard	3 - 66
▶ How to Copy and Paste Code Lists	3 - 67
▶ How to Validate Data Against Code List Tables	3 - 68
▶ How to Map Code Item Descriptions	3 - 69
▶	

Introduction

What is a standard rule?

The Gentran:Server Application Integration subsystem provides you with standard rules that you can apply to fields and elements. Standard rules give you access to functions that are necessary for mapping operations that are more complex than simple linking, but less involved than extended rules.

Each of the standard rules are mutually exclusive (you can use only one on a particular field, element, or TFD).

Note

When an element contains a standard rule a black asterisk appears to the right of the element icon.

Related topic

Extended rules enable you to use a Gentran:Server proprietary programming language to perform virtually any mapping operation you require.

References

See *How to Create Simple Links* on page 2 - 114 and *Introducing Extended Rules* on page 6 - 4 for more information.

Using the Select Standard Rule

Overview

Introduction

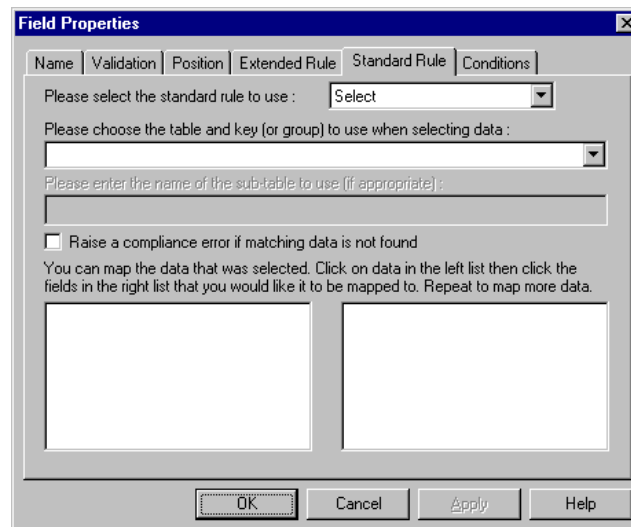
The Select function enables you to select entries from a location table, cross-reference table, partner table, or lookup table (all tables created in the Gentran:Server Partner Editor). You can then map the fields in those tables to one or more fields in the data. The Select function uses the value of the current field, element, or TFD to perform the selection.

The Select function allows you to use information from the Gentran:Server Partner Editor in your maps. You can map information from your partner's profile or Internal System Partner in the Partner Editor to a selected element or field in the map. The definition information and location tables that you define for the internal system partner can be used for all partners. The information that you can use includes:

- Partner or Internal System Partner Definition (Name, EDI Code, Application Code)
- Partner or Internal System Partner Location Table Fields
- Lookup Table Fields
- Cross-Reference Table Fields
- Interchange, Group, or Document Record

Standard Rule tab (Select)

This diagram illustrates the Standard Rule tab with the Select function selected.



Parts and functions

This table describes the parts of the Select standard rule and their functions.

Part	Function	
table and key (or group)	Specifies the table and key the system uses to select data. This table lists the valid values.	
name of sub-table	Contains the sub-table name if appropriate. Note The sub table box is only active if you select Partner Lookup, Partner cross-reference by my item, Partner xref by partner item, Division lookup, Division cross-reference by my item, or Division cross-reference by partner item from the table and key list.	
	Part	Function
	Partner by EDI code	Indicates that the key is the partner's EDI Code and enables you to map from one of three fields for this partner (Name, EDI Code, Alternate Code) from the partner table.
	Partner by alternate code	Indicates that the key is the partner's Application Code and enables you to map from one of three fields for this partner (Name, EDI Code, Alternate Code) from the partner table.
	Partner by partner key	Indicates that the key is the partner's Profile ID enables you to map from one of three fields for this partner (Name, EDI Code, Alternate Code) from the partner table.
	Partner location by name	Indicates that the name on the Locations dialog box (for the current partner) of the Partner Editor is the key and enables you to map from any field in that table.
	Partner lookup	Indicates that the key is the partner lookup table name for this partner and enables you to map from any field in that table.
	Partner cross-reference by my item	Indicates that the key is the your item (value) on the partner cross-reference table for this partner and enables you to map from any field in that table.
		(Continued on next page)

(Contd) Part	Function	
name of sub-table (contd)	Partner cross-reference by partner item	Indicates that the key is the your partner's item (value) on the partner cross-reference table for this partner and enables you to map from any field in that table.
	Division	Enables you to map from one of three fields (Name, EDI Code, Alternate Code) for the <Internal System User> (system partner) in that table.
	Division location by name	Indicates that the key is the name on the Locations dialog box of the Partner Editor for the <Internal System User> and enables you to map from any field in that table.
	Division lookup	Indicates that the key is the partner lookup table name for the <Internal System User> partner and enables you to map from any field in that table.
	Division cross-reference by my item	Indicates that the key is the your item (value) on the partner cross-reference table for the <Internal System User> partner and enables you to map from any field in that table.
	Division cross-reference by partner item	Indicates that the key is the your partner's item (value) on the partner cross-reference table for the <Internal System User> partner and enables you to map from any field in that table.
	Document record	<p data-bbox="917 1262 1412 1323">Enables you to map from any of the fields within the current document record.</p> <p data-bbox="917 1350 980 1373">Note</p> <p data-bbox="917 1379 1412 1503">This sub-table can only be accessed from the following map types; Transaction Break/Build, Import, Export, Print, Screen, Group Build, and Interchange Build.</p> <p data-bbox="917 1522 1385 1677">The information that is accessed will be the information of the last document that was processed at the table level accessed by the rules. This must be taken into consideration when using rules that access these tables.</p> <p data-bbox="917 1696 1390 1757">See <i>Table access examples</i> on page 3 - 8 for examples.</p> <p data-bbox="1157 1776 1403 1801">(Continued on next page)</p>

(Contd) Part	Function	
name of sub-table (contd)	Generic envelope segment	<p>Enables you to map from any of the fields within the current envelope information, regardless of the EDI standard used.</p> <p>Note This function is normally not used unless you are familiar with the generic envelope table structure for a given standard.</p>
	Interchange	<p>Enables you to map from any of the fields within the current interchange record.</p> <p>Note This sub-table can only be accessed from the following map types: Interchange Break or Build, Group Break, Transaction Break, and Export.</p> <p>The information that is accessed will be the information of the last interchange that was processed at the table level accessed by the rules. This must be taken into consideration when using rules that access these tables.</p> <p>See <i>Table access examples</i> on page 3 - 8 for examples.</p>
	Group	<p>Enables you to map from any of the fields within the current group record.</p> <p>Note Can only be accessed from the following map types: Group Break or Build, Transaction Break, Export, and Interchange Build.</p> <p>The information that is accessed will be the information of the last group that was processed at the table level accessed by the rules. This must be taken into consideration when using rules that access these tables.</p> <p>See <i>Table access examples</i> on page 3 - 8 for examples.</p>
Raise compliance error	<p>Specifies whether you want the system to generate an error if the Select does not find a valid table entry.</p> <p>Note The default is cleared (do not generate an error if the Select is not successful).</p> <p style="text-align: right;">(Continued on next page)</p>	

(Contd) Part	Function
Map from	<p>Specifies the element/field/TFD from the specified table entry from which you want to map the contents.</p> <p>Note Entries are displayed in the map from list <i>only</i> after you select a table and key.</p>
Map to	<p>Specifies the element/field/TFD to which you want to map the contents of the map from box.</p> <p>Notes</p> <ul style="list-style-type: none"> ▶ The system displays each element/field/TFD from the side of the map with which you are currently working (input or output) in this list. ▶ A total of eight fields can be mapped using one Select rule. ▶ Entries are displayed in the map to list <i>only</i> after you select a table and key, and these entries are activated <i>only</i> after you highlight an entry in the map from box.

Table access examples

Example 1

When accessing the Document table from the Interchange build map, the information retrieved will be from the last document processed through the Transaction Build map. If an import map updates a unique value to appfield6 for each document and the interchange build map tries to retrieve this information, and if all documents are sent in the same interchange, the only information retrieved in the Interchange Build map will be the information from appfield6 for the last document sent.

To retrieve the value in appfield6 for each document, the documents must be sent in their own interchange. This is set in the outbound partner relationship in the Interchange Definition under advanced settings (**Set Max number of documents per interchange** must be set to 1).

Example 2

When accessing the Group table from an Export map, the information retrieved will be from the last group processed through the Group Break map. If an inbound interchange file contained two groups: the first group contains 850 information and the second group contains 810 information, the rule in the 850 Export map will retrieve 810 information because the last group processed through the Group Break map was the 810 group.

How to Use Information from the Partner Definition

Introduction

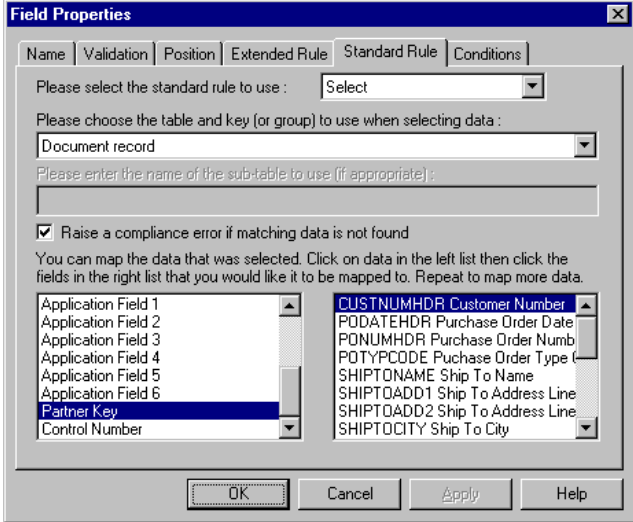
The Gentran:Server Application Integration subsystem allows you to use selected information from the Gentran:Server Partner Editor in your maps via a Select standard rule. You can map information from your trading partner's profile in the Partner Editor to a selected element, field, or TFD in the map. The information that you can use in the map includes any field in a table (location, lookup, or cross-reference) or data from the partner table.

For this example, you need to pull information from the partner definition in order to populate the Customer Number field (CUSTNUMHDR) on the application side of the map. Your partner's customer number identifies which trading partner sent the purchase order. This customer number is already part of the partner definition for this partner, so you can map it from the partner definition to the customer number field.

Procedure

Use this procedure to use the Select function to map the customer number from the partner definition.

Step	Action
1	<p>Double-click an existing element or field, or create a new element or field.</p> <p>Reference See <i>Using a Fixed-Format File</i> on page 2 - 51 and <i>Using an EDI File</i> on page 2 - 68 for more information on creating fields and elements.</p> <p>System response The system displays the Field (or Element or CII TFD) Properties dialog box.</p>
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose "Select."
4	<p>From the table and key list, select "Document record."</p> <p>Note This indicates that the system updates this field with selected information from the document record.</p>
5	Select the compliance error check box to indicate that a compliance error should be generated if the select does not find a valid entry.
6	<p>From the map from list, choose "Partner Key."</p> <p>Note This is the field that contains the customer number for this partner.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
7	<p>From the map to list, choose “CUSTNUMHDR.”</p> <p>Note</p> <p>This is the field to which the information from the partner key of the document record is mapped.</p> <p>System response</p> <p>This diagram illustrates how the Standard Rules tab should look.</p> 
8	Click OK to add the standard rule.

How to Use Information from a Cross-reference Table

Introduction

For this example, you need to map information from a cross-reference table to translate your partner's code for a purchased item into your code for the item. Your partner uses a customer product code (CUSTPROCEDURE) for each item that is meaningful to him. However, your company uses a UPC code (UPCCODE) for each item. After you and your partner determine what the equivalent customer product code is for each UPC code, you can create a cross-reference table in Partner Editor and use that information in your map.

A cross-reference table is used when you and your partner each reference an item by a different name (or number). For example, if your unique name for a "widget" is "wid" and your partner's unique name for the same "widget" is "1234." The system uses the cross-reference table to translate the two names for the "widget" item.

In this example, you have one code (UPC code) for each item ordered. Your partner refers to the same item by a different code (customer product code). You need to map the customer product code from the Product/Service ID element to the customer product code field (CUSTPROCEDURE) on the application side of the map. Also, you must create a cross-reference table to translate the customer product code to your UPC code.

Tip

Before you create the cross-reference table, you and your partner must determine what the equivalent customer product code is for each UPC code. Then you can create the cross-reference table in Partner Editor and use that information in your map.

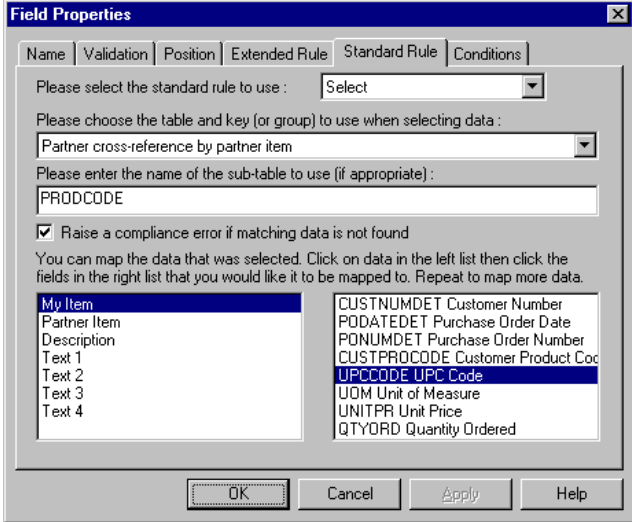
Finally, you need to use a standard rule to map the translated value from the CUSTPROCEDURE field to the UPCCODE field.

Procedure

Use this procedure to translate your partner's customer product code to your UPC Code for each item your partner ordered.

Step	Action
1	Map the customer product code from the Product/Service ID element to your CUSTPROCEDURE field. Reference See <i>How to Create Simple Links</i> on page 2 - 114 for more information. (Continued on next page)

(Contd) Step	Action
2	<p>In the Partner Editor, create a cross-reference table named “PRODCODE.”</p> <p>Reference See How to Create a Table in the <i>Gentran:Server User’s Guide</i> for more information on creating a cross-reference table.</p> <p>Notes</p> <ul style="list-style-type: none"> ▶ The cross-reference table you create only contains six entries. This is an example – an actual table would contain all of the items manufactured by your company that your partner purchases. ▶ In the cross-reference table, the “My Value” box contains your UPC code for the item. The “Partner Value” box contains your partner’s customer product code for the item. The “Description” box contains a description (and size, if applicable) of the item, and the Reference Data box contains additional quantity or color information.
3	<p>Double-click the CUSTPROCODE field.</p> <p>System response The system displays the Field Properties dialog box.</p>
4	<p>Select the Standard Rule tab to access standard rule options.</p>
5	<p>From the standard rule list, choose “Select.”</p>
6	<p>From the table and key, select “Partner cross-reference by partner item.”</p> <p>Note This indicates that you are using a cross-reference table and translating by your partner’s item.</p>
7	<p>In the sub table box, type “PRODCODE.”</p> <p>Note This is the name of the cross-reference table you created.</p>
8	<p>Select the compliance error check box to signal the system that if this translation fails, you want the system to generate an error.</p>
9	<p>From the map from list, choose “My Item.”</p> <p>Note This is the field from which the contents is mapped.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
10	<p>From the map to list, choose “UPCCODE.”</p> <p>Note This is the field to which the information from the lookup table is mapped.</p> <p>System response This diagram illustrates how the Standard Rule tab should look.</p> 



How to Use Information from Location Tables

Introduction

Each Partner Profile and the Internal System Partner Profile can have many associated location tables. The location table can be used to contain address-related information about the partner. You can store your partner's store addresses, warehouse addresses, or "invoice to" addresses.

To use information from a location table, you must have created that table already in the Partner Editor. See *How to Create a Table* in the *Gentran:Server User's Guide* for more information on creating a partner location table.

Procedure

Use this procedure to map information from a location table.

Step	Action
1	Double-click an existing element or field, or create a new element or field. Reference See <i>Using a Fixed-Format File</i> on page 2 - 51 and <i>Using an EDI File</i> on page 2 - 68 for more information on creating fields and elements. System response The system displays the Field (or Element or CII TFD) Properties dialog box.
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose "Select."
4	From the table list, select the key that is used to look up the Partner Location entry. Note You are allowed to map from any field on the Locations dialog box. Valid selections are: <ul style="list-style-type: none"> • Partner location by name • Division location by name
5	If you want the system to generate an error if the lookup fails, select the compliance error check box. <div style="text-align: right; color: green;">(Continued on next page)</div>

(Contd) Step	Action
6	From the map from list, select the field from which the contents are mapped. Note You can select from the following fields: Name, Address 1, Address 2, Address 3, City, State, Zip, Country, Telephone Number, Fax Number, Contact Name, Primary Reference, Secondary Reference.
7	From the map to list, select the element, field, or TFD to which the information from the Partner Editor is mapped.
8	Click OK to add the standard rule.

How to Use Information from Lookup Tables

Introduction A lookup table is used to select information related to a value in inbound or outbound data. Each partner profile and the internal system partner profile can have many lookup tables associated with it.

To use information from a lookup table, you must have created that table already in the Partner Editor. See *How to Create a Table* in the *Gentran:Server User's Guide* for more information on creating Partner lookup tables.

Procedure Use this procedure to map information from a Lookup Table.

Step	Action
1	<p>Double-click an existing element or field, or create a new element or field.</p> <p>Reference See <i>Using a Fixed-Format File</i> on page 2 - 51 and <i>Using an EDI File</i> on page 2 - 68 for more information on creating fields and elements.</p> <p>System response The system displays the Field (or Element or CII TFD) Properties dialog box.</p>
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose "Select."
4	<p>From the table list, select the key from which the system looks up the trading partner.</p> <p>Note Valid selections are:</p> <ul style="list-style-type: none"> ▶ Partner lookup ▶ Division lookup
5	In the sub table box, type the name of the lookup table.
6	If you want the system to generate an error if the lookup fails, select the compliance error check box.
7	From the map from list, select the field (Item, Description, Text 1, Text 2, Text 3, Text 4) from which the contents are mapped.
8	From the map to list, select the element, field, or TFD to which the information from the Partner Editor is mapped.
9	Click OK to add the standard rule.

Using the Update Standard Rule

Overview

Introduction

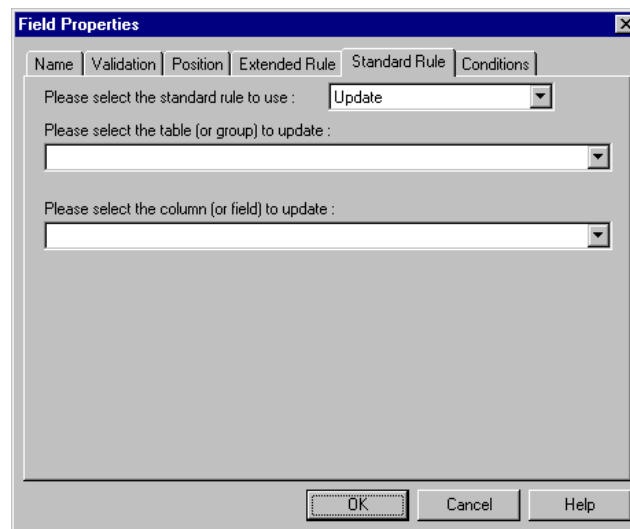
The Update function enables you to update a specific field in a document record, envelope segment, interchange, group, current partner, or document (if the map side format type is EDI), with the contents of the element, field, or TFD.

Warning

This function updates the internal Gentran:Server database tables. We recommend that you use this function only if you want to update the internal database tables. Typically, you use this function *only* if you want to update the document name and reference in the document table.

Standard Rule tab (Update)

This diagram illustrates the Standard Rule tab with the Update function selected.



Parts and functions

This table describes the parts of the Update standard rule and their functions.

Part	Function
table	<p>Contains the table that the system updates with the contents of the current field/element/TFD. Valid values are:</p> <ul style="list-style-type: none"> ▶ Document record (use to update the document name and reference. <p>Note This stable can only be accessed from the following map types; Transaction Break/Build, Import, Export, Print, Screen, Group Build, and Interchange Build.</p> <p>The information that is accessed will be the information of the last document that was processed at the table level accessed by the rules. This must be taken into consideration when using rules that access these tables.</p> <ul style="list-style-type: none"> ▶ Generic envelope segment (advanced use only) ▶ Interchange <p>Note This sub-table can only be accessed from the following map types: Interchange Break or Build, Group Break, Transaction Break, and Export.</p> <p>The information that is accessed will be the information of the last interchange that was processed at the table level accessed by the rules. This must be taken into consideration when using rules that access these tables.</p> <ul style="list-style-type: none"> ▶ Group <p>Note Can only be accessed from the following map types: Group Break or Build, Transaction Break, Export, and Interchange Build.</p> <p>The information that is accessed will be the information of the last group that was processed at the table level accessed by the rules. This must be taken into consideration when using rules that access these tables.</p> <ul style="list-style-type: none"> ▶ Partner (advanced use only) ▶ Document (advanced use only - if the format type of this side of the map is EDI) <p>See <i>Table access examples</i> on page 3 - 8 for examples.</p>
column	<p>Contains the database field from the column (or field) list that the system updates with the contents of the current field/element.</p> <p>Note The fields that are available in this list depend on the table selected.</p>

How to Set Up the Document Name and Reference

Introduction

The Document Name and Reference are two recommended fields that you can specify in the Application Integration subsystem. For an Invoice this field is typically the Invoice Number, and for a Purchase Order this field is typically the Purchase Order Number. If you specify either the Document Name or Reference, the corresponding field in Gentran:Server document browsers (In Documents, ?In Documents, Out Documents, ?Out Documents, Workspace, In Drawer, Out Drawer, Send Queue, Interchanges) is populated. The Document Name and Reference makes the identification of a document created by the map easier in Gentran:Server.

Setting up the Document Name is strongly recommended if you want to be able to differentiate between documents in the document browsers in Gentran:Server. If you set up the Document Name in the Application Integration subsystem, the Name box of the document browsers that contain this document in Gentran:Server contains the data from that mapped field. If you set up the Reference in the Application Integration subsystem, the RefData box of the document browsers that contain this document in Gentran:Server contains the data from that mapped field.

You can select any element as the Document Name, but we strongly recommend that you choose an element that only occurs *once* in the map. The element that you select should have a data type of “string” (data type is specified on the Element Properties dialog box). If you need to use an element with a data type of Date/Time or Number for the Document Name or Reference, you can either:

- Change the data type to “String” on the Element Properties dialog box.
- Set up an extended rule to update the document record.

Warning

If you change a Date/Time or Number element’s data type to “String,” you lose the data type compliance checking for the original type.

Procedure: setting up the document name

Use this procedure to set up the Document Name.

Step	Action
1	<p>Select a non-recurring element from the header or trailer segment of the EDI side of the document.</p> <p>Note The element that you select varies for each type of document. The selected element should be something that identifies this document meaningfully in Gentran:Server.</p> <p>Example</p> <ul style="list-style-type: none"> • For an Invoice this is typically the Invoice Number. • For a Purchase Order this is typically the Purchase Order Number. <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
2	Right-click the appropriate element and select Properties from the shortcut menu. System response The system displays the Element Properties dialog box (Name tab).
3	Select the Standard Rule tab to access the standard rule options.
4	From the standard rule list, select “Update.”
5	From the table to update list, select “Document record.”
6	From the column to update list, select “Document Name.”
7	Click OK to set up the Document Name.

Procedure: setting up the reference

Use this procedure to set up the Reference.

Step	Action
1	Select an element from the first segment of the EDI side of the document. Note The element that you select varies for each type of document. The selected element should be something that identifies this document meaningfully in Gentran:Server Example <ul style="list-style-type: none"> • For an Invoice this is typically the Purchase Order Number. • For a Purchase Order this is typically the Purchase Order Date. However, to use Purchase Order Date, you must use an extended rule on that element, instead of following the steps below.
2	Right-click the appropriate element and select Properties from the shortcut menu. System response The system displays the Element Properties dialog box (Name tab).
3	Select the Standard Rule tab to access the standard rule options.
4	From the standard rule list, select “Update.”
5	From the table to update list, select “Document record.”
6	From the column to update list, select “Reference Data.”
7	Click OK to set up the Reference.

Using the System Variable Standard Rule

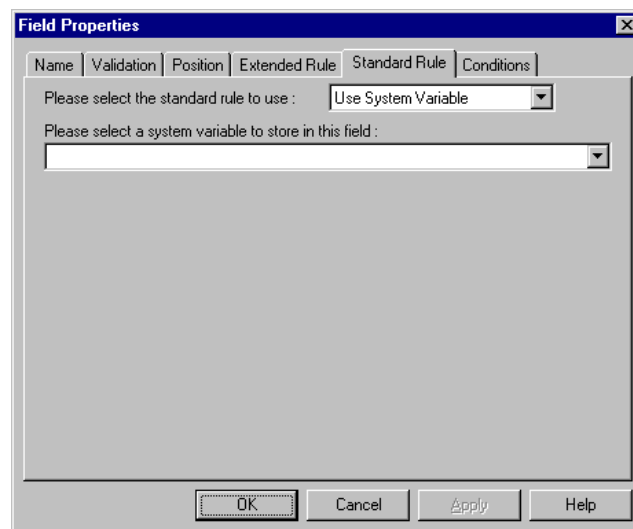
Overview

Introduction

The Use System Variable function enables you to set a variable that maps the current date and time to the selected element, field, or TFD. The selected map component must have a data type of “Date/Time.”

Standard Rule tab (System Variable)

This diagram illustrates the Standard Rule tab with the System Variable function selected.



Standard rule tab (System Variable) parts and functions

This table describes the parts of the System Variable standard rule and their functions.

Part	Function
System variable	Allows you to map “Current date and time” variables to the element, field, or TFD.

How to Use the System Date and Time

Procedure Use this procedure to use the system date and time.

Step	Action
1	Double-click an existing element, field, or TFD, or create a new one. Reference See <i>Using a Fixed-Format File</i> on page 2 - 51, <i>Using an EDI File</i> on page 2 - 68, and <i>Using a CII File</i> on page 2 - 89 for more information on creating fields, elements, and TFDs. System response The system displays the Field (or Element or CII TFD) Properties dialog box.
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use System Variable.”
4	Select “Current date and time” from the system variable list to map those variables to the element, field, or TFD.
5	Click OK to set up the system variable.

Using the Use Constant Standard Rule

Overview

Introduction

The Use Constant function enables you to move a literal constant value to the specified element, field, or TFD and indicate a qualifying relationship with another element, field, or TFD.

Why constants are used

Constants are used in maps to hold information that is needed later in the map, either to be moved to an output field or used in a conditional statement. Typically, constants are used to move a literal constant value to the specified element, field, or TFD on an inbound map. Outbound EDI qualifiers are typically generated from constants (to indicate a qualifying relationship with another element or field).

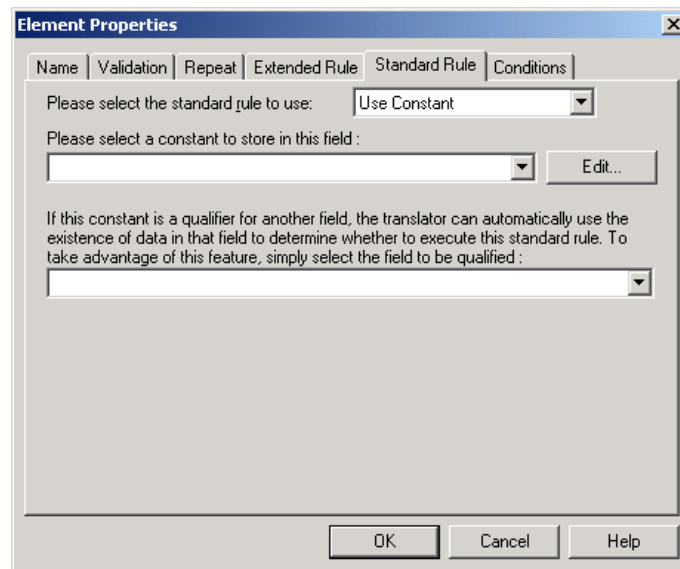
Literal constants are used by the system as a repository to store information that is used later in the map. Typically, constants are used to load common data into each application record on a map. You create the literal constant and name it. You can “hard code” (enter) a value that is loaded into the element, field, or TFD that uses the constant. However, to *use* a constant (the information that is stored in it), you need to use an standard rule, as explained below.

For an inbound map, constants are typically used to define the record types. For an outbound map, constants are typically used to define qualifying relationships.

A qualifying relationship establishes a relationship between an element and its qualifier. A qualifier contains a code that further defines the element. Qualifying relationships are typically defined in outbound maps.

Standard Rule tab (Use Constant)

This diagram illustrates the Standard Rule tab with Use Constant selected.



Standard rule tab (Use Constant) parts and functions

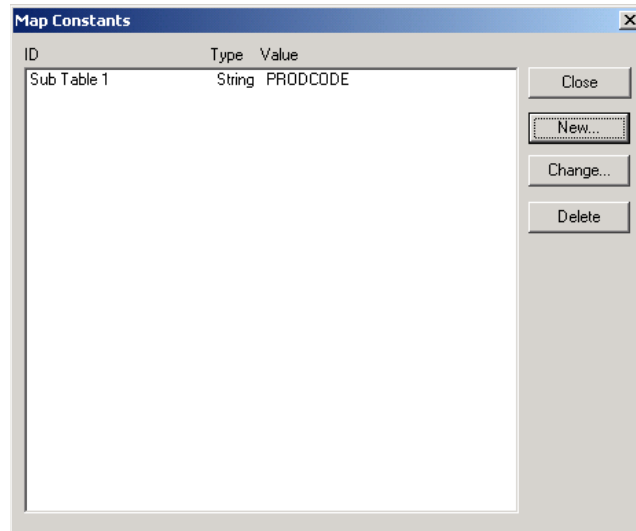
This table lists the parts of the Standard Rule tab (with the Use Constant standard rule selected) and their functions.

Part	Function
constant	Lists the available constants. The selected constant is mapped to the current element, field, or TFD. Note If the necessary constant is not present in the constant list, you need to create it by using the Map Constants dialog box, which you can access by clicking Edit .
Edit	Access the Map Constants dialog box, which allows you to create, edit, and delete literal constants.
qualifier	Lists the elements, fields, or TFDs that you can use to qualify the selected map component to establish a qualifying relationship between the two map components. Note If the qualified element is not generated because of a lack of data, the constant is not moved to the current element, field, or TFD.

(Continued on next page)

Map Constants dialog box

This diagram illustrates the Map Constants dialog box.



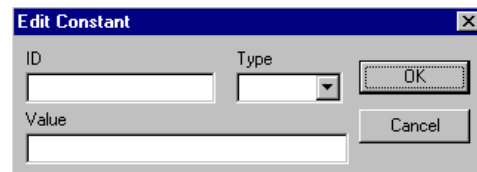
Map Constants dialog box parts and functions

This table lists the parts of the Map Constants dialog box and their functions.

Part	Function
Constants list	Contains a list of all literal constants currently defined in the system.
Close	Exits the Map Constants dialog box.
New	Access the Edit Constant dialog box, which allow you to create a new constant.
Edit	Access the Edit Constant dialog box, which allows you to edit the selected constant.
Delete	Removes the selected constant from the system.

Edit Constant dialog box

This diagram illustrates the Edit Constant dialog box.



Edit Constant dialog box parts and functions

This table lists the parts of the Edit Constant dialog box and their functions.

Part	Function
ID	Contains the literal constant identifier. Note This is typically a description of the field or element in which the constant is used. If you need to refer to the constant in an extended rule, you should use the data from this field.
Type	Specifies the category of this literal constant. Valid selections are: <ul style="list-style-type: none"> ▶ Integer (select for numeric constants that are a positive or negative natural (non-fraction) number or “0” (zero)) ▶ Real (select for numeric constants that are a positive or negative integer with an explicit decimal point) ▶ String (select for alphanumeric constants)
Value	Specifies the actual constant expression. Note This is the value of the literal constant.

How to Use a Constant in a Map

Procedure Use this procedure to use a constant in your map..

Step	Action
1	<p>Double-click an existing element, field, or TFD, or create a new one.</p> <p>Reference See <i>Using a Fixed-Format File</i> on page 2 - 51, <i>Using an EDI File</i> on page 2 - 68, and <i>Using a CII File</i> on page 2 - 89 for more information on creating fields, elements, and TFDs.</p> <p>System response The system displays the Field (or Element or CII TFD) Properties dialog box (Name tab).</p>
2	Select the Standard Rule tab to access standard rule options.
3	<p>From the standard rule list, select “Use Constant.”</p> <p>System response The system displays the constant options.</p>
4	<p>Do you need to create or edit a constant to use for the current element, field, or TFD?</p> <ul style="list-style-type: none"> • If <i>yes</i>, click Edit to access the Map Constants dialog box. <p>Reference See <i>How to Define and Edit Literal Constants</i> on page 3 - 30 for more information on creating and editing constants.</p> <ul style="list-style-type: none"> • If <i>no</i>, continue with the next step.
5	From the constants list, select the constant that you want to map to the current element, field, or TFD.
6	<p>Do you want to establish a qualifying relationship between the current map component and another component?</p> <ul style="list-style-type: none"> • If <i>yes</i>, from the qualifies list, select the element, field, or TFD that the system uses to determine whether to execute this standard rule (if the qualifying component contains data). • If <i>no</i>, continue with the next step.
7	Click OK to save the standard rule and exit the Properties dialog box.

How to Define a Qualifying Relationship

Introduction

Literal constants are used by the system as a repository to store information that is used at a later point in the map. Typically, constants are used in an outbound map to generate a qualifier. A qualifier is an element that has a value expressed as a code that gives a specific meaning to the function of another element. A qualifying relationship is the interaction between an element and its qualifier. The function of the element changes depending on which code the qualifier contains.

In this example, you use a constant to define a qualifier for a Product/Service ID.

Procedure

Use this procedure to define a qualifying relationship.

Step	Action
1	Double-click the Product/Service ID Qualifier that you want to use to further define (qualify) the Product/Service ID. System response The system displays the Element Properties dialog box (Name tab).
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Constant.”
4	Click Edit . System response The system displays the Map Constants dialog box.
5	Click New . System response The system displays the Edit Constant dialog box.
6	In the ID box, type the literal constant identifier. Note This is typically a description of the field or element in which the constant is used. If you need to refer to the constant in an extended rule, you should use the data from this box.
7	From the Type list, select String. Note The Product/Service ID Qualifier is formatted as a string data type. This indicates the category of the literal constant.

(Continued on next page)

(Contd) Step	Action
8	In the Value box, type “UP”. Note This indicates that the Product/Service ID field must contain the UPC Code. This is the value of the literal constant.
9	Click OK to add the constant to the system.
10	Click Close to exit the Map Constants dialog box.
11	From the qualifies list, select Product/Service ID. This is the element that the Product/Service ID Qualifier qualifies. Note This list contains only the other <i>active</i> fields or elements in the same record or segment as the qualifying field.
12	Click OK on the Element Properties dialog box and the qualifying relationship between the two elements is established.

How to Define and Edit Literal Constants

Procedure Use this procedure to create or edit a literal constant so you can use it to store information.

Step	Action
1	From the Edit menu, select Constants . System response The system displays the Map Constants dialog box.
2	Do you want to create a new constant? <ul style="list-style-type: none"> • If <i>yes</i>, click New. • If <i>no</i> (you want to edit an existing constant), select the constant and click Edit. System response The system displays the Edit Constant dialog box.
3	In the ID field, type the literal constant identifier.
4	From the Type list, select the category of this literal constant.
5	In the Value field, type the actual constant expression.
6	Click OK to add the constant to the system.
7	Click Close to exit the Map Constants dialog box.

How to Delete Literal Constants

Procedure Use this procedure to delete a literal constant.

Step	Action
1	From the Edit menu, select Constants . System response The system displays the Map Constants dialog box.
2	Select the constant you want to delete.
3	Click Delete . System response The system removes the constant without warning.

How to Map Literal Constants

Procedure Use this procedure to map a constant in which you previously stored information using an extended rule.

Step	Action
1	Double-click the element, field, or TFD in which you want to use the constant. System response The system displays the Field (or Element or CII TFD) Properties dialog box.
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Constant.”
4	From the constant list, select the constant that you want to use.
5	Click OK . System response The data that was stored in the selected constant is loaded in the element, field, or TFD.

How to Generate Qualifiers

Introduction A qualifier is typically an element that has a value expressed as a code that gives a specific meaning to the function of another element. A qualifying relationship is the interaction between an element and its qualifier. The function of the element changes depending on which code the qualifier contains.

Procedure Use this procedure to define qualifying relationships.

Step	Action
1	Double-click the element you want to use to further define (qualify) another element. System response The system displays the Element Properties dialog box.
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Constant.”
4	From the qualifies list, select the element that this element qualifies. Note This list contains only the other <i>active</i> elements in the same record or segment as the qualifying element.
5	Click OK . System response The system establishes the qualifying relationship between the two elements.

Using the Loop Count Standard Rule

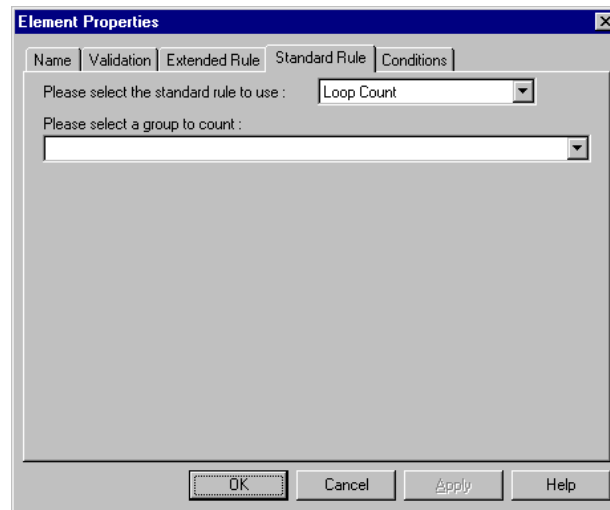
Overview

Introduction

The Loop Count function enables you to count the number of times a loop is repeated, if the element, field, or TFD is part of a loop. If the loop is a nested loop, you can track the current loop *or* the outer loop. For example, if the Y loop is nested within the X loop, and the Y loop has cycled through 15 iterations and the X loop has cycled through 3 iterations, you can choose to count either the “15” (Y loop) or the “3” (X loop).

Standard Rule tab (Loop Count)

This diagram illustrates the Standard Rule tab with Loop Count selected.



Standard rule tab (Loop Count) parts and functions

This table lists the parts of the Standard Rule tab (with the Loop Count standard rule selected) and their functions.

Part	Function
group	<p>Contains the loop that you want to count.</p> <p>Note If the loop is a nested loop, you can track the current loop <i>or</i> the outer loop.</p> <p>Example If the Y loop is nested within the X loop, and the Y loop has cycled through 15 iterations and the X loop has cycled through 3 iterations, you can choose to count either the “15” (Y loop) or the “3” (X loop).</p>

How to Use the Loop Count Function

Procedure Use this procedure to use the Loop Count function.

Step	Action
1	Double-click an existing element, field, or TFD, or create a new one. Reference See <i>Using a Fixed-Format File</i> on page 2 - 51, <i>Using an EDI File</i> on page 2 - 68, and <i>Using a CII File</i> on page 2 - 89 for more information on creating fields, elements, and TFDs). System response The system displays the Field (or Element or CII TFD) Properties dialog box.
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Loop Count.”
4	Select the loop that you want to count.
5	Click OK to add the standard rule.

Using the Use Accumulator Standard Rule

Overview

Introduction

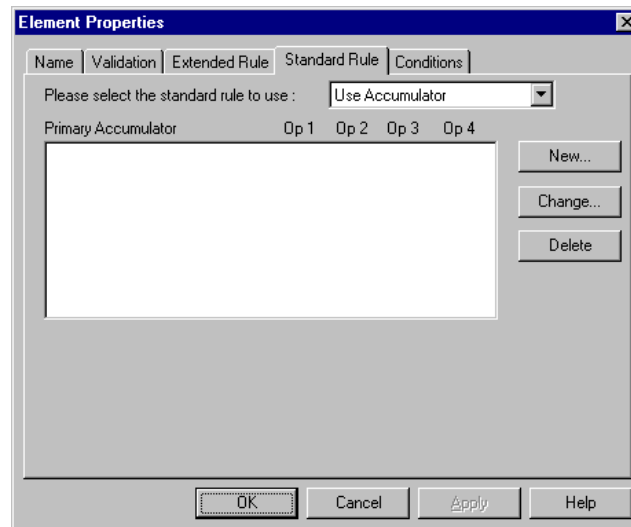
The Use Accumulator function gives you access to a set of numeric variables that you can manipulate via numeric operations, and then transfer to and from fields. This function enables you to add, change, or delete calculations for the element, field, or TFD, including hash totals (used to accumulate numeric field values, i.e., quantity, price, etc.). This function also enables you to map the accumulated total into a control total field, and use accumulators. Accumulators are used generally for counting the occurrences of a specific element or generating increasing or sequential record or line item numbers.

Notes

- Accumulators are global variables.
- Accumulators are set to zero prior to being used in calculations.
- The order in which accumulator operations are performed depends on the hierarchical order of the map components.

Standard Rule tab (Use Accumulator)

This diagram illustrates the Standard Rule tab with Use Accumulator selected.



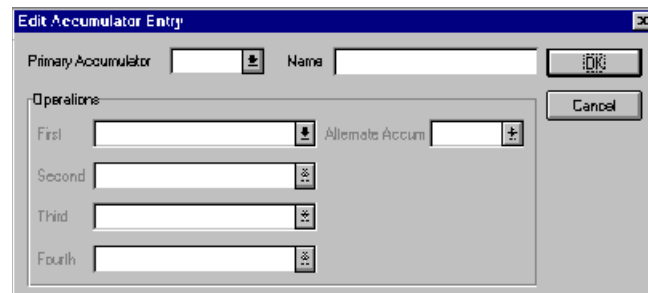
Standard rule tab (Use Accumulator) parts and functions

This table lists the parts of the Standard Rule tab (with the Use Accumulator standard rule selected) and their functions.

Part	Function
Primary accumulator	Contains all existing calculations that were created for this element, field, or TFD.
New	Accesses the Edit Accumulator Entry dialog box to create a new calculation for this element, field, or TFD.
Change	Highlight a calculation in the Primary Accumulator list and click this to access the Edit Accumulator Entry dialog box to edit the selected calculation.
Delete	Highlight a calculation in the Primary Accumulator list and click this to delete the selected calculation. Warning The selected calculation is deleted without warning.

Edit Accumulator Entry dialog box

This diagram illustrates the Edit Accumulator Entry dialog box.



Edit Accumulator Entry dialog box parts and functions

This table lists the parts of the Edit Accumulator Entry dialog box and their functions.

Part	Function
Primary accumulator	<p>Specifies the primary accumulator.</p> <p>Notes</p> <ul style="list-style-type: none"> • Before any calculations are performed on an accumulator, its content is “0” (zero). When you use an accumulator, the system adds a new accumulator to the bottom of this list. • There is <i>only one</i> set of accumulators for each map. This means that accumulator “0,” whether it is used in the Primary Accumulator or Alternate Accum box is the same accumulator with the same contents. If you assign calculations to accumulator “0” at the beginning of the map and then use accumulator “0” again later in the map, the content of that accumulator is the result of the earlier calculation. Any additional calculations you assign to that accumulator are performed on the contents resulting from an earlier calculation.
Name	<p>Contains a descriptive alias that enables you to differentiate what the accumulators you create are used for.</p>
First	<p>Specifies the first operation that the system performs.</p> <p>Notes</p> <ul style="list-style-type: none"> • The First box is active only after you select a Primary Accumulator. • Before any calculations are performed on an accumulator, its content is “0” (zero). When you use an accumulator, the system adds a new accumulator to the bottom of this list. <p>Reference See <i>Accumulator operations</i> on page 3 - 39 for more information.</p>
Second	<p>Specifies the second operation that the system performs, after the First operation is completed.</p> <p>Note The Second box is active only after you select a First operation that does not involve the Alternate Accum.</p> <p>Reference See <i>Accumulator operations</i> on page 3 - 39 for more information.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Part	Function
Third	<p>Specifies the third operation that the system performs, after the Second operation is complete.</p> <p>Note The Third box is active only after you select a Second operation.</p> <p>Reference See <i>Accumulator operations</i> on page 3 - 39 for more information.</p>
Fourth	<p>Specifies the fourth operation that the system performs, after the Third operation is complete</p> <p>Note The Fourth box is active only after you select a Third operation.</p> <p>Reference See <i>Accumulator operations</i> on page 3 - 39 for more information.</p>
Alternate Accum	<p>Specifies an alternate accumulator that participates in the accumulator operation.</p> <p>Note The Alternate Accum box is only active if you select an operation from the First field that involves an alternate operand.</p> <p>Reference See <i>Accumulator operations</i> on page 3 - 39 for more information.</p>

Accumulator operations

This table lists the available accumulator operations and their functions.

Part	Function
Increment primary	Adds "1" (one) to the contents of the Primary Accumulator (i.e., Primary = Primary + 1).
Decrement primary	Subtracts "1" (one) from the contents of the Primary Accumulator (i.e., Primary = Primary - 1).
Sum in primary	Adds the numeric value (takes the positive or negative sign of the numbers into account) of the field to the contents of the Primary Accumulator (i.e., Primary = (+/-)Primary + (+/-)Field).

(Continued on next page)

(Contd) Part	Function
Hash sum in primary	Adds the absolute value (does not take the positive or negative sign of the numbers into account) of the field to the contents of the Primary Accumulator (i.e., Primary = Primary + Field).
Load primary	Loads the contents of the field into the Primary Accumulator (i.e., Primary = Field).
Use primary	Loads the contents of the Primary Accumulator into the field (i.e., Field = Primary).
Zero primary	Sets the value of the Primary Accumulator to zero (i.e., Primary = 0).
Multiply with primary	Multiplies the field with the contents of the Primary Accumulator, and stores the result in the Primary Accumulator (i.e., Primary = Primary * Field).
Divide by primary	Divides the field with the contents of the Primary Accumulator, and stores the result in the Primary Accumulator (i.e., Primary = Field / Primary).
Divide primary by field	Divides the contents of the Primary Accumulator with the field, and stores the result in the Primary Accumulator (i.e., Primary = Primary / Field).
Modulo with primary	Divides the contents of the Primary Accumulator with the contents of the field, and stores the <i>remainder</i> of that operation in the Primary Accumulator (i.e., Primary = Field % Primary).
Modulo with field	Divides the contents of the field with the contents of the Primary Accumulator, and stores the <i>remainder</i> of that operation in the Primary Accumulator (i.e., Primary = Primary % Field).
Negate primary	<p>Makes the contents of the Primary Accumulator negative (i.e., Primary = Primary * -1).</p> <p>Note The only way to subtract the Primary Accumulator from the field is to “Negate” the Primary Accumulator and then use the “Sum in primary” operation to add the negative Primary Accumulator to the field.</p>
Move primary to alternate	Copies the contents of the Primary Accumulator to the Alternate Accum. This overwrites the current contents of the Alternate Accum field (i.e., Alternate = Primary).
Add primary to alternate	<p>Adds the contents of the Primary Accumulator to the contents of the Alternate Accum, and stores the result in the Primary Accumulator (i.e., Primary = Primary + Alternate).</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Part	Function
Multiply primary by alternate	Multiplies the contents of the Primary Accumulator with the contents of the Alternate Accum, and stores the result in the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} * \text{Alternate}$).
Divide primary by alternate	Divides the contents of the Primary Accumulator with the contents of the Alternate Accum, and stores the result in the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} / \text{Alternate}$).
Modulo primary with alternate	Divides the contents of the Primary Accumulator with the contents of the Alternate Accum, and stores the <i>remainder</i> of that operation in the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} \% \text{Alternate}$).

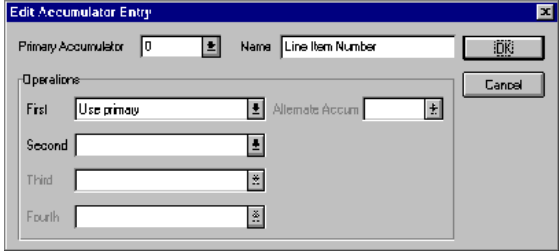
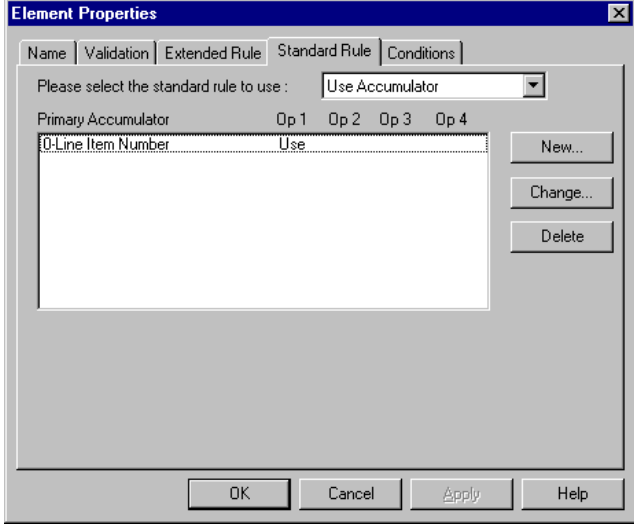
How to Count Line Items

Introduction In this example, you want an incremental count of the number of line items, and you want to use that total value in the Number of Line Items Total field.

Procedure Use this procedure to count line items and generate a control total for an outbound ANSI X12 purchase order.

Step	Action
1	<p>Double-click the P0101 element (in the P01 segment in the P01 group).</p> <p>Note This is the element that you typically use to count the line items.</p> <p>System response The system displays the Element Properties dialog box.</p>
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Accumulator.”
4	<p>Click New.</p> <p>System response The system displays the Edit Accumulator Entry dialog box to create a new calculation for this element.</p>
5	<p>From the Primary Accumulator list, select “0.”</p> <p>Note There is <i>only one</i> set of accumulators for each map. This means that accumulator “0,” whether it is used in the Primary Accumulator or Alternate Accum box is the same accumulator with the same contents. If you assign calculations to accumulator “0” at the beginning of the map and then use accumulator “0” again later in the map, the content of that accumulator is the result of the earlier calculation. Any additional calculations you assign to that accumulator are performed on the contents resulting from an earlier calculation.</p>
6	<p>In the Name box, type “Line Item Number.”</p> <p>Note This is a descriptive alias that enables you to differentiate what the accumulators you create are used for.</p> <p style="text-align: right;">(Continued on next page)</p>
7	From the First list, select Increment Primary . This is the first operation that the system performs. This specifies that the system will increment the PO101 element by one.

(Contd) Step	Action
8	From the Second list, select “Use primary.” Note This is the second operation that the system performs, after the First operation is completed. This specifies that the system loads the current value of the accumulator into the P0101 (Assigned Identification) element.
9	Click OK to add the accumulator.
10	Click OK on the Element Properties dialog box to add the standard rule to the P0101 element.
11	Double-click the CTT01 element (in the CTT segment). Note This is the element that typically contains the total number of line items. System response The system displays the Element Properties dialog box.
12	Select the Standard Rule tab to access standard rule options.
13	From the standard rule list, select “Use Accumulator.”
14	Click New . System response The system displays the Edit Accumulator Entry dialog box to create a new calculation for this element.
15	From the Primary Accumulator list, select primary accumulator “0.” Note This accumulator currently contains the total number of line items. (Continued on next page)

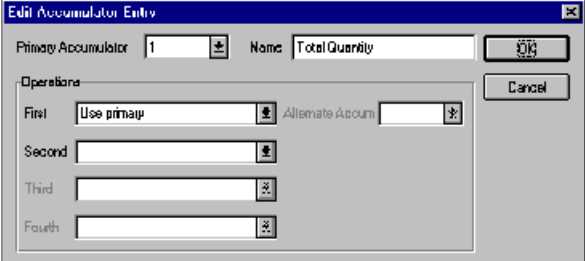
(Contd) Step	Action
16	<p>From the First list, select “Use primary.”</p> <p>Note This specifies that the system loads the current value of the accumulator into the CTT01 (Number of Line Items Total) element.</p> <p>System response This diagram illustrates how the Edit Accumulator dialog box should look.</p> 
17	<p>Click OK to add the accumulator.</p> <p>System response This diagram illustrates how the Standard Rule tab should look.</p> 
18	<p>Click OK on the Element Properties dialog box to add the standard rule to the CTT01 element.</p> <p>Note The CTT01 element now contains the total number of line items in the purchase order.</p>

How to Calculate Hash Totals

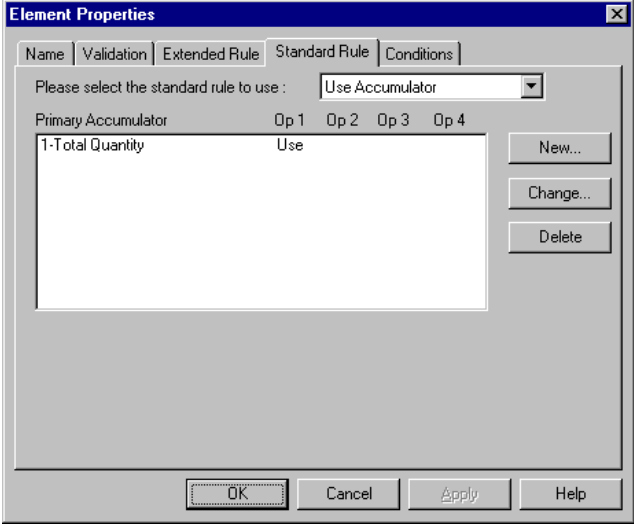
Introduction In this example, you want to count the quantity ordered for each line item and load the total quantity in the CTT02 (Hash Total) element.

Procedure Use this procedure to count the quantity ordered and generate a hash total for an outbound ANSI X12 purchase order.

Step	Action
1	Double-click the P0102 element (in the P01 segment in the P01 group). Note This is the element that you typically use to count the line items. System response The system displays the Element Properties dialog box.
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Accumulator.”
4	Click New . System response The system displays the Edit Accumulator Entry dialog box to create a new calculation for this element.
5	From the Primary Accumulator list, select “1.”
6	In the Name box, type “Total Quantity.”
7	From the First list, select Hash Sum in Primary . Note This is the first operation that the system performs. This specifies that the system will add the numeric value of the PO102 element to the contents of the Primary Accumulator.
8	Click OK to add the accumulator.
9	Click OK on the Element Properties dialog box to add the standard rule to the P0102 element. <i>(Continued on next page)</i>

(Contd) Step	Action
10	<p>Double-click the CTT02 element (in the CTT segment).</p> <p>Note This is the element that typically contains the total quantity of the purchase order.</p> <p>System response The system displays the Element Properties dialog box.</p>
11	Select the Standard Rule tab to access standard rule options.
12	From the standard rule list, select “Use Accumulator.”
13	<p>Click New.</p> <p>System response The system displays the Edit Accumulator Entry dialog box to create a new calculation for this element.</p>
14	<p>From the Primary Accumulator list, select primary accumulator “1.”</p> <p>Note This accumulator currently contains the total quantity.</p>
15	<p>From the First list, select “Use primary.”</p> <p>Note This operation specifies that the system loads the current value of the accumulator into the CTT02 (Hash Total) element.</p> <p>System response This diagram illustrates how the Edit Accumulator dialog box should look.</p> 

(Continued on next page)

(Contd) Step	Action
16	<p>Click OK to add the accumulator.</p> <p>System response This diagram illustrates how the Standard Rule tab should look.</p> 
17	<p>Click OK on the Element Properties dialog box to add the standard rule to the CTT02 element.</p> <p>Note The CTT02 element now contains the total quantity of the purchase order.</p>

How to Reset and Calculate a Value Total

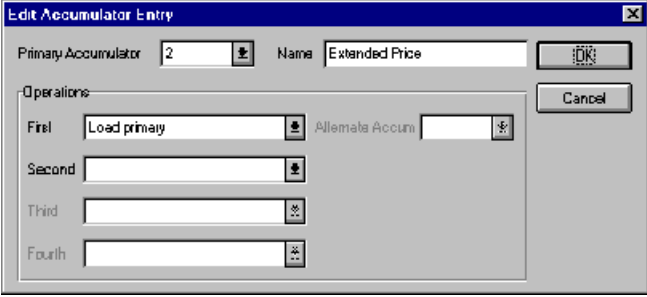
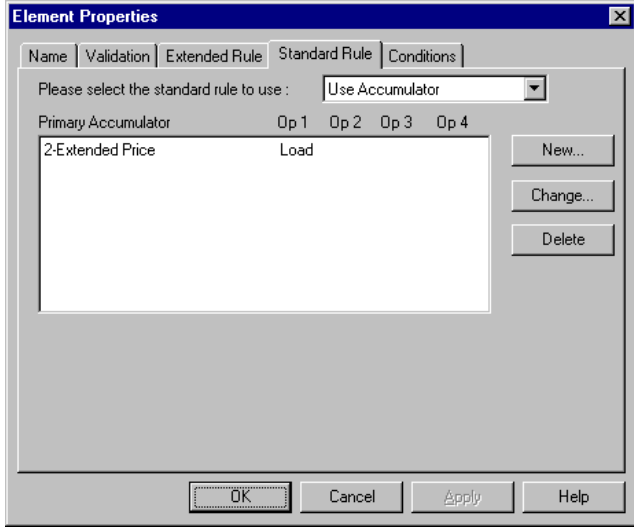
Introduction

In this example, you want to multiply the quantity invoiced for each line item by the unit price to obtain the extended price. Then, you want to generate a running total of extended price and load the final total into the TDS01 (Total Invoice Amount) element.

Procedure: multiplying quantity invoiced by unit price

Use this procedure to multiply the quantity invoiced for each line item by the unit price to obtain the extended price for an outbound ANSI X12 invoice.

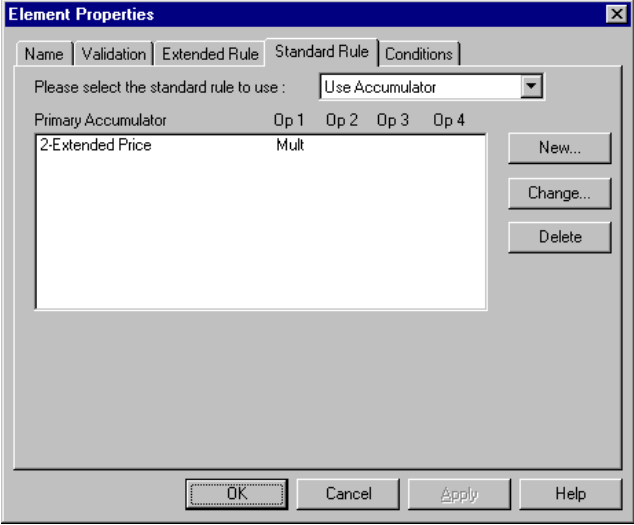
Step	Action
1	Double-click the IT102 element (in the IT1 segment in the IT1 group). Note This is the element that you typically use to count the quantity invoiced. System response The system displays the Element Properties dialog box.
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Accumulator.”
4	Click New . System response The system displays the Edit Accumulator Entry dialog box to create a new calculation for this element.
5	From the Primary Accumulator list, select “2.”
6	In the Name box, type “Extended Price.” <div style="text-align: right;">(Continued on next page)</div>

(Contd) Step	Action
7	<p>From the First list, select “Load primary.”</p> <p>Note This operation specifies that the system loads the contents of the element into the Primary Accumulator for each iteration of the IT1 group.</p> <p>System response This diagram illustrates how the Edit Accumulator dialog box should look.</p> 
8	<p>Click OK to add the accumulator.</p> <p>System response This diagram illustrates how the Standard Rule tab should look.</p> 
9	<p>Click OK on the Element Properties dialog box to add the standard rule to the IT102 element.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>



(Contd) Step	Action
10	Double-click the IT104 element (in the IT1 segment in the IT1 group). Note This is the element that contains the unit price for each line item. System response The system displays the Element Properties dialog box.
11	Select the Standard Rule tab to access standard rule options.
12	From the standard rule list, select “Use Accumulator.”
13	Click New . System response The system displays the Edit Accumulator Entry dialog box to create a new calculation for this element.
14	From the Primary Accumulator list, select primary accumulator “2.”
15	From the First list, select “Multiply with primary.” Note This operation specifies that the system multiplies the value of the IT104 (Unit Price) element with the contents of the primary accumulator, and store the result in the primary accumulator for each iteration of the IT1 group. System response This diagram illustrates how the Edit Accumulator dialog box should look. <div data-bbox="656 1266 1300 1556" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> </div>

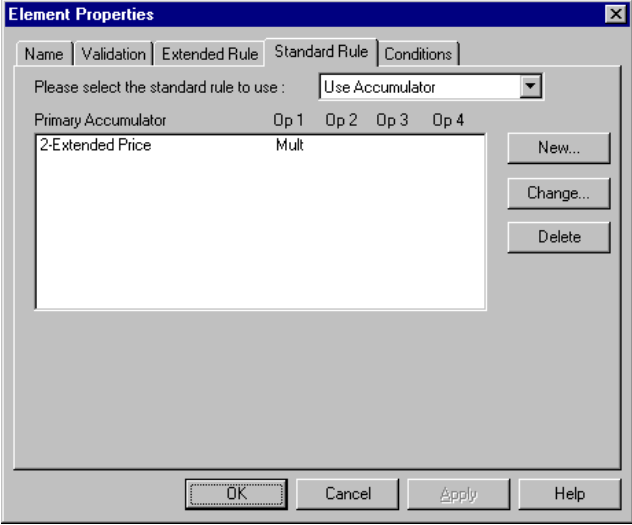
(Continued on next page)

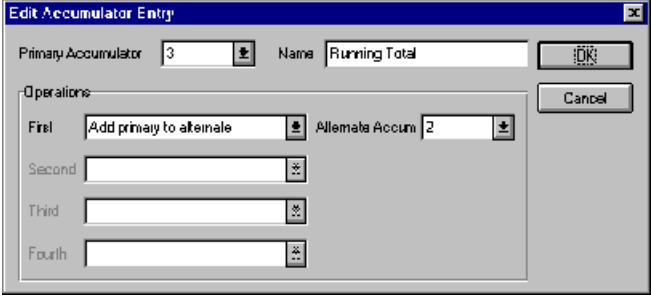
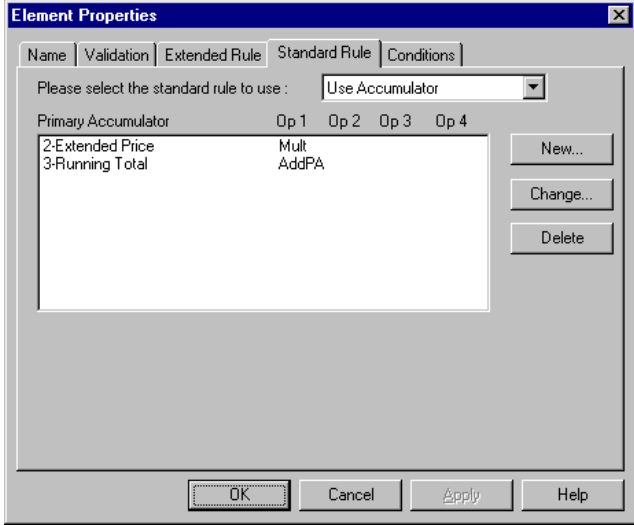
(Contd) Step	Action
16	<p>Click OK to add the accumulator.</p> <p>System response This diagram illustrates how the Standard Rule tab should look.</p> 
17	<p>Click OK on the Element Properties dialog box to add the standard rule to the IT104 element.</p> <p>Tip If there is an extended price element in your EDI file, you could load the total from the extended price calculation into that element. To do this, you need to use an accumulator on that extended price element that specifies “Use primary” for accumulator 2.</p>



**Procedure:
generating a
running total of
extended price**

Use this procedure to generate a running total of the extended price.

Step	Action
1	<p>Double-click the IT104 element (in the IT1 segment in the IT1 group).</p> <p>Note This is the element that contains the unit price for each line item.</p> <p>System response You already established one accumulator that the system displays in the list on the Standard Rule tab.</p> 
2	<p>Click New.</p> <p>System response The system displays the Edit Accumulator Entry dialog box to create a new calculation for this element.</p>
3	<p>From the Primary Accumulator list, select “3”.</p>
4	<p>In the Name box, type “Running Total.”</p>
5	<p>From the First list, select “Add primary to alternate.”</p> <p>Note This operation specifies that the system adds the contents of the primary accumulator to the contents of the alternate accumulator, and stores the result in the primary accumulator for each iteration of the IT1 group.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

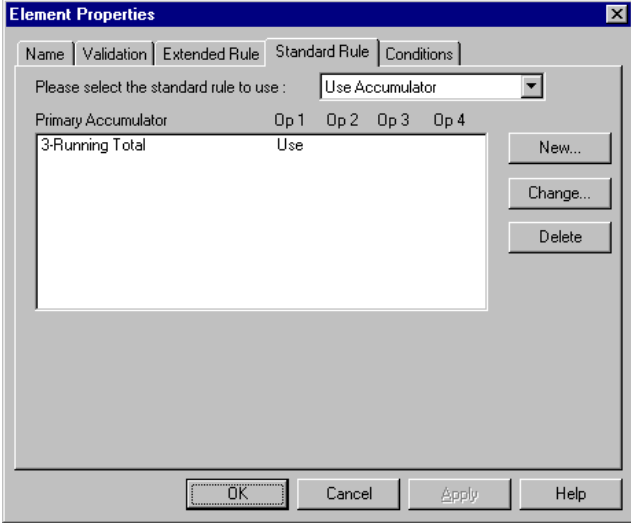
(Contd) Step	Action
6	<p>From the Alternate Accum list, select “2.”</p> <p>Note This operation specifies that the system add the value of accumulator “2” (which contains the extended price for a line item) to the value of accumulator “3.” The system stores the sum in accumulator “3,” which therefore contains a running total of the extended price with each iteration of the IT1 group.</p> <p>System response This diagram illustrates how the Edit Accumulator dialog box should look.</p> 
7	<p>Click OK to add the accumulator.</p> <p>System response This diagram illustrates how the Standard Rule tab should look.</p> 
8	<p>Click OK on the Element Properties dialog box to add the standard rule to the IT104 element.</p>

**Procedure: loading
a running total of
extended price**

Use this procedure to load the running total of the extended price into the TDS01 (Total Invoice Amount) element.

Step	Action
1	Double-click the TDS01 element. Note This is the element that contains the total invoice amount for each line item. System response The system displays the Element Properties dialog box.
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Accumulator.”
4	Click New . System response The system displays the Edit Accumulator Entry dialog box to create a new calculation for this element.
5	From the Primary Accumulator list, select “3.”
6	From the First list, select “Use primary.” Note This operation specifies that the system loads the contents of the primary accumulator into the TDS01 (Total Invoice Amount) element. System response This diagram illustrates how the Edit Accumulator dialog box should look. <div data-bbox="651 1346 1235 1608" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div>

(Continued on next page)

(Contd) Step	Action
7	<p>Click OK to add the accumulator.</p> <p>System response This diagram illustrates how the Standard Rule tab should look.</p> 
8	Click OK on the Element Properties dialog box to add the standard rule to the TDS01 element.

Using the Use Code Standard Rule

Overview

Introduction

The Use Code function enables you to match an element, field, or TFD against a predefined code table and specify whether or not a compliance error is generated if the map component does not contain one of the values in the code table. This function also allows you to store a code's description in another element, field, or TFD. You can also create a unique code table, use code values from a code table, and flag whether or not the system generates an error if a validation against the code table fails. You can import and export code lists and copy and paste code lists between maps.

The Gentran:Server Application subsystem enables you to create code tables to be used with the current map. You can set up code tables in the Application Integration subsystem to function like the partner cross-reference and lookup tables in Gentran:Server. However, code tables that are set up in the Application subsystem can be used *only* for the current map. Code tables that you create in Gentran:Server can be used globally for all maps.

Code List Tables are used by EDI standards as repositories for lists of codes. Each EDI standard provides a code list for each element that can be further defined with a code. Gentran:Server allows you to load code lists from the standard. You can either load all the codes in the table, or you can select only one or more codes from the table. Once you load a code table, you can use a "Use Code" standard rule to either look up a value from a code table or validate the contents of an element, field, or TFD against the values in the code table.

A element, field, or TFD with a Use Code rule enables values to either be checked against or selected from the codes in a specified code table. Codes are typically used to further qualify another element. For example, if the XX element contains address information, you can further qualify that element by choosing the code "SU" from the 0222 table. In the 0222 table, the code "SU" is described as a "supplier's address." Therefore, by using this code with the XX element, you are indicating that the XX element is not just address information, but address information for the *supplier*.

Standard Rule tab (Use Code)

This diagram illustrates the Standard Rule tab with Use Code selected.

Standard rule tab (Use Code) parts and functions

This table lists the parts of the Standard Rule tab (with the Use Code standard rule selected) and their functions.

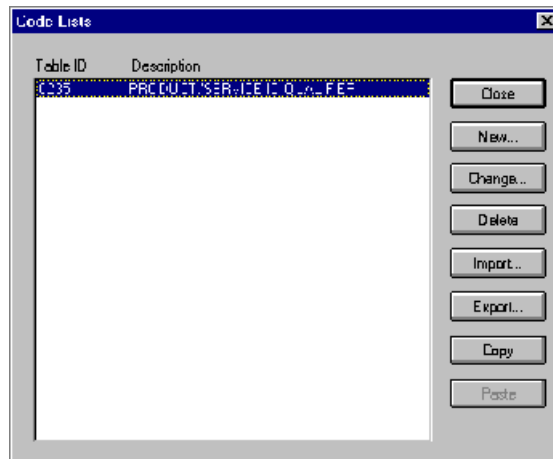
Part	Function
Code list	Contains all the code tables. Note If the necessary code table is not listed, click Edit to load or create a code table.
Edit	Accesses the Edit Code List dialog box, which enables you to create and edit code list tables.
Raise compliance error	Indicates that, for compliance reasons, the element, field, or TFD <i>must</i> contain one of the codes from the specified table (nothing else is valid for that field). Example If a field is defined as containing only “YES” or “NO,” you can set up an exclusive code table that contains only YES and NO. Then if you receive a “MAYBE” in that field, the system flags it as an error.

(Continued on next page)

(Contd) Part	Function
Code description	<p>Contains the element, field, or TFD where you want the description of the code (that is used) to appear when the selection is made.</p> <p>Example If the code is “SU,” it is much more useful to view the description of the code (“Supplier's Address). If you selected element “XX” from the store description list, the description for the code used is mapped to element “XX.”</p>

Code List dialog box

This diagram illustrates the Code List dialog box.



Edit Code List dialog box parts and functions

This table lists the parts of the Edit Code List dialog box and their functions.

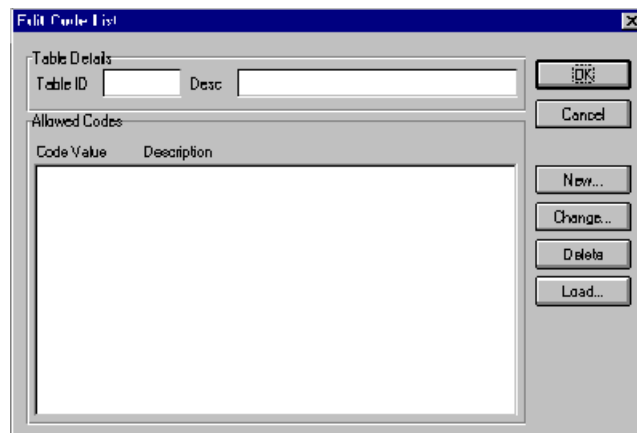
Part	Function
Table list	Specifies the table identifier.
New	Accesses the Edit Code List dialog box, which allows you to create a new code list.
Change	Accesses the Edit Code List dialog box, which allows you to edit the selected code list.
Delete	Deletes the selected code list
Import	Accesses the Open dialog box, which allows you to import a code list.

(Continued on next page)

(Contd) Part	Function
Export	Accesses the Save As dialog box, which allows you to export the selected code list.
Copy	Copies the selected code list.
Paste	Pastes a previously-copied code list in a map.

Edit Code List dialog box

This diagram illustrates the Edit Code List dialog box.



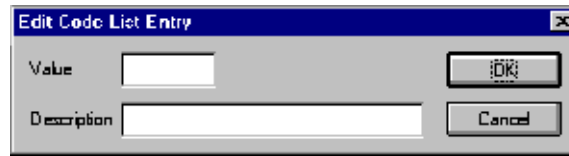
Edit Code List dialog box parts and functions

This table lists the parts of the Edit Code List dialog box and their functions.

Part	Function
Table ID	Contains the name of the field or element for which this code list table is used.
Desc	Contains the description of the field or element for which this code list table is used.
Allowed Codes	Specifies the codes that are allowed for this table.
New	Accesses the Edit Code List Entry dialog box, which allows you to create a new code.
Change	Accesses the Edit Code List Entry dialog box, which allows you to edit the selected code.
Delete	Accesses the Edit Code List Entry dialog box, which allows you to delete the selected code.
Load	Accesses the Load Code List dialog box, which allows you to select, from a standard code table, specific codes that you want to load or select the entire list of codes.

Edit Code List Entry dialog box

This diagram illustrates the Edit Code List Entry dialog box.



Edit Code List Entry dialog box parts and functions

This table lists the parts of the Edit Code List Entry dialog box and their functions.

Part	Function
Value	Specifies the actual value of the code.
Description	<p>Contains the code value description.</p> <p>Note The description is used if you specify an element or field (in the Store Fields list on the Field Properties dialog box) to which you want the code description mapped.</p>

How to Validate Data Against a Code List Table

Introduction For this example you need to map the customer product code from the Product/Service ID to the customer product code field (CUSTPROCEDURE) on the application side of the map. To ensure that the data that is mapped from the Product/Service ID element really is your partner's customer product code, you need to establish and use a code table with the Product Service ID *Qualifier*.

Procedure Use this procedure to validate the data in the Product/Service ID field against a code list table.

Step	Action
1	<p>Load the code list table from the standard for the Product/Service ID Qualifier element.</p> <p>Reference See <i>How to Load a Code List Table from the Standard</i> on page 3 - 66 for more information.</p> <p>Note The standards provide code list tables only for elements that use them. For example, in the TD4 segment, the TD401 (Special Handling Code) and TD403 (Hazardous Material Class Code) have code tables provided by the standard.</p>
2	<p>Define a code list table for the Product/Service ID Qualifier element that only contains the code value "BP" (Buyer's Part Number).</p> <p>Reference See <i>How to Define and Modify a Code List</i> on page 3 - 62 for more information.</p>
3	<p>Double-click the element for which you need to validate data against a code table (Product/Service ID Qualifier).</p> <p>System response The system displays the Element Properties dialog box.</p>
4	Select the Standard Rule tab to access standard rule options.
5	From the standard rule list, select "Use Code."
6	From the code list, select the code list table that the data in this field is validated against (0235).
7	If you need to specify that, for compliance reasons, the element <i>must</i> contain one of the codes from the specified table (nothing else is valid for that field), select the compliance error check box.
8	Click OK to add this standard rule to the field.

How to Define and Modify a Code List

Procedure Use this procedure steps to define or modify a code list table.

Step	Action
1	From the Edit menu, select Code Lists . System response The system displays the Code Lists dialog box.
2	Do you want to create a new code list? <ul style="list-style-type: none"> • If <i>yes</i>, click New. • If <i>no</i> (you want to edit a code list), select a code list and click Change. System response The system displays the Edit Code List dialog box.
3	In the Table ID box, type the name of the field or element for which this code list table is used.
4	In the Description box, type the description of the field or element for which this code list table is used.
5	Do you want to create a new code? <ul style="list-style-type: none"> • If <i>yes</i>, click New. • If <i>no</i> (you want to edit a code), select a code and click Change. System response The system displays the Edit Code List Entry dialog box.
6	In the Value box, type the actual value of the code.
7	In the Description box, type a description of the code value.
8	Click OK to save the code list entry.
9	Repeat steps 5 through 8 to add more code list entries to the code list table.
10	Click Close to save and exit the Edit Code List dialog box.
11	Click Close to exit the Code Lists dialog box.

How to Delete a Code List or Code List Entry

Procedure: deleting a code list

Use this procedure steps to delete a code list table.

Step	Action
1	From the Edit menu, select Code Lists . System response The system displays the Code Lists dialog box.
2	Select the code list you want to delete.
3	Click Delete to delete the code list table. System response The selected table is deleted without warning.

Procedure: deleting a code list entry


Use this procedure steps to delete a code list entry.

Step	Action
1	From the Edit menu, select Code Lists . System response The system displays the Code Lists dialog box.
2	<ul style="list-style-type: none"> ▶ Select the code list from which you want to delete an entry and ▶ click Change. System response The system displays the Edit Code List Entry dialog box.
3	Select the entry and click Delete to remove the entry. System response The selected entry is deleted without warning.
4	Click OK to save the code list table.

How to Import a Code List

Introduction The Code List Import function enables you to import code lists created for another map and share code lists with other users of Gentran:Server.

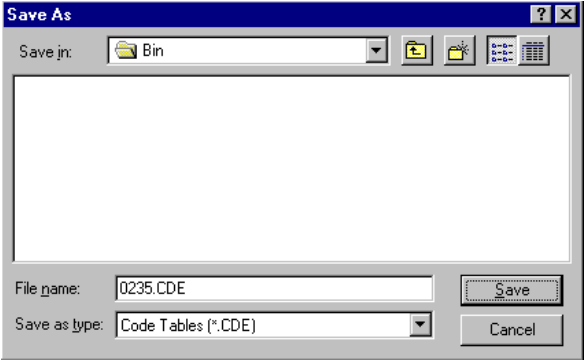
Procedure Use this procedure steps to import a code list table.

Step	Action
1	<p>From the Edit menu, select Code Lists.</p> <p>System response The system displays the Code Lists dialog box.</p>
2	<p>Click Import.</p> <p>System response The system displays the Open dialog box.</p> 
3	<p>Select the location of the code list file.</p> <p>Note The default location is Application Integration install folder (the default is GENSRVNT). The default file extension for code lists is “.CDE.”</p>
4	<p>Select the code list file from the list and click Open.</p> <p>System response The system imports the code list and returns to the Code Lists dialog box. The imported code list is available for your use.</p>
5	<p>Click OK to exit the Code Lists dialog box.</p>

How to Export a Code List

Introduction The Code List Export function enables you to export code lists to file. This enables you to define a code list for one map and use that code list in another map. This function also allows you to share code lists with other users of Gentran:Server.

Procedure Use this procedure steps to export a code list table.

Step	Action
1	<p>From the Edit menu, select Code Lists.</p> <p>System response The system displays the Code Lists dialog box.</p>
2	<p>Select a code list and click Export.</p> <p>System response The system displays the Save As dialog box.</p> 
3	<p>Do you want to change the name?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, type the export file name in the “File name” box. ▶ If <i>no</i>, continue with the next step. <p>Note The file name defaults to the table ID with a “.CDE” file extension. The default location is Application Integration install folder (the default is GENSRVNT).</p>
4	<p>Click Save.</p> <p>System response The system exports the code list and returns to the Code Lists dialog box.</p>
5	<p>Click Close to exit the Code Lists dialog box.</p>

How to Load a Code List Table from the Standard

Procedure Use this procedure steps to load a code list table from the standard.

Step	Action
1	<p>Double-click the element for which you need to use a code table.</p> <p>System response The system displays the Element Properties dialog box.</p> <p>Note The standards provide code list tables only for elements that use them. For example, in the TD4 segment, the TD401 (Special Handling Code) and TD403 (Hazardous Material Class Code) have code tables provided by the standard.</p>
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Code.”
4	<p>Click Edit.</p> <p>System response The system displays the Edit Code List dialog box.</p>
5	<p>Click Load.</p> <p>System response The system prompts you to select a data source name (DSN) from which to access the EDI standard.</p>
6	<p>Select the appropriate DSN and click OK.</p> <p>System response The system displays the Load Code List dialog box.</p>
7	<p>Select either specific codes that you want to load or select the entire list of codes.</p> <ul style="list-style-type: none"> • To select specific codes only, from the Codes in Standard list, highlight each code that you want to be loaded. Click Add to move it to the Codes Selected list. • To load the entire code list, click Add All to select all the codes and move them to the Codes Selected list. <p>Recommendation Only add the codes that you and your trading partners are able to create or accept. Adding all codes in the code table (using the Add All) creates a much larger translation object than if you only use selected codes.</p>
8	Click OK to load the code list.
9	Click Close to exit the Code Lists dialog box.

How to Copy and Paste Code Lists

Introduction The Code List Copy and Paste function enables you to copy code lists to from one map to another.

Procedure Use this procedure steps to copy and paste a code list table.

Step	Action
1	From the Edit menu, select Code Lists . System response The system displays the Code Lists dialog box.
2	Select a code list and click Copy . System response The system copies the code list to the clipboard.
3	Click Close to exit the Code Lists dialog box.
4	Open the map in which you want to use the code list, if the map is not already open.
5	From the Edit menu, select Code Lists . System response The system displays the Code Lists dialog box.
6	Click Paste . System response The system adds the copied code list to this map.
7	Click Close to exit the Code Lists dialog box.

How to Validate Data Against Code List Tables

Procedure Use this procedure steps to validate data against a code list table.

Step	Action
1	<p>Double-click the element for which you need to validate data against a code table.</p> <p>System response The system displays the Element Properties dialog box.</p> <p>Note The standards provide code list tables only for elements that use them. For example, in the TD4 segment, the TD401 (Special Handling Code) and TD403 (Hazardous Material Class Code) have code tables provided by the standard.</p>
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Code.”
4	<p>From the code list, select the code list table that the data in this element is validated against.</p> <p>Note If this list is empty, you need to load a code table.</p> <p>Reference See <i>How to Load a Code List Table from the Standard</i> on page 3 - 66 for more information.</p>
5	If you need to specify (for compliance reasons) that the element <i>must</i> contain one of the codes from the specified table (nothing else is valid for that field), select the compliance error check box.
6	Click OK to add this standard rule to the element.

How to Map Code Item Descriptions

Procedure Use this procedure steps to map a code item description.

Step	Action
1	Double-click the element for which you need to map a code description. System response The system displays the Element Properties dialog box.
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, select “Use Code.”
4	From the code list, select the code list table that the data in this element is validated against. Note If this list is empty, you need to load a code table. Reference See <i>How to Load a Code List Table from the Standard</i> on page 3 - 66 for more information.
5	If you need to specify (for compliance reasons) that the element <i>must</i> contain one of the codes from the specified table (nothing else is valid for that field), select the compliance error check box.
6	From the store description list, the element to which you want the description of the code item (that is used) to be mapped to when the selection is made.
7	Click OK to add this standard rule to the element.

Setting Up the Import Process

Contents

- ▶ Overview 4 - 2
 - ▶ Importing Data 4 - 3
 - ▶ How to Modify System Configuration 4 - 5
 - ▶ How to Create a System Import Map 4 - 6
 - ▶ How to Define the Six-Field Key 4 - 8
 - ▶ How to Define the Alternate Key 4 - 15
 - ▶ How to Compile the System Import Translation Object. 4 - 20
-

Overview

Introduction

The import process is part of the outbound translation process. In the outbound translation process, the system translates your application file format to EDI standard formats, so you can send documents to your partners.

Import translation object

To translate outbound data, you need to create an import translation object and a system import translation object in the Gentran:Server Application Integration subsystem. The import translation object defines how to move data from your application file (flat file definition), which may contain multiple documents, to the EDI standard-formatted documents that your partners expect to receive from you.

System import translation object

You need to create a system import translation object to determine which trading relationship (established in Partner Editor) corresponds to each document in the application file, so the system knows which import translation object to use to process the document.

Importing Data

Introduction During the import process, Gentran:Server imports a specified application file into the Workspace. An import file can contain multiple flat file definitions (documents) that need to be sent to multiple trading partners.

What you can import You can import any type of flat file, including TDF (Transaction Data File). TDF is a flat file representation of the EDI standard document. The flat file characteristics can include multiple record types (more than one record definition) and records delimited by carriage returns/line feeds.

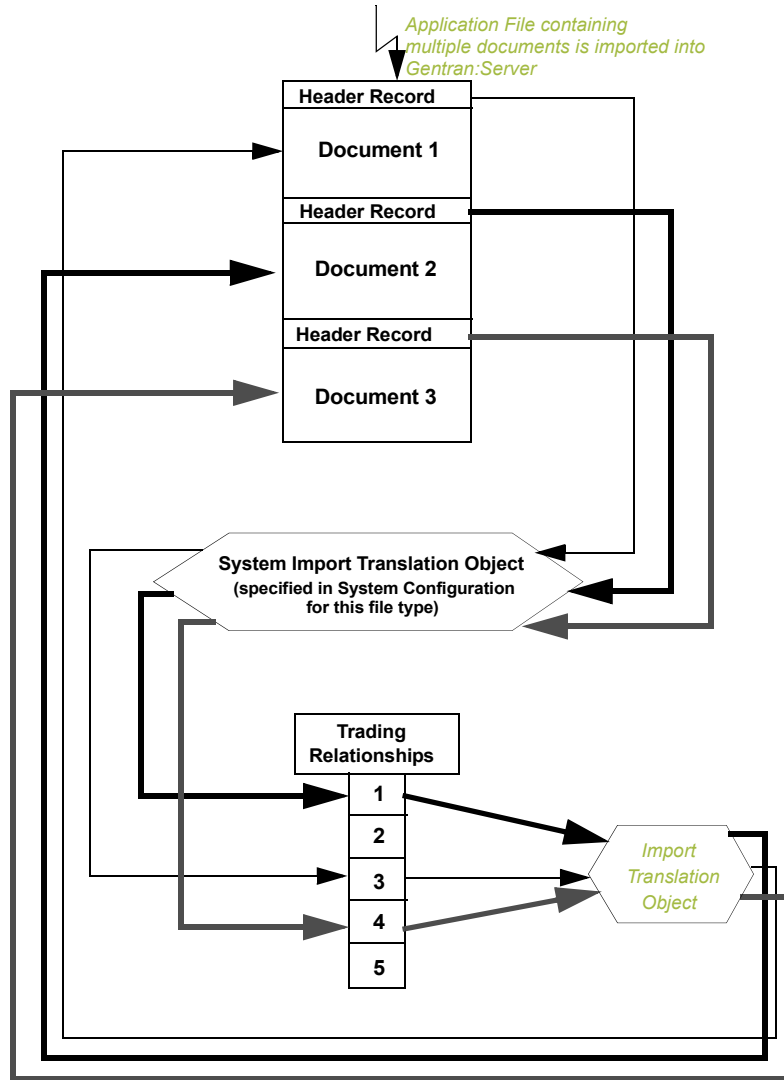
Restrictions You *cannot* import EDI or field-delimited files. The field and record lengths for import files are defined in the Application Integration subsystem application file definition. For example, the maximum length of a field in an import file must not exceed the maximum length (Max) specified on the Field Properties dialog box for that field. And, the maximum length of a record in an import file must not exceed the record length (if specified) on the Positional File Format Properties dialog box for that record. The system default is variable length records. If you specify record delimiters on the Positional File Format Properties dialog box, the records in the import file must begin and end with those delimiters. If you do not specify record delimiters, the default delimiters are carriage return and line feed.

Determining trading relationships The system must determine which trading relationship (established in Partner Editor) corresponds to each document in the application file, so the system knows which import translation object to use to process the document. Each application file must have a corresponding system import map. For each document in the application file, the system import translation object identifies the partner and locates the trading relationship. The trading relationship for that partner must have an import translation object associated with it.

Translating the document After the system import translation object determines which trading relationship corresponds with a document, it ascertains which import translation object is specified in that relationship. The translator uses the import translation object to translate the document. If the document is compliant (valid), it is moved to the Workspace. If the document is not compliant (invalid), it is moved to the ?Out Documents. Then, if there is another document remaining in the application file, the translator repeats this process until all the documents are processed.

Import process diagram

This diagram illustrates the import process (translator not pictured).



How to Modify System Configuration

Introduction

When you import a file, the system checks the Imports tab of the System Configuration program (shipped with Gentran:Server) to verify that the file is matched with a registered system import translation object. Your system administrator may need to add the match to the System Configuration program.

Reference

See How to Edit an Import Specification in the *Administration Guide* for more information on defining or modifying import specifications.

How to Create a System Import Map

Introduction

A system import map is used by the system to find the partner relationship for a document (flat file definition), to determine which import map is used to translate the data. The system import map builds the key that the translator uses to find the partner relationship. The sole function of the system import map is to identify the appropriate partner relationship; the system import map does *not* map any data.

Building a key: method 1

There are two ways to build the key in a system import map. The method that we recommend requires six EDI-specific fields in the header record: partner key, standard, version, transaction set, release (for TRADACOMS only), and test/production status. The combination of these six fields defines a unique key that identifies the appropriate partner relationship. We recommend using this method because it is very flexible. You typically use this method when you are defining your application from scratch and can easily add the EDI-specific fields that are not already present in the header record.

Building a key: method 2

The second method is easier to build because it only requires three fields in the header record (partner key, application ID or application alias value, and test/production status). The combination of these three fields defines a unique key that identifies the appropriate partner relationship. However, because only three fields are combining to build the key, this method is not as flexible as the preferred method. You typically use this method when you are using a legacy (existing) application definition, and you do not want to add EDI-specific information.

Procedure: creating a system import map

Use this procedure to create a system import map.

Step	Action
1	From the File menu, select New . System response The system displays the New Map Wizard.
2	From the type of map list, select System Import.
3	In the name box, type the name of the map.
4	Verify that your name is already entered in the name box and click Next . System response The system displays the New Map Wizard - Input Format dialog box.
5	From the Create a new data format list, select Positional and click Next . System response The system displays the New Map Wizard - Output Format dialog box. (Continued on next page)

(Contd) Step	Action
6	From the Create a new data format list, select Positional and click Next . Note The system actually ignores the Output side of the map and only processes the Input side.
7	Click Finish to create the new system import map (this may take a few seconds). System response The system displays the new map in the Gentran:Server - Application Integration Window.
8	After creating the system import map, you need to define the header record for the system import map. You can use either a six-field key (this is the preferred method) or the alternate key method. Reference Continue with <i>How to Define the Six-Field Key</i> on page 4 - 8.

How to Define the Six-Field Key

Introduction

To use the preferred method of building the system import key, you need to define at least five fields for the header record (partner key, standard, version, transaction, and test/production status). If it is a TRADACOMS map, you must also define a release field. These fields do not have to be in any order or sequence, but they must be part of the header record.

If you want to also define the other fields that the corresponding import map header record contains, you can do so at this time.

Reference

- See *How to Create the First Record* on page 2 - 54 for more information.
- See *How to Create Fields* on page 2 - 64 for more information.

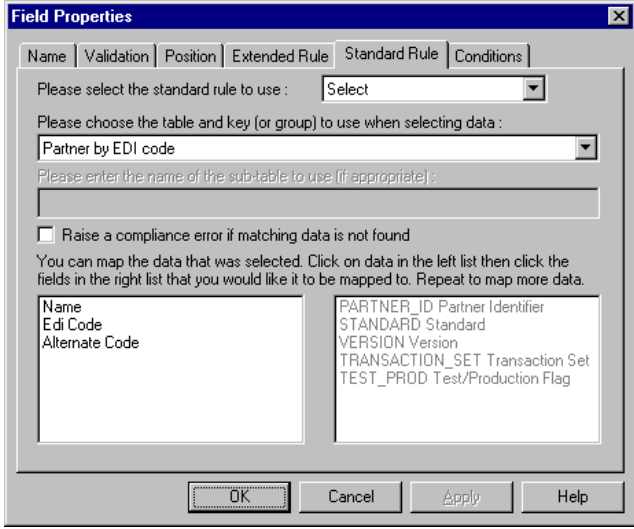
After defining the key

Once you define the header record and fields, you can set up the mapping operations that defines the key that the translator uses to find the partner relationship.

Procedure: defining the partner key

Use this procedure to define the partner key.

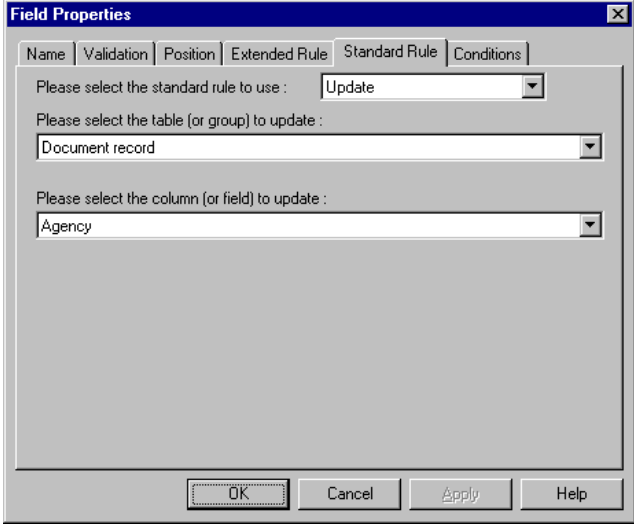
Step	Action
1	Double-click the partner key field. System response The system displays the Field Properties dialog box (Name tab).
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose Select. <i>(Continued on next page)</i>

(Contd) Step	Action
4	<p>From the table and key list, select Partner by EDI Code, Partner by Alternate code, or Partner by Partner Key.</p> <p>Note This indicates that the system updates this field with the indicated partner key.</p> <p>System response This is an example how the Standard Rule tab of the Field Properties dialog box should look.</p> 
5	Click OK to add the standard rule to the partner key field.

**Procedure:
defining the
standard field**

Use this procedure to define the EDI standard field.

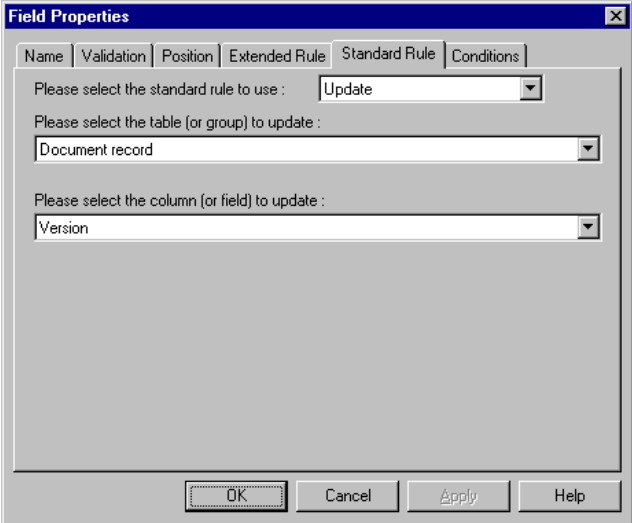
Step	Action
1	<p>Double-click the standard field.</p> <p>System response The system displays the Field Properties dialog box (Name tab).</p>
2	Select the Standard Rule tab to access standard rule options.
3	<p>From the standard rule list, choose Update.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Step	Action
4	<p>From the table list, choose Document record.</p> <p>Note This indicates that you are updating the document record in the Gentran:Server internal system buffer (because the translator is not mapping data at this point in the process).</p>
5	<p>From the column list, choose Agency.</p> <p>Note This indicates that you are updating the agency (standard) field in the system buffer.</p> <p>System response This is an example how the Standard Rule tab of the Field Properties dialog box should look.</p> 
6	Click OK to add the standard rule to the standard field.

**Procedure:
defining the
version field**

Use this procedure to define the version.

Step	Action
1	<p>Double-click the version field.</p> <p>System response The system displays the Field Properties dialog box (Name tab).</p>
2	<p>Select the Standard Rule tab to access standard rule options.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

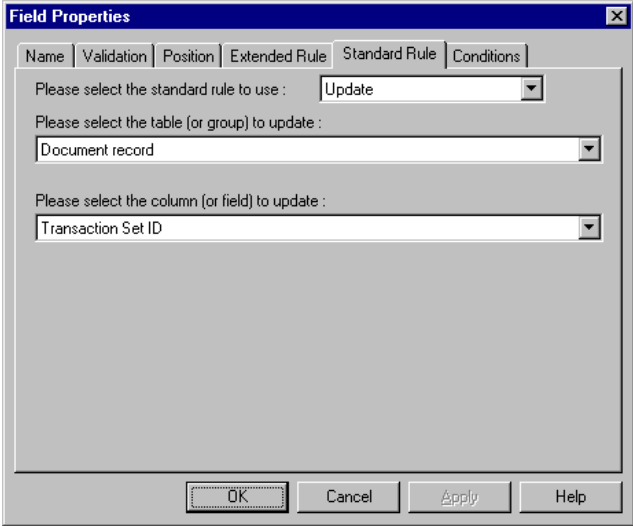
(Contd) Step	Action
3	From the standard rule list, choose Update.
4	From the table list, choose Document record. Note This indicates that you are updating the document record in the Gentran:Server internal system buffer (because the translator is not mapping data at this point in the process).
5	From the column list, choose Version. Note This indicates that you are updating the version field in the system buffer. System response This is an example how the Standard Rule tab of the Field Properties dialog box should look. 
6	Click OK to add the standard rule to the version field.

**Procedure:
defining the
transaction field**

Use this procedure to define the transaction set (document).

Step	Action
1	Double-click the transaction set field. System response The system displays the Field Properties dialog box (Name tab). (Continued on next page)



(Contd) Step	Action
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose Update.
4	<p>From the table list, choose Document record.</p> <p>Note This indicates that you are updating the document record in the Gentran:Server internal system buffer (because the translator is not mapping data at this point in the process).</p>
5	<p>From the column list, choose Transaction Set ID.</p> <p>Note This indicates that you are updating the transaction set field in the system buffer.</p> <p>System response This is an example how the Standard Rule tab of the Field Properties dialog box should look.</p> 
6	Click OK to add the standard rule to the transaction set field.

**Procedure:
defining the
release field**

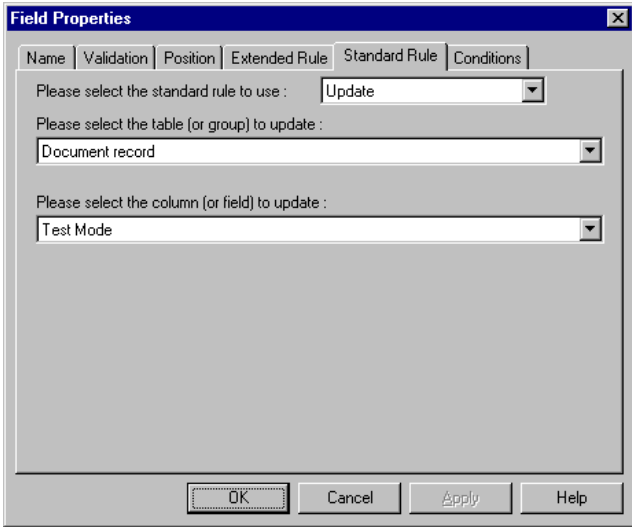
Use this procedure to define the release of the selected standard version (TRADACOMS *only*).

Step	Action
1	Double-click the release field. System response The system displays the Field Properties dialog box (Name tab).
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose Update.
4	From the table list, choose Document record. Note This indicates that you are updating the document record in the Gentran:Server internal system buffer (because the translator is not mapping data at this point in the process).
5	From the column list, choose Release. Note This indicates that you are updating the release field in the system buffer. System response This is an example how the Standard Rule tab of the Field Properties dialog box should look. <div data-bbox="657 1199 1286 1724" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div>
6	Click OK to add the standard rule to the release field.



**Procedure:
defining the test/
production field**

Use this procedure to define the test or production status field.

Step	Action
1	<p>Double-click the test or production field.</p> <p>System response The system displays the Field Properties dialog box (Name tab).</p>
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose Update.
4	<p>From the table list, choose Document record.</p> <p>Note This indicates that you are updating the document record in the Gentran:Server internal system buffer (because the translator is not mapping data at this point in the process).</p>
5	<p>From the column list, choose Test Mode.</p> <p>Note This indicates that you are updating the test field in the system buffer.</p> <p>System response This is an example how the Standard Rule tab of the Field Properties dialog box should look.</p> 
6	Click OK to add the standard rule to the test field.

How to Define the Alternate Key

Introduction

To use the alternate key method of building the system import key (formerly referred to as the three-key method), you need to define (create new or update existing) three or four fields for the header record. You must define a field for partner key, a field for either one or both application ID and application alias value (you only need to use one of these two fields), and a field for test/production status. If you define all four fields, you are using the alternate key method; if you use only one of the application ID or application alias value fields, you are actually using a three-field key. These fields do not have to be in any order or sequence, but they must be part of the header record.

Tip

You do *not* have to define new fields to use this key. You can add the necessary standard rules to three fields that already exist in your application file.

If you want to also define the other fields that the corresponding import map header record contains, you can do so at this time.

Tip

- See *How to Create the First Record* on page 2 - 54 for more information on how to define the header record.
- See *How to Create Fields* on page 2 - 64 for more information on how to define the fields in the record.

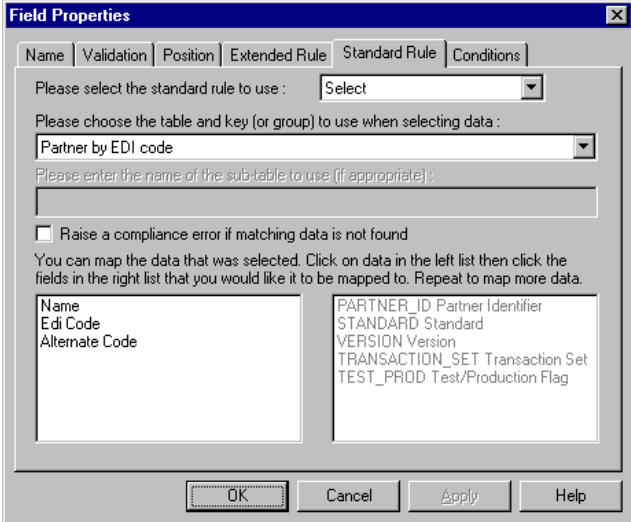
After defining the key

Once you define the header record and fields, you can set up the mapping operations that defines the key that the translator uses to find the partner relationship.

Procedure: defining the partner key

Use this procedure to define the partner key.

Step	Action
1	Double-click the partner key field. System response The system displays the Field Properties dialog box (Name tab).
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose Select. (Continued on next page)

(Contd) Step	Action
4	<p>From the table and key list, select Partner by EDI Code, Partner by Alternate code, or Partner by Partner Key.</p> <p>Note This indicates that the system updates this field with the indicated partner key.</p> <p>System response This is an example how the Standard Rule tab of the Field Properties dialog box should look.</p> 
5	Click OK to add the standard rule to the partner key field.

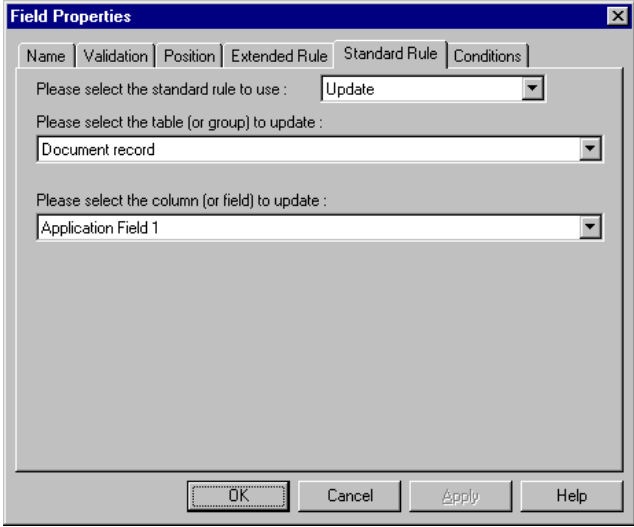
**Procedure:
defining the
application ID
field**

Use this procedure to define the application ID field.

Note

You must define either the application ID field or the Application Alias Value field (three-field key). You may define both of these fields if you wish to use the alternate key method.

Step	Action
1	<p>Double-click the application ID field.</p> <p>System response The system displays the Field Properties dialog box (Name tab).</p>
2	Select the Standard Rule tab to access standard rule options.
3	<p>From the standard rule list, choose Update.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
4	<p>From the table list, choose Document record.</p> <p>Note This indicates that you are updating the document record in the Gentran:Server internal system buffer (because the translator is not mapping data at this point in the process).</p>
5	<p>From the column list, choose Application Field 1.</p> <p>Note This indicates that you are updating an optional key field in the system buffer.</p> <p>System response This is an example how the Standard Rule tab of the Field Properties dialog box should look.</p> 
6	<p>Click OK to add the standard rule to the application ID field.</p>

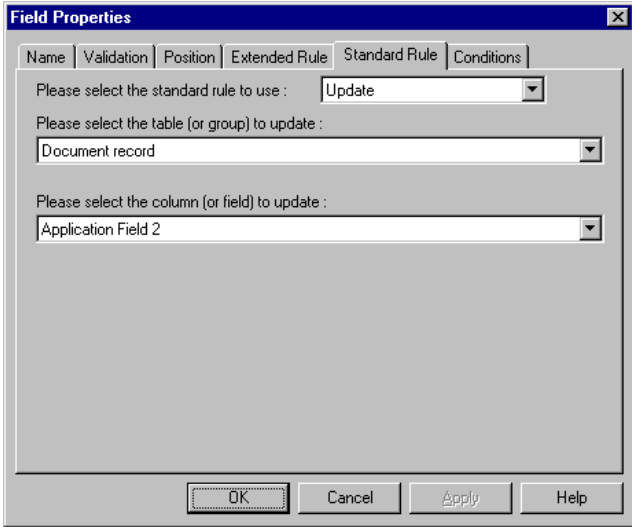


**Procedure:
defining the
application alias
value field**

Use this procedure to define the application alias value field.

Note

You must define either the application ID field or the Application Alias Value field (three-field key). You may define both of these fields if you wish to use the alternate key method.

Step	Action
1	Double-click the application alias value field. System response The system displays the Field Properties dialog box (Name tab).
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose Update.
4	From the table list, choose Document record. Note This indicates that you are updating the document record in the Gentran:Server internal system buffer (because the translator is not mapping data at this point in the process).
5	From the column list, choose Application Field 2. Note This indicates that you are updating an optional key field in the system buffer. System response This is an example how the Standard Rule tab of the Field Properties dialog box should look.
	
6	Click OK to add the standard rule to the application alias value field.

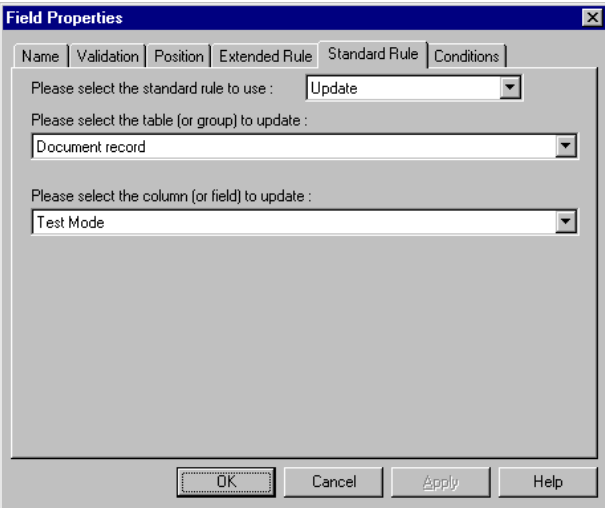
**Procedure:
defining the test/
production field**

If you do not define the test/production status field, the document record is set to production by default. In this situation, the system translation object can only locate the correct relationship if the test mode of the partner relationship is actually set to production.

Recommendation

Define the test/production status field.

Use this procedure to define the test/production status field:

Step	Action
1	Double-click the test or production field. System response The system displays the Field Properties dialog box (Name tab).
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose Update.
4	From the table list, choose Document record. Note This indicates that you are updating the document record in the Gentran:Server internal system buffer (because the translator is not mapping data at this point in the process).
5	From the column list, choose Test Mode. Note This indicates that you are updating the test field in the system buffer. System response This is an example how the Standard Rule tab of the Field Properties dialog box should look. 
6	Click OK to add the standard rule to the test field.



How to Compile the System Import Translation Object

Introduction After you create, define, and save the system import map, you need to compile the map and create a system import translation object.

Procedure Use this procedure to generate the system import translation object.

Step	Action
1	From the File menu, select Compile . System response The system displays the Run-time Translation Object Name dialog box.
2	In the File name box, type the name of the translation object, if necessary. Warning Be very careful not to overlay the source map with the compiled translation object. Use the .TPL file extension to distinguish the translation object.
3	From the Save in list, change the Drive and Folder where the compiled translation object is stored, if necessary
4	Click Save . System response The system compiles the map and creates the system import translation object.
5	The translation object must registered with the Gentran:Server system before you can use it. Reference See How to Register a New Translation Object in the Gentran:Server <i>User's Guide</i> for information on registering a translation object.
6	If your system administrator has not already added this system import translation object to the System Configuration program, do so now. Reference See How to Edit an Import Specification in the <i>Administration Guide</i> for more information.

Setting Up the Export Process

Contents

- ▶ Overview 5 - 2
 - ▶ Inbound Process Prior to Exporting Data 5 - 3
 - ▶ How to Use Supplementary Envelope Information 5 - 5
-

Overview

-
- Introduction** To translate inbound data, you need to create an export map in Gentran:Server (because the system is *exporting to* your application file).
-
- Export map** The export map defines *how* to move data from the EDI standard-formatted documents that your partners send you to your application file (flat file definition). This map also enables you to populate your application file fields with supplementary information from the interchange, group, and transaction set envelopes.
-
- In this chapter** This chapter explains how to set up the export process, including the following topics:
- Inbound process prior to exporting data
 - Using supplementary envelope information
-

Inbound Process Prior to Exporting Data

Introduction

During the export process, the Communicator receives interchanges from your trading partners via a network and passes the interchanges to the translator. The translator uses a system interchange break translation object (shipped with Gentran:Server) to unwrap the interchange envelopes and separate each group into temporary storage. The translator uses a system group break translation object (shipped with Gentran:Server) to unwrap the group envelopes and separate each transaction set into temporary storage. The translator uses a system transaction break translation object (shipped with Gentran:Server) to unwrap the transaction envelopes and separate each document into the ?In Documents in Gentran:Server.

Locating trading relationships

The translator attempts to locate the trading relationship for each document. If a trading relationship is located, the translator then attempts to identify the export, document turnaround, or print translation object associated with that relationship. If the translator does not locate the trading relationship or translation object, the document stays in the ?In Documents. If the translator does locate a trading relationship and translation object, it uses that translation object to compliance check the document. If the document is not compliant with the EDI standard, it stays in the ?In Documents. If the document is compliant with the EDI standard, the translator moves it to the In Documents. From the In Documents, you can print (if there is an associated print translation object), or export valid documents.

Generating acks

If you specify (in the trading relationship) that the system needs to generate a functional acknowledgement (ack) for a document, the translator uses the system acknowledgement translation object (shipped with Gentran:Server) to generate the acknowledgement. If the generated acknowledgement was compliant, the translator moves the acknowledgement to the Out Documents to be sent to the trading partner. If the generated acknowledgement was not compliant or if an error occurred with the acknowledgement translation object, the translator moves the acknowledgement to the ?Out Documents.

Automatic export

If you specified either automatic export or automatic turnaround in the trading relationship, the translator uses the specified export or document turnaround translation object to either export or generate the appropriate response document.

Procedure Use this procedure to set up the export process.

Step	Action
1	Create an export map. Reference See <i>How to Create a New Map</i> on page 2 - 20 for more information on how to create a new map.
2	Register the export map with Gentran:Server. Reference See <i>How to Register a New Translation Object in the Gentran:Server User's Guide</i> for more information on registering a translation object.
3	Create the appropriate inbound trading relationship. Reference See <i>How to Create a New Inbound Relationship in the Gentran:Server User's Guide</i> for more information on creating an inbound relationship.

How to Use Supplementary Envelope Information

Introduction

The Gentran:Server Application Integration subsystem enables you to use a Select standard rule to populate your application file fields (in an export map) with information from the EDI envelopes associated with the documents your trading partner sends you.

This table lists the correspondence between the selections available for the Select standard rule and the envelope information that you want to map to your application field.

IF you select this value from the Map from list...	THEN the system maps this information...
Field 1	Transaction Set ID
Field 2	Document Name
Field 3	Group ID
Field 5	Partner Name
Field 6	Interchange Control Number
Field 7	Group Control Number
Field 8	Document Control Number
Field 9	Received Version
Field 15	Received Agency
Field 16	Used Agency
Field 17	Used Version
Field 22	ISA Test Mode
Field 23	Document Test Mode

Procedure Use this procedure to use supplementary envelope information in your application file.

Step	Action
1	Double-click an existing application field or create a new field. Reference See <i>Using a Fixed-Format File</i> on page 2 - 51 for more information on creating fields. System response The system displays the Field Properties dialog box (Name tab).
2	Select the Standard Rule tab to access standard rule options.
3	From the standard rule list, choose Select.
4	From the table and key list, select Generic envelope segment. Note This enables you to map from any of the fields within the current envelope information, regardless of the EDI standard used.
5	Select the compliance error check box. Note This indicates that a compliance error should be generated if the select does not find a valid entry.
6	From the map from list, choose the envelope information that you want to map to your application field. Note See the table on the previous page for selection definitions. Warning Verify that you select the correct map from selection. Any selection that is not listed in the above table is currently reserved and <i>should not be selected under any circumstances</i> .
7	From the map to list, choose the field to which you want to map the contents of the Map From field. Note The system displays each field from the output side of the map in this list.
8	If you want to map additional envelope information, repeated steps 7 and 8 as many times as necessary. Note A total of eight fields can be mapped using one Select rule.
9	Click OK to add the standard rule to the application field.

Using Extended Rules

Contents	Introducing Extended Rules.....6 - 4
	▶ Overview 6 - 4
	▶ Declarations and Initialization 6 - 5
	▶ Statements 6 - 7
	▶ When Extended Rules are Processed 6 - 8
	Defining Extended Rules.....6 - 12
	▶ Overview 6 - 12
	▶ How to Define a Session Rule..... 6 - 13
	▶ How to Define a Map Component Rule 6 - 15
	Extended Rule Syntax.....6 - 17
	▶ Overview 6 - 17
	▶ Keywords and Commands 6 - 18
	▶ Operators 6 - 20
	▶ Symbols..... 6 - 21
	Common Statements and Examples6 - 25
	▶ Overview 6 - 25
	▶ Assignment 6 - 26
	▶ Datetime Expressions 6 - 27
	▶ Conditional Logic 6 - 29
	▶ String Conditions and Functions..... 6 - 31
	▶ Numerical Functions..... 6 - 33
	▶ Raise Compliance Error Function (CERROR)..... 6 - 34
	▶ Remove Field Value Function (EMPTY) 6 - 35
	▶ Existence of Data Function (EXIST) 6 - 36
	▶ Count Function (COUNT) 6 - 37
	▶ Delete Function (DELETE) 6 - 38
	▶ File Pointer Functions (FSEEK, FTELL) 6 - 39
	▶ Block of Data Functions (READBLOCK, WRITEBLOCK)..... 6 - 40
	▶ Select Function (SELECT) 6 - 41

▶ Update Function (UPDATE).....	6 - 42
▶ Insert Function (INSERT).....	6 - 43
▶ User Exit Function (EXEC).....	6 - 44
▶ ActiveX and User Exit Functions.....	6 - 45
Alphabetic Language Reference	6 - 47
▶ Overview.....	6 - 47
▶ atoi.....	6 - 48
▶ aton.....	6 - 49
▶ auditlog.....	6 - 50
▶ begin.....	6 - 52
▶ break.....	6 - 53
▶ cerror.....	6 - 54
▶ concat.....	6 - 61
▶ continue.....	6 - 62
▶ count.....	6 - 63
▶ createobject.....	6 - 64
▶ date.....	6 - 65
▶ delete.....	6 - 67
▶ deleteobject.....	6 - 68
▶ empty.....	6 - 69
▶ end.....	6 - 70
▶ exec.....	6 - 71
▶ exist.....	6 - 72
▶ fseek.....	6 - 73
▶ ftell.....	6 - 74
▶ get.....	6 - 75
▶ getiid.....	6 - 76
▶ if then...else.....	6 - 77
▶ index.....	6 - 78
▶ insert.....	6 - 79
▶ left.....	6 - 81
▶ len.....	6 - 82
▶ messagebox.....	6 - 83
▶ mid.....	6 - 85
▶ ntoa.....	6 - 86
▶ param.....	6 - 87
▶ queryobject.....	6 - 88
▶ readblock.....	6 - 89
▶ readbytes.....	6 - 90
▶ right.....	6 - 91
▶ select.....	6 - 92
▶ set.....	6 - 93

- ▶ strdate 6 - 94
- ▶ strstr 6 - 96
- ▶ unreadblock 6 - 97
- ▶ update 6 - 98
- ▶ while...do 6 - 99
- ▶ winexec 6 - 100
- ▶ writeblock 6 - 102
- ▶ writebytes 6 - 103
- ▶ Select and Update Available Options 6 - 104



Introducing Extended Rules

Overview

In this chapter

This chapter explains how to use extended rules, including the following topics:

- What rules are and when they are processed
 - Defining extended rules
 - Extended rule syntax
 - Alphabetic language reference guide
-

What are extended rules?

Extended rules enable you to use a Gentran:Server proprietary programming language to perform virtually any mapping operation you require.

Uses of extended rules

You can use these rules to define more complex translations than are available through the link function (simple mapping) and standard rules. You can use extended rules to define operations that are not possible using standard rules.

You define extended rules with the Gentran:Server proprietary programming language. This is a full programming language that gives you access to the entire Gentran:Server internal storage area.

What comprises an extended rule?

Any variables that are not already defined as part of the input or output specification of the map, that you use in a rule must be declared before you use those variables.

Therefore, an extended rule consists of two sections, a *declarations* section followed by a *statements* section.

The declarations section is only required if you use additional variables. This is where you declare the names and types of any variables you use either in this rule or in any other rule that is within the scope of this rule.

The *statements* section is where you define the actions that you want the rule to execute.

Declarations and Initialization

Introduction

The variables that you define in the declarations section are used to store values. Variables consist of a name and a data type. Variable names can include alphanumeric characters and the colon (:) and underscore (_). The first character in a variables name may not be a numeric. All variable names are case-sensitive.

Note

- A declaration must be terminated with a semicolon (;). To improve readability, you typically include a blank link in between the declaration and statement sections.
- The Translator does not initialize extended rule variables. You must initialize all variables after declaring them. Variables that are not initialized can cause incorrect results or translation failures.

Supported data types

This table describes data types that are supported by extended rules.

Data Type	Description	Example
Integer	a whole number with no decimal component.	Declare i as an integer and initialize. integer i; //Declaration i = 0; //Initialization of 'i'
Real	a whole number that may have a decimal component.	Declare r as a real number and initialize. real r; //Declaration r = 0; //Initialization of 'r'
String	contains one or more printable characters.	Declare s as a 20-character string and initialize. string[20] s; //Declaration s = ""; //Initialization of 's'
Datetime	contains a date or time.	Declare d as a date or time and initialize. datetime d; //Declaration d = date(0,0,0); //Initialization of 'd'

(Continued on next page)

Data Type	Description	Example
Array	defines a table of multiple occurrences of a single data type.	<p>Declare a as an array of 10 integers and initialize.</p> <pre>integer a[10]; //Declaration integer i; //Declaration of 'i', which is //used to initialize array 'a' i = 1; //Initialization of 'i' //Initialization of the variable array 'a' while i < 11 do begin a[i] = 0; i = i + 1; end</pre> <p>Declare p as an array of 50 10-character strings and initialize.</p> <pre>string[10] p[50]; //Declaration integer i; //Declaration of 'i', which is //used to initialize array 'p' i = 1; //Initialization of 'i' //Initialization of the variable array 'p' while i < 51 do begin p[i] = ""; i = i + 1; end</pre>
Object	is used for user exits; it exposes the internal functions of an ActiveX Automation Server to Gentrans:Server.	<p>Declare ob as an object and initialize.</p> <pre>object ob; //Declaration ob = CreateObject("ADODB.Connections"); //Initialization of 'ob'</pre>

Statements

Introduction

The actual work performed by an extended rule is defined in the statements section. A rule consists of a statement or a combination of statements (to perform more complex operations). A statement is a single operation that consists of a combination of expressions, keywords, commands, operators, and symbols.

What is an expression?

An *expression* is a logical unit (e.g., $A = B$ or $A + B$) that the system evaluates. The statements section consists of a sensible combination of keywords, operators, and symbols.

Reference

Each of these components is explained in detail in *Extended Rule Syntax* on page 6 - 17.

When Extended Rules are Processed

Introduction

You can specify pre- and post-session rules on the Session Rules dialog box. Pre-session extended rules are processed before the translation object, and are in scope for every extended rule defined in the translation object. Post-session rules are executed after the translation object is processed.

Levels of extended rules

You can attach extended rules to three different levels of map components:

- EDI file and application file groups, using the Loop Level Extended Rules dialog box.
- Groups, sub-groups, repeating records, and repeating segments, using the Loop Level Extended Rule dialog box.
- Fields and elements, using the Field Level Extended Rule dialog box.

Scope

The *scope* of an extended rule determines which variables are accessible from within a given extended rule. The scope varies depending on the current state of the map. This list defines the scope of an extended rule.

- Pre-session extended rules (defined on the Session Rules dialog box) are in scope for every rule in the translation object.
- On Begin extended rules (defined on the Loop Level Extended Rules dialog box) are in scope until the conclusion of its companion On End rule (also defined on the Loop Level Extended Rules dialog box).
- Field level extended rules are only in scope for the duration of the field or element.

An extended rule attached to the current map component depends on the type and state of the map component.

Example

If a current group to which the rule is attached is subordinate to another group, the parent group is automatically in scope for the duration of the entire child group, and the current hierarchical structure is also in scope for the duration of the child by using an addressing method that is explained on 6 - 21.

An extended rule that is attached to a field or element is only in scope for the duration of the field or element. Field level extended rules are always processed after standard rules.

A variable is considered to be “in scope” if it was declared in the current rule, in the On Begin rule of a group that contains the current map component, or in the Pre-Session rule.

(Continued on next page)

Scope (contd)

The translator builds the data storage area for a map based on the structure of the Input side of the map (the source side of the map). Therefore, extended rules address the map based on the hierarchy of the Input side. When you use extended rules, you must be careful to always address the Input side of the map so the translator can locate the map component that the rule accesses. On the Output side of a map, extended rules only have access to the current record and the entire Input side of the map. However, from the Input side of a map, you have access to the entire file structure. As long as you address the source end of a link, you can write to any field on the Output side, even fields that have already been written.

Input rule processing

The translator processes the Input side of the map first and then the Output side. This table lists the sequence of how the translator processes rules on the Input side of a map.

Step	Action
1	Load the Input definition.
2	Read the Input file.
3	Determine if data is present for the first/next group and then run the group On_Begin rule, if present.
4	Load each field within the first/next record within the current group and then execute field level rules (for each field) in the following sequence: <ul style="list-style-type: none"> • standard rules • extended rules
5	At the end of the group, execute the On_End rule, if present.
6	Repeat steps 2 - 5 for each group in the Input file.

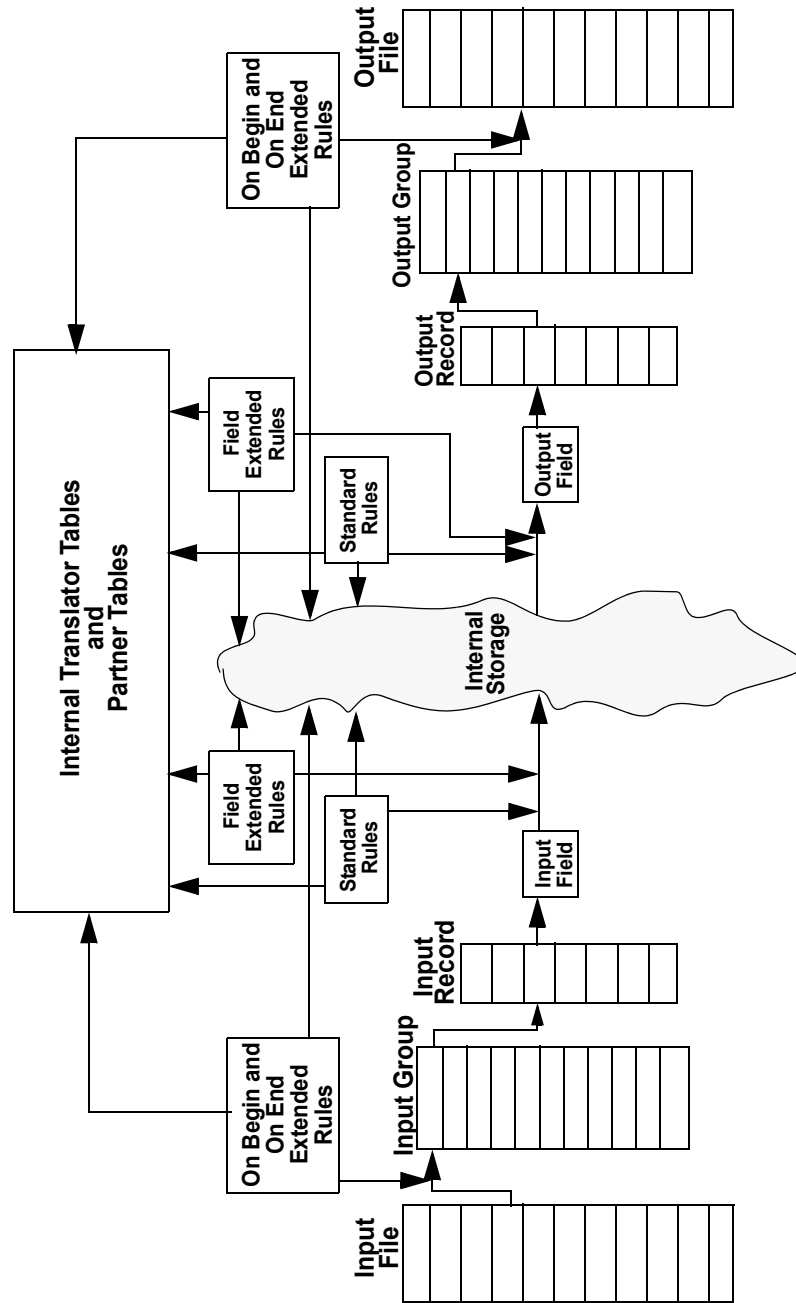
Output rule processing

This table lists the sequence of how the translator processes rules on the Output side of a map.

Step	Action
1	Verify whether or not data exists for the first/next record.
2	If the record is the first record of a group, run On_Begin rule, if present.
3	For each field in the record, execute field level rules (for each field) in the following sequence: <ul style="list-style-type: none">▶ Run standard rules▶ Run extended rules
4	Format data according to specified field properties (on Field Properties dialog box).
5	Write the record to the Output file.
6	At the end of the group, execute On_End rule, if present.
7	Repeat steps 1 - 6 for each record in the Output file.
8	Create or update the document entry in the database.

Rule processing diagram

This diagram illustrates when loop level (On Begin and On End) extended rules and field level extended rules are processed in relation to the system process flow.



Defining Extended Rules

Overview

Introduction

The map component that an extended rule accesses depends on what you want the scope of the rule to be. You also need to determine when you want the rule to be executed (e.g., before or after the map component is processed).

The process you need to follow to define an extended rule varies slightly, depending on whether you are defining a session rule or a rule for a map component. You can define extended rules to access three different levels of map components.

- the entire session (input and output sides of the map)
 - looping map components (groups, segments, etc.)
 - fields
-

How to Define a Session Rule

Introduction

Pre-session rules are used to define variables that must have global scope (can be accessed from any other extended rule in the map). Pre-session extended rules are processed before the translation object, and are in scope for every extended rule defined in the translation object.

Post-session rules are executed after the translation object is processed and thus have no permanent scope.

You can define both a Pre-session and a Post-session rule for a given session.

Procedure

Use this procedure to define a session rule.

Step	Action
1	From the Edit menu, select Session Rules . System response The system displays the Session Level Extended Rules dialog box.
2	Do you want to define a pre-session rule? <ul style="list-style-type: none"> • If <i>yes</i>, click the Pre-session option. • If <i>no</i> (you want to define a Post-session rule), click the Post-session option.
3	In the Editor list, type the extended rule. Note The Session Level Extended Rules dialog box contains line and character number (within a line) indicators. These indicators, which are displayed to the lower right of the Editor box (the first indicator references the line number and the second references the character number), enable you to easily debug compile errors.
4	Do you want to check the rule for errors? <ul style="list-style-type: none"> • If <i>yes</i>, click Compile to compile the extended rule. System response Any warnings or errors are displayed in the Errors list. Double-click an error to instantly navigate to the line containing the error. <ul style="list-style-type: none"> • If <i>no</i>, continue with step 6. Note The Compile function gives you immediate feedback about the accuracy of your rule. The rule is compiled when you compile the entire translation object. (Continued on next page)

(Contd) Step	Action
5	Correct any errors that the system flagged and click Compile again. Note Repeat this process until there are no errors generated.
6	Click OK to add the extended rule.

How to Define a Map Component Rule

Procedure Use this procedure to define an extended rule for a map component.

Step	Action
1	<p>Right-click the map component and select Rules (Extended Rule if the map component is a field, element, or TFD) from the shortcut menu.</p> <p>System response</p> <ul style="list-style-type: none"> • The system displays the Loop Level Extended Rules dialog box if the map component is an EDI file, application file, group, subgroup, repeating record, or repeating segment. • The system displays the Field Level Extended Rule dialog box if the map component is a field or element.
2	<p>Do you want the extended rule to be executed before the system processes the map component?</p> <ul style="list-style-type: none"> • If <i>yes</i>, select the On Begin option. • If <i>no</i> (you want the rule to be executed when the system concludes its processing of that map component), select the On End option. <p>Note You can define both an On Begin and an On End rule for the a single map component.</p>
3	<p>In the Editor list, type the extended rule.</p> <p>Reference See <i>Alphabetic Language Reference</i> on page 6 - 47 for more information on rule syntax.</p> <p>Note The Extended Rules dialog boxes contain line and character number (within a line) indicators. These indicators, which are displayed to the lower right of the Editor box (the first indicator references the line number and the second references the character number), enable you to easily debug compile errors.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Step	Action
4	<p>Do you want to check the rule for errors?</p> <ul style="list-style-type: none"> ▶ If <i>yes</i>, click Compile to compile the extended rule. <p>System response Any warnings or errors are displayed in the Errors list. Double-click an error to instantly navigate to the line containing the error.</p> <ul style="list-style-type: none"> ▶ If <i>no</i>, continue with step 6. <p>Note The Compile function gives you immediate feedback about the accuracy of your rule. The rule is compiled when you compile the entire translation object.</p>
5	<p>Correct any errors that the system flagged and click Compile again.</p> <p>Note Repeat this process until there are no errors generated.</p>
6	<p>Click OK to add the extended rule.</p> <p>Note When an element contains an extended rule a black asterisk appears to the right of the element icon.</p>

Extended Rule Syntax

Overview

Introduction

The statements section of an extended rule consists of a sensible combination of keywords, operators, and symbols. The correct syntax for each of these component is explained below.

Note

You use spaces and operators to separate keywords and symbols. You cannot string two keywords sequentially together without an operator.

Keywords and Commands

Introduction

A *keyword* is a fixed defined use of a word that indicates how the programming language should be interpreted. There are two types of keywords. The first type of keyword controls the flow of execution of the defined rule. These keywords are used to evaluate conditions and perform looping operations.

The second type of keyword is a *command*. Commands perform actions on variables and are responsible for the movement of data.

List of keywords

This is a list of Gentran:Server execution control keywords.

- ▶ IF
 - ▶ THEN
 - ▶ ELSE
 - ▶ BEGIN
 - ▶ END
 - ▶ WHILE
 - ▶ DO
 - ▶ CONTINUE
 - ▶ BREAK
-

List of commands

This is a list of Gentran:Server commands.

- ▶ AUDITLOG
 - ▶ GET
 - ▶ SET
 - ▶ STRDATE
 - ▶ CONCAT
 - ▶ LEN
 - ▶ ATOI
 - ▶ ATON
 - ▶ CERROR
 - ▶ EMPTY
 - ▶ EXIST
 - ▶ INDEX
 - ▶ NTOA
 - ▶ COUNT
-

(Continued on next page)

**List of commands
(contd)**

- ▶ DELETE
- ▶ FSEEK
- ▶ FTELL
- ▶ READBLOCK
- ▶ UNREADBLOCK
- ▶ READBYTES
- ▶ LEFT
- ▶ MID
- ▶ RIGHT
- ▶ SELECT
- ▶ UPDATE
- ▶ INSERT
- ▶ WRITEBLOCK
- ▶ WRITEBYTES
- ▶ CREATEOBJECT
- ▶ DELETEOBJECT
- ▶ QUERYOBJECT
- ▶ GETIID
- ▶ WINEXEC
- ▶ DATE
- ▶ EXEC
- ▶ PARAM

Note

A statement must be terminated with a semicolon (;).

Operators

Introduction Operators define the simplest operation in an expression.

List of keywords This table lists the operators that can be used in extended rules and their functions.

Part	Function
+	addition, concatenation
-	subtraction
*	multiplication
/	division
=	assignment, equality
>	greater-than
<	less-than
>=	greater-than or equal to
<=	less-than or equal to
!=	not equal to
!	logical not
&	logical and
	logical or
<<	date modification

Symbols

Introduction

Operations are performed on symbols. The symbols that you can use in Gentran:Server extended rules are variables, constants, map components/internal storage, arrays, and accumulators.

You can address existing map components and you also have the ability to create additional instances of map components, as long as the map component is originally defined in internal storage.

Example

You can use this function to create extra line items when one line item field is already defined in internal storage.

Symbol syntax

You must address each type of symbol in the proper syntax.

String constant

To address a string constant, you must enclose the constant value in quotes:

```
#fieldname = "HDR";
```

where HDR is the constant value

Addressing or creating a field in internal storage

To address a field or create a field in internal storage, within the scope of the current mapping action, the syntax is #FIELD_NAME

```
#field_1 = 2;
```

where 2 is a numeric constant value

Addressing or creating a field in a group

To address a field within a group or create a field within a group in internal storage within the scope of the current hierarchy, the syntax is \$GROUP.#FIELD_NAME

```
$N1.#0234
```

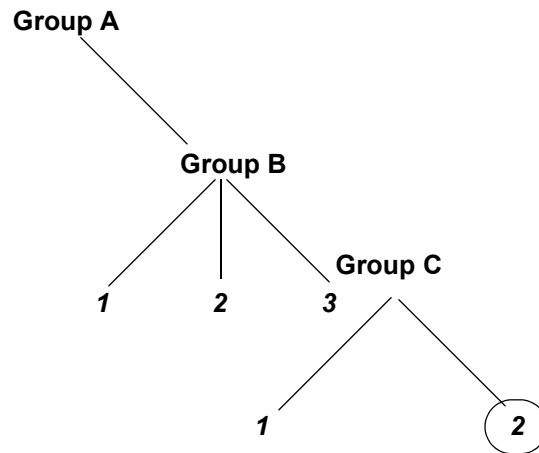
(Continued on next page)

**Symbol syntax
(contd.)****Addressing or creating a group in internal storage**

To fully address a group in the entire internal storage area or create a group in internal storage, the syntax is \$LOOP[index1][index2][index3] where the index entries indicate the hierarchical structure of the loop and enable you to address specific instances of a group:

\$Group_C[3][2].#Field_2

where you are specifying the second instance of Group_C within the third instance of Group_B:

**Addressing an array**

To address an array (of any type), you address each element of the array individually. For example, if array_1 is an array (of integers) and is declared:

integer array_1[5]

With variables 0 through 4, each element of the array is addressed individually as follows:

```

array_1[0]
array_1[1]
array_1[2]
array_1[3]
array_1[4]

```

Accessing an accumulator

An accumulator can be accessed in the same manner as variables or internal storage. To address an accumulator, use the syntax accum(n) where “n” is the number (not the name) of the accumulator:

```
accum(2) = 5;
```

(Continued on next page)

Symbol syntax (contd.)

Accessing repeating elements

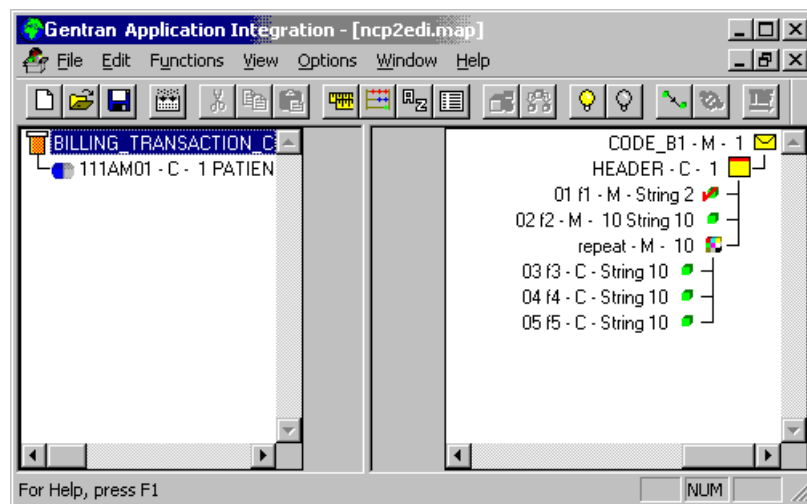
You can access a specific occurrence of a repeating element (for EDI data) and access a specific occurrence of a field within a repeating composite (for EDI data).

This is the syntax for accessing a specific occurrence of a repeating field and a field within a repeating composite.

```
field_name[index_variable] = string;
```

where: integer_variable = integer variable that indicates the specific occurrence of a repeating field or field within a repeating composite.

This is an example of accessing a specific occurrence of a repeating field and a field within a repeating composite.



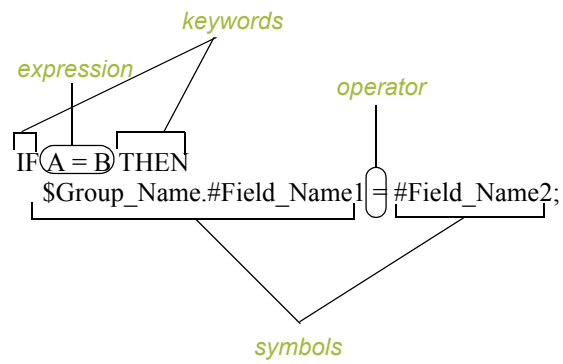
```
string [32]strMsg;

strMsg = "Test";
#f2[1] = strMsg;
//access single repeating field

#f3[1] = strMsg;
//access a field within a repeating composite
//The rule assigns a string value to 2 different fields, which are
//displayed in the diagram above, #f2 and #f3 -- #f2 is a single
//repeating field that can loop up to 10 times and #f3 is a field
//within a repeating composite where the composite can loop up to
//10 times.
```

Example of a simple statement

This diagram illustrates an example of a simple statement.



Common Statements and Examples

Overview

Introduction

This section lists examples of some of the common statements.

- assignment
 - datetime expressions
 - conditional logic
 - string conditions and functions
 - numerical functions
 - raise compliance error function
 - remove field value function
 - existence of data function
-

Assignment

Introduction

The assignment statement is the most powerful and most often used extended rule statement. In the most simple form, it is written as follows:

```
variable=expression
```

However, you can use this statement in more flexible and complex ways, defined as follows:

```
numeric_variable=numeric_expression  
numeric_field=numeric_expression  
string_variable=string_expression  
string_field=string_expression  
datetime_variable=datetime_expression  
datetime_field=datetime_expression
```

Numeric expression

A numeric expression can consist of numbers, numeric fields, numeric variables, and numeric functions combined with the standard arithmetic operators.

String expression

A string expression can consist of string constants, string fields, string variables, and string functions concatenated with the “+” operator.

Datetime expression

A datetime expression can consist of a datetime constant, datetime field or datetime variable.

Examples

Some examples of assignment expressions are as follows:

```
a = 5;  
a = b + c;  
s = "hello";  
s = s + "world";
```

Datetime Expressions

Introduction

Datetime expressions consist of a datetime variable and (optionally) datetime modifiers. Datetime expressions can be written using datetime constants if you are using the standard syntax, as follows:

```
year/month/day
hour:minute:second
year/month/day/hour:minute:second
```

Datetime expressions can also be written with datetime fields, variables, or using *date* and *time* functions.

Date syntax

Date functions are written as follows (month specified as 1-12):

```
datetime d;

d = date(1995, 4, 6);
d = date(1995, 4, 6, 12, 0);
d = date("%y/%m/%d", "95/4/6");
```

The **d = date("%y/%m/%d", "95/4/6");** format enables you to convert any string format type into a datetime format type by indicating a format mask ("**%y/%m/%d**") along with the string ("**01/4/6**") you want to convert. You use this function if you are using non-standard syntax and need to specify the syntax you are using.

<< operator

You can use the << operator to modify your datetime variable by adding time increments (e.g., days, weeks, years). For example:

```
datetime d;
d=d <<weeks(2);
//This adds 2 weeks to d.
```

Time syntax

Time functions are written as follows:

```
d = time(12, 0);
d = time(12, 0, 59);
```

You can use the << operator to modify your datetime variable by adding time increments (e.g., seconds, minutes, years). For example:

```
datetime d;
d=d <<seconds(1);
//This adds 1 second to d.
```

Get and set syntax

The *get* and *set* functions enable you to access (get) or modify (set) individual components of a datetime type. These functions are used as follows:

```
integer a;  
datetime d;  
a = get days (d);  
a = get hours (d);  
set hours (d,a);  
set days (d,a);
```

Conditional Logic

Introduction Gentran:Server uses conditional logic to test conditions and then, depending on the results of the test, perform different operations. Conditions can be nested to any level. Do not end conditions with a semicolon (;) – this terminating syntax is necessary for statements *only*.

**IF...THEN...
ELSE** You can use the IF/THEN keywords to execute one or more statements conditionally. The condition is typically a comparison, but it can be any expression that concludes with a numeric value. Gentran:Server interprets the value as either *true* or *false*. The system interprets a zero value as false and a nonzero value as true.

Note

If you include more than one statement in the body of an IF/THEN loop, you must surround the statements with the BEGIN/END keywords. If you only use a single statement, you can omit the BEGIN and END.

Gentran:Server evaluates the IF/THEN condition, and if it is true, the system executes all the statements that follow the THEN keyword. If the condition is false, none of the statements following THEN are executed.

You can use the ELSE keyword in conjunction with IF/THEN to define several blocks of statements, one of which is executed. Gentran:Server tests the first IF/THEN condition. If the condition is false, the system proceeds to test each sequential condition until it finds one that is true. The system executes the corresponding block of statements for the true condition. If none of the IF/THEN conditions are true, the system executes the statements following the ELSE keyword.

Syntax

```
IF condition THEN
BEGIN
    statement1;
    statement2;
END
ELSE
BEGIN
    statement3;
    statement4;
END
```

Example An example of when you may use conditional logic is if you need to evaluate whether an N1 or NAD group contains billing or shipping information (this depends on the qualifier that a field in the group contains), and then map that information to the appropriate application fields.

For this example, you need to add an On End extended rule to the N1/NAD. The rule is executed when the group terminates. An example of the syntax of the rule follows:

```
IF #0098 = "BT" THEN
BEGIN
  $Group_Name.#BILLTONAME = #0093;
  $Group_Name.#BILLTOADDR1 = #0166;
  $Group_Name.#BILLTOADDR2 = #0166:2;
  $Group_Name.#BILLTOCITY = #0019;
  $Group_Name.#BILLTOSTATE = #0156;
  $Group_Name.#BILLTOPCODE = #0116;
END
IF #0098 = "ST" THEN
BEGIN
  $Group_Name.#SHIPTONAME = #0093;
  $Group_Name.#SHIPTOADDR1 = #0166;
  $Group_Name.#SHIPTOADDR2 = #0166:2;
  $Group_Name.#SHIPTOCITY = #0019;
  $Group_Name.#SHIPTOSTATE = #0156;
  $Group_Name.#SHIPTOPCODE = #0116;
END
```

String Conditions and Functions

Introduction

You can use string conditions in IF/THEN and IF/THEN/ELSE statements to perform comparisons between strings. Examples of the syntax are as follows:

```
IF s1 = s2 THEN
IF s1 < s2 THEN
IF s1 > s2 THEN
```

The following string functions are also available for you to use:

- left
- right
- mid
- strdate
- concat
- strstr

Left, right, mid syntax

The *left*, *right*, and *mid* functions enable you to extract substrings from a string. The left function extracts a specified number of characters from the left of the string variable or field and returns the result as a string. The right function extracts a specified number of character from the right of the string variable and returns the result as a string. The mid function extracts from a specified position in the string to the right, for a specified number of characters. This is an example of how the statements are used.

```
string[10] s;
string[3] s1;
string[3] s2;
string[4] s3;
string[7] s4;

s = "abcdefghij";
s1 = left(s,3);
s2 = right(s,3);
s3 = mid(s,3,4);
```

Strdate syntax

The *strdate* function converts a datetime type into a string using a format that you specify. This function allows you to include static characters such as a slash (/), which gives you access to full date support.

Syntax

```
datetime d;
string[8] s;

strdate(d, "%y/%m/%d", s);
```

Concat syntax

The *concat* function concatenates a specified number of characters from one string onto the end of another string. The following example demonstrates the syntax to concatenate five characters from string “s2” onto the end of string “s1”:

```
string[10] s1,s2;  
concat(s1,s2,5);
```

Strstr syntax

The *strstr* function finds a substring inside a string. This function returns the position of the first instance of the designated substring. If this function does not find the specified substring inside the string, it returns a value of -1.

Syntax

integer d;

```
d = strstr("hello", "el");
```

Numerical Functions

Introduction

The numerical functions enable you to convert one data type to another. The numerical functions that are available for you to use are the following:

- `len`
 - `atoi`
 - `atof`
 - `ntoa`
-

Len syntax

The `len` function counts and returns the number of characters in a string.

Example

```
integer a;  
a = len("hello");
```

Atoi, atof, ntoa syntax

The `atoi` function converts strings into integers.

The `atof` function converts string into real numbers.

The `ntoa` function converts integers and real numbers into strings.

Example

```
integer a;  
real b;  
string[8] s;  
a = atoi("5");  
b = atof("5.5");  
ntoa(5.5, s);
```

Raise Compliance Error Function (CERROR)

Introduction

The *cerror* function raises a compliance error and reports the target statement (the statement you specify) on the translation report.

Syntax

```
cerror(100,$ILD[0][1][1].#VATC);
```

The above example raises compliance error 100 on the VATC element of the specified instance of the ILD group. You typically specify this function as a action to be performed if a condition is false.

Remove Field Value Function (EMPTY)

Introduction

The *empty* function sets the value of a field in internal storage to null. This function is not the same as setting the value of a field to a zero length string (" ") or to zero.

Syntax

```
empty($ILD[0][1][1].#VATC);
```

The above example sets the value of the specified instance of the VATC element to null. This function is typically used to prevent output to the specified field.

Existence of Data Function (EXIST)

Introduction

The *exist* function tests to determine if a field is empty (null). It returns a non-zero (true) value if there is data in a specified field in internal storage. If data is not present in the specified field, this function returns a zero (false) value. This function is typically used as a part of a condition.

Syntax

```
IF exist($ILD[0][1][1].#VATC) THEN
```

The above example returns a non-zero value if the condition is true (data is present in the specified instance of the VATC element). A zero value is returned if the condition is false (no data is present in the specified instance of the VATC element).

There are some situations when the IF EXIST returns a non-zero (true) value whether or not the condition is true. You can work around this by only using IF EXIST for date- and number-type fields. Be certain that all references to the field which is interrogated are nested within the IF EXIST begin block. For string-type fields, use the following format:

```
IF Field1 = " "
```

Count Function (COUNT)

Introduction

The *count* function returns the total number of times a group (or repeating record) has iterated.

Syntax

```
count (GROUPNAME [3] [*]) ;
```

The above example returns the total iterations of the GROUPNAME group within the third iteration of the parent group.

Delete Function (DELETE)

Introduction The *delete* function deletes the specified iteration of a repeating record or group.

Syntax

```
delete (GROUPNAME [iteration]);
```

File Pointer Functions (FSEEK, FTELL)

Introduction

The *fseek* function moves the file pointer to a new location which is a specified number of bytes (offset) from the designated point of origin in the file (the point of origin may be the beginning of the file or relative to either the end of the file or the current position).

To invoke the *fseek* function against the input file, the value for *current_file* is "0." To invoke the *fseek* function against the output file, the value for *current_file* is "1." The *fseek* function is typically only used in conjunction with the *ftell*, *readblock*, and *writeblock* functions.

Syntax

```
fseek(current_file, offset, origin);
```

The *ftell* function obtains the current position of the file pointer and returns it as an integer. To invoke the *ftell* function against the input file, the value for *current_file* is "0." To invoke the *ftell* function against the output file, the value for *current_file* is "1." The *fseek* function is typically only used in conjunction with the *fseek*, *readblock*, and *writeblock* functions.

Syntax

```
numeric_variable = ftell(current_file);
```

Block of Data Functions (READBLOCK, WRITEBLOCK)

Introduction

The *readblock* function reads a block of data (segment or record) from the input file and places it into the argument of a string variable. The *writeblock* function writes the data contains in the argument of a string variable to the output file.

The *readblock* and *writeblock* functions are used in conjunction with each other to pass a block of data from the input file to the output file without compliance checking or testing for proper EDI syntax. Together these functions provide a more efficient alternative of using “wildcard” segments, which are typically implemented in build and break maps.

Syntax

```
readblock(string_variable);  
writeblock(string_variable);
```

Note

See the “XML Build and Break Maps” appendix in the *XML User's Guide* for special considerations when using these functions with XML data.

Select Function (SELECT)

Introduction

The *select* function allows information to be retrieved from the database tables. Only the tables and fields available in the Select standard rule are available for the select extended rule. In the command syntax “expression” and “receiverlist” can be a string field, string variable, or string literal. It is important to note that the table and field names for the select extended rule are slightly different than those depicted in the standard rule. A listing of these table and field names can be found at the end of this chapter.

Syntax

```
select fieldname into receiverlist from tablename where key =  
expression [and key = expression];
```

Update Function (UPDATE)

Introduction

The *update* function allows information in the database tables to be updated. This function is similar to the Update standard rule, except that it provides more flexibility. Only the tables and fields available in the Update standard rule are available for the update extended rule. In the command syntax “expression” can be a string field, string variable, or string literal. It is important to note that the table and field names for the update extended rule are slightly different than those depicted in the standard rule. A listing of these table and field names can be found at the end of this chapter.

Syntax

```
update tablename set fieldname =expression [fieldname =expression]  
where key = expression [and key = expression];
```

Insert Function (INSERT)

Introduction

The *insert* function allows information in the database tables to be updated.

Syntax

```
insert into tablename [ (fieldlist) ] [ (valuelist) ];
```

User Exit Function (EXEC)

What is a user exit?

A user exit is an extended rule that enables the map to temporarily exit translation to enhance your functionality or fulfill specific requirements that Gentran:Server does not perform during normal translation.

Reference

See the User Exits, Appendix B, for more information on user exits.

Placing the extended rule

You can use the exec function in an extended rule for a map component at any hierarchical level, including field level extended rules, if appropriate.

Note

When you apply a user exit to a map component, subordinate map components may also be able to execute the same user exit.

Allowable data types

These are the extended rule data types that you can use with exec:

- INTEGER
- REAL
- STRING

Note

You cannot use date and time data unless you process it as a string data type.

ActiveX and User Exit Functions

Introduction

Gentran:Server now supports additional extended rule functionality to allow you to use Microsoft's ActiveX Data Objects (ADO) from within extended rules, as well as providing enhancements to user exit support.

ActiveX indexed properties

Gentran:Server now supports ActiveX indexed properties, specifically these types of index:

- variant
- numeric
- string

The index support allows you to use these syntaxes (if they are supported by your Automation server):

```
n = ob.property[1];
n = ob.property["Count"];
n = ob.property[ob.method()];
```

Gentran:Server also supports chaining of ActiveX method calls and properties, which simplifies extended rules for Automation servers with moderately complex object models (e.g., ADO).

Example

The following is an example of chaining statements:

```
recordset.fields.item["MessageId"].value
```

Createobject syntax

The *createobject* function enables you to create an instance of an ActiveX Automation Server.

Syntax

```
object = createobject("ProgID");
```

ProgID is the programmatic identifier. An example of a ProgID is:

"InternetExplorer.Application"

**Deleteobject
syntax**

The *deleteobject* function enables you to delete an instance of an ActiveX Automation Server. An object must be deleted before the end of the map that uses it. It is more efficient to delete the object immediately on completion (using the DELETEOBJECT command), although the Gentran:Server translator will delete the object automatically at the end of the map. Also, if you assign one object to another one, both copies of the object must be deleted for that object to be properly unloaded.

Syntax

```
deleteobject (object);
```

Getiid syntax

The *getiid* function enables you to obtain the unique identifier for an interface, by using the string-character name of the interface to return the globally unique identifier that is used by software to run the interface.

Syntax

```
string_variable = getiid("ProgID", "InterfaceID");
```

InterfaceID (IID) is the interface identifier. An example of an IID is:
"IWebBrowser2"

**Queryobject
syntax**

The *queryobject* function is used to request a different interface on an existing object.

Syntax

```
object2 = queryobject(object1, "{IID}");
```

Note

You may not use extended rules that compare two ActiveX properties or method results, or any combination of properties or method results. This type of comparison is invalid because property and method types are unknown prior to compilation and thus it is not possible to generate the correct comparison code.

Alphabetic Language Reference

Overview

Introduction

This section contains an alphabetic reference for the Gentran:Server proprietary programming language keywords and commands. This enables you to quickly and easily refer to information on specific keywords or commands.

Typeface conventions

This table lists the typeface conventions used in this section.

IF the typeface follows this example...	THEN it is used to indicate...
While	what you need to type.
condition	in syntax, information that you need to provide.
[1]	that, in syntax, items inside square brackets are optional.

Typeface conventions

The following programming guidelines are utilized in this section:

- Keywords and commands are shown in all lower case letters.
- Map components are shown in all upper case letters.
- A double slash (//) introduces comments.
- Lines that are too long to fit on one line in this section may be continue on the next line using a line-continuation character (Â).

atoi

Introduction The **atoi** function is a numerical function that converts strings into integers. The numerical functions enable you to convert one data type to another.

Syntax `int = atoi(string);`

where: `int` = integer variable
 `string` = string variable

Example

```
integer a;  
string[20] s;  
s = "5";  
a = atoi(s);  
// "a" contains the value 5
```

aton

Introduction The **aton** function is a numerical function that converts strings into real numbers. The numerical functions enable you to convert one data type to another.

Syntax `real = aton(string);`

where: `real` = real number variable
`string` = string variable

Example

```
real a;  
string[20] s;  
s = "5.5";  
a = aton(s);  
// "a" contains the value 5.5
```

auditlog

Introduction	<p>The auditlog function enables you to write user-defined audit messages to the Gentran:Server Audit Log. When an extended rule that calls an auditlog operation is executed, it writes the specified user-defined message to the Audit Log.</p> <p>Note This function does not return a value.</p>
Syntax	<pre>auditlog(MessageID, Type, Key, [, string1] [, string2] [, string3] [, string4] [, string5] [, string6] [, string7]);</pre>
MessageID	<p>The first parameter, MessageID, is the user-defined audit message identifier, which must be an integer value.</p>
Type	<p>The second parameter, Type, is a keyword that identifies the type of audit message. These pre-defined keywords are valid for Gentran:Server:</p> <ul style="list-style-type: none">• AL_PROC (processing)• AL_MSG (message) <p>Note You may also supply an integer value to account for a type for which a keyword is not currently defined.</p>
Key	<p>The third parameter, Key, is a keyword that is either zero (if the type of parameter two is “AL_PROC”) or the identification of a piece of data of the specified type. These pre-defined keywords are valid for Gentran:Server:</p> <ul style="list-style-type: none">• AL_KEY_INPUT• AL_KEY_OUTPUT <p>Note You may also supply an integer value to account for a key for which a keyword is not currently defined.</p>

Parameters 4 through 10

Parameters four through ten are optional string values that the user-defined message may require to fill in variables defined in the audit message.

Example1

```
auditlog(MessageID, AL_PROC, 0, ...);  
//Issues a processing message in any map.
```

Example2

```
auditlog(MessageID, AL_MSG, 0atoi(param(1)), ...);  
//Issues a data audit for the message which is currently  
//processing.
```

begin

Introduction

Gentran:Server uses conditional logic to test conditions and then, depending on the results of the test, perform different operations. Conditions can be nested to any level. You can use the **if/then** keywords to execute one or more statements conditionally. If you include more than one statement in the body of an **if/then** loop, you must surround the statements with the **begin/end** keywords. If you only use a single statement, you can omit the **begin** and **end**.

Note

Do not end conditions with a semicolon (;) – this terminating syntax is necessary for statements *only*.

Syntax

```
if condition then
begin
    statement1;
    statement2;
end
```

break

Introduction

The **break** function terminates the execution of the nearest enclosing **while** loop, and passes control to the statement that follows the **end** keyword. The **break** keyword is generally used in complex loops to terminate a loop before several statements have been executed.

Example

```
integer i
  i = 0;

While i<10 do
Begin
  #Total = Total + 50;
  i = i + 1;

  If #Total > 100000
    Break;
  Else
    Continue;
End
//While the value contained in the variable "i" is less than ten,
//50 will be added to the field Total.
//If the value in the field Total becomes greater than 100000
//before "i" equals 10, break out of the while loop else continue
//processing until "i" equals 10.
```

cerror

Introduction

The *cerror* function raises a compliance error and reports the target statement (the statement you specify) on the translation report. You typically specify this function as an action to be performed if a condition is false. This table describes when the *cerror* function is supported in Gentran:Server.

IF the map/form is of type...	THEN the cerror function is...
Screen entry	Not valid.
Print	Not valid.
Export	Only valid on the input side of the map.
Import	Valid on the input or output side of the map.
Break (Interchange, Group, or Transaction Set)	Only valid on the input side of the map.
Build (Interchange, Group, or Transaction Set)	Not valid.

Syntax

```
cerror(error_number,$GROUP_NAME[index][index][index].
&#FIELD_NAME);
```

Example

```
cerror(100,$ILD[0][1][1].#VATC);
//This raises compliance error 100 on the VATC element of the
//specified instance of the ILD group
```

Compliance error codes

The compliance error codes are as follows

Error Number	Error Generated
100	Mandatory Element
105	Conditional Element
110	Incorrect Element Format
111	Invalid Date
112	Invalid Time

(Continued on next page)

(Contd) Error Number	Error Generated
113	Invalid String
114	Invalid Number
115	Incorrect Element Length
116	Element Too Short
117	Element Too Long
118	Invalid Overpunch Data
119	Invalid Numeric XData
120	Too Many Components
121	Too Many Elements
125	Element Delimiter
130	Conditional Relation
135	Element Code
140	Standard Rule Failure
145	Binary Substitution Failed
200	Mandatory Component
205	Conditional Component
210	Incorrect Component Format
215	Incorrect Component Length
220	Component Delimiter
225	Component Code
300	Mandatory Segment
305	Segment Sequence
310	Invalid Start End
315	Incorrect Block Format
	(Continued on next page)
320	Segment Terminator

(Contd) Error Number	Error Generated
330	Unexpected Segment
400	Session Error
405	Unknown Partner
410	Invalid Control Number
415	Invalid Sub Count
420	Unknown Relationship
425	Unknown Standard
430	Unknown Version
435	Unknown Transaction
440	Duplicate Control Number
445	Duplicate Doc Name
500	ActiveX Bad Param Count
505	ActiveX Bad Argument Type
510	ActiveX Unknown Name
515	ActiveX Type Mismatch
520	ActiveX Missing Required Param
525	ActiveX Server Unavailable
530	ActiveX Object Not Created
535	ActiveX Other Error
600	XML Element Type Match
601	XML Element Root Type
602	XML Element Required
620	XML Attribute Undeclared
621	XML Attribute Not Unique
	(Continued on next page)
622	XML Attribute Required

(Contd) Error Number	Error Generated
623	XML Attribute Fixed Error
624	XML Attribute ID Not Unique
625	XML Attribute ID REF Match
630	XML Attribute Parse Name Error
631	XML Attribute Parse Equal Error
632	XML Attribute Parse OQuote Error
633	XML Attribute Parse CQuote Error
640	XML Attribute Name Error
641	XML Attribute Names Error
642	XML Attribute NMTOKEN Error
643	XML Attribute NMTOKENS Error
644	XML Attribute ENTITY Error
645	XML Attribute ENTITIES Error
660	XML Reference Char Invalid
661	XML Reference ENTITY Recursion
662	XML Reference ENTITY Undeclared
699	XML Unknown Error
700	ODBC Data Source Open Error
701	ODBC Data Source Rollback
702	ODBC Data Source Commit Error
703	ODBC Data Source Rollback Error
710	ODBC Query Open Error
711	ODBC Command Error
712	ODBC Cursor Error
720	ODBC Output Table Error

(Continued on next page)

(Contd) Error Number	Error Generated
721	ODBC Output Operation Error
730	ODBC Mandatory Element
731	ODBC Data Conversion Error
732	ODBC Incorrect Element Format
733	ODBC Standard Rule Failure
799	ODBC Unknown Error
800	NCPDP BFHS Missing
801	NCPDP BFHS Seg ID Error
802	NCPDP BFHS Transmission Type Error
803	NCPDP BFHS Sender ID Missing
804	NCPDP BFHS Batch ID Error
805	NCPDP BFHS File Type Error
806	NCPDP BFHS Receiver ID Missing
807	NCPDP BFDDR Missing
808	NCPDP BFDDR Seg ID Error
809	NCPDP BFDDR Transaction Ref Missing
810	NCPDP BFTR Missing
811	NCPDP BFTR Seg ID Error
812	NCPDP BFTR Batch ID Error
813	NCPDP BFTR Record Count Error
814	NCPDP BFTR Record Count Mismatch Error
815	NCPDP DR Repeating Segment Error
816	NCPDP DR Transaction Code Error
817	NCPDP DR Transaction Count Error
	(Continued on next page)
818	NCPDP DR Transaction Count Mismatch Error

(Contd) Error Number	Error Generated
819	NCPDP DR Header Response Code Error
820	NCPDP DR Transaction Response Code Error
830	NCPDP Null Char
832	NCPDP Invalid Char
833	NCPDP Unexpected EOF
835	NCPDP Unexpected STX
836	NCPDP Unexpected ETX
837	NCPDP Missing Stream Separator 1
838	NCPDP Missing Stream Separator 2
839	NCPDP Unexpected Separator
840	NCPDP Unexpected FS
841	NCPDP Unexpected GS
842	NCPDP Missing GS
845	NCPDP Positional Segment Tag Off
855	NCPDP Delimited Segment Empty
856	NCPDP Delimited Segment Missing First FS
857	NCPDP Delimited Field ID Invalid
858	NCPDP Delimited Field ID Invalid Char
859	NCPDP Delimited Field ID Null Char
860	NCPDP Delimited Field ID Incomplete
861	NCPDP Delimited Field Data Missing
862	NCPDP Delimited Field Data Null Char
863	NCPDP Delimited Field Data Invalid
864	NCPDP Delimited Field Data Invalid Char
	(Continued on next page)
865	NCPDP Delimited Segment ID Invalid

(Contd) Error Number	Error Generated
866	NCPDP Delimited Segment ID Invalid Char
867	NCPDP Delimited Segment ID Null Char
868	NCPDP Delimited Segment ID Incomplete
869	NCPDP Delimited Field ID Size
885	NCPDP Batch Nothing After STX
886	NCPDP Batch Unexpected STX
887	NCPDP Batch Missing ETX
888	NCPDP Batch Data File Open
890	NCPDP Data Field String Separator
891	NCPDP Data Field String No Filename
892	NCPDP Data Field String Start Byte
893	NCPDP Data Field String End Byte
894	NCPDP Data Field String Incomplete
895	NCPDP Data Field String Extra Characters
896	NCPDP Data Field Size Incorrect
897	NCPDP Data Field Start Out of Bounds
898	NCPDP Data Field Start Beyond End
899	NCPDP Unknown Error

continue

Introduction

The **continue** function continues the execution of the innermost loop without processing the statements in the loop that follow the **continue** statement.

Example

```
integer i
  i = 0;

While i<10 do
Begin
  #Total = Total + 50;
  i = i + 1;

  If #Total > 100000
    Break;
  Else
    Continue;
End

//While the value contained in the variable "i" is less than ten,
//50 will be added to the field Total.
//If the value in the field Total becomes greater than 100000
//before "i" equals 10, break out of the while loop else continue
//processing until "i" equals 10.
```

count

Introduction

The **count** function counts and returns the number of iterations of a group. An example is as follows:

```
integer i;  
i = count($N1[*]);  
//The [*] is a wildcard that counts the number of iterations of the //  
N1 group.
```

createobject

Introduction The **createobject** function enables you to create an instance of an ActiveX Automation Server.

Syntax `object = createobject("ProgID");`
where: ProgID = programmatic identifier

Example

```
object ob;  
ob = createobject("InternetExplorer.Application");  
//Creates an instance of the default interface of an ActiveX  
//Automation Server.  
//Note:  
//The createobject command is more efficient if you use the IID  
//instead of the interface name.
```

date

Introduction The **date** function converts a string type into a datetime type using a format that you specify. This function allows you to include static characters such as a slash (/), which gives you access to full date support.

Syntax Datetime = date("format",string);

where: datetime = datetime variable (month specified as 1-12)
 format = desired date format
 string = string variable

Example

```
datetime d;
d = date(1995,4,6);
d = date(1995,4,6,12,0);
d = date("%y/%m/%d","95/4/6");
d = date("%y/%m/%d",#strdate);
```

Format specifiers This table lists the format specifiers.

Format Specifier	Description
%8	ISO-8601 date format Format YYYYMMDDTHHMMSS.mmmZ Four-digit year, two-digit month, two-digit day, T (time) indicator, two-digit hour, two-digit minutes, two-digit seconds in Universal Time (also called Zulu Time or Greenwich Mean Time), Z (Zulu time) indicator (example: 20031209T123000.000Z) Note This date format cannot be combined with any other format specifier.
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%d	Day of the month as a decimal number (01 – 31)

(Continued on next page)

(Contd) Format Specifier	Description
%D	ISO-8601 date format (date component only) Format YYYYMMDDZ Note This date format cannot be combined with any other format specifier.
%H	Hour in 24-hour format (00 – 23)
%I	Hour in 12-hour format (01– 12)
%j	Day of the year as a decimal number (001 – 366)
%m	Month as a decimal number (01 – 12)
%M	Minute as a decimal number (00 – 59)
%S	Second as a decimal number (00 – 59)
%U	Week of the year as a decimal number, with Sunday as the first day of the week (00 – 51)
%w	Weekday as a decimal number (0 – 6, with Sunday as “0”)
%W	Week of the year as a decimal number, with Monday as the first day of the week (00 – 51)
%y	Year without the century as a decimal number (00 – 99)
%Y	Year with the century as a decimal number
%%	Percent sign

delete

Introduction The **delete** function deletes a specified occurrence of a group.

Syntax `delete($GROUP_NAME[N]);`
where [n] is the occurrence of the group that you want to delete.

Example `delete($ILD[2]);`
`//Deletes the second occurrence of the ILD group.`

deleteobject

Introduction

The **deleteobject** function enables you to delete an instance of an ActiveX Automation Server. An object must be deleted before the end of the map that uses it. It is more efficient to delete the object immediately on completion, although the Gentran:Server translator will delete the object automatically at the end of the map. Also, if you assign one object to another one, both copies of the object must be deleted for that object to be properly unloaded.

Syntax

```
deleteobject(object);
```

Example

```
object ob;  
ob = createobject("InternetExplorer.Application");  
deleteobject(ob);  
//Deletes the instance of the object.
```

empty

Introduction The **empty** function sets the value of a field in internal storage to null. This function is not the same as setting the value of a field to a zero length string (" ") or to zero.

Syntax empty(\$GROUP_NAME[index][index][index]. #FIELD_NAME)

Example

```
empty($ILD.#VATC);  
//Set the value of the specified instance of the VATC element  
//to null
```

end

Introduction

Gentran:Server uses conditional logic to test conditions and then, depending on the results of the test, perform different operations. Conditions can be nested to any level. You can use the **if/then** keywords to execute one or more statements conditionally. If you include more than one statement in the body of an **if/then** loop, you must surround the statements with the **begin/end** keywords. If you only use a single statement, you can omit the **begin** and **end**.

Note

Do not end conditions with a semicolon (;) – this terminating syntax is necessary for statements *only*.

Syntax

```
if condition then
begin
    statement1;
    statement2;
end
```

exec

Introduction

The **exec** function invokes the execution of a batch file or program. The translator waits until the script finishes before continuing with translation. After the script runs and returns a numeric return code, the **exec** function:

- Retrieves the return code
- Returns to translation
- Uses the return code in translation.

Note

You must set an integer value equal to the return value of the **exec (...)** call for the rule to compile.

Syntax

```
nReturn = exec(string)
```

where:

nReturn = return value

string = string variable or literal value that represents the shell script

Example

```
integer nReturn;  
nReturn = 0;  
nReturn = exec ("c:\addrunm.sh");
```

exist

Introduction

The **exist** function tests to determine if a field is empty (null). It returns a non-zero (true) value if there is data in a specified field in internal storage. If data is not present in the specified field, this function returns a zero (false) value. This function is typically used as a part of a condition.

There are some situations (for example, if the field or element has a "Use Code" standard rule applied to it) when the IF EXIST returns a non-zero (true) value whether or not the condition is true. You can work around this by only using IF EXIST for date- and number-type fields. Be certain that all references to the field which is interrogated are nested within the IF EXIST begin block.

For string-type fields, use the following format:

```
IF Field1 = ""
```

Syntax

```
if exist($GROUP_NAME[index][index][index]. #FIELD_NAME) then
```

Example

```
if exist($ILD.#VATC) then
    #TEMP_FIELD = #VATC;

//Return a non-zero value if the condition is true (data is present
//in the
//specified instance of the VATC element). A zero value is returned
//if the condition is false (no data is present in the specified
//instance of
//the VATC element).
```

fseek

Introduction

The **fseek** function moves the file pointer to a new location which is a specified number of bytes (offset) from the designated point of origin in the file (the point of origin may be the beginning of the file or relative to either the end of the file or the current position).

To invoke the fseek function against the input file, the value for `current_file` is "0." To invoke the fseek function against the output file, the value for `current_file` is "1." The fseek function is typically only used in conjunction with the ftell, readblock, and writeblock functions.

Syntax

```
fseek(current_file,offset,origin);
```

where:

<code>current_file</code>	=	0 indicates the input file; 1 indicates the output file
<code>offset</code>	=	The position to which the file pointer is moved relative to the origin.
<code>origin</code>	=	The keyword depends on the starting location: begin = start at beginning of the file end = start at the end of the file current = start at the current position in the file

Example

```
string[1024]temp_buffer;
Integer Position;
Position = ftell(0);
while readblock(temp_buffer) do
begin
    if left(temp_buffer,3) = "IEA" then
        begin
            fseek(0,Position,begin);
            break;
        end
    writeblock(temp_buffer);
    Position = ftell(0);
end
//Read a segment from input file and place in temp_buffer. Look for
//"IEA" segment tag. If found, reset file pointer to where it was
//before the "IEA" segment was read. Write contents of temp_buffer
//to output file. Set "Position" = the current file pointer
//position.
```

ftell

Introduction

The **ftell** function obtains the current position of the file pointer and returns it as an integer. To invoke the **ftell** function against the input file, the value for `current_file` is "0." To invoke the **ftell** function against the output file, the value for `current_file` is "1." The **fseek** function is typically only used in conjunction with the **fseek**, **readblock**, and **writeblock** functions.

Syntax

```
numeric_variable = ftell(current_file);
```

Example

```
string[1024]temp_buffer;
Integer Position;
Position = ftell(0);
while readblock(temp_buffer) do
begin
    if left(temp_buffer,3) = "IEA" then
        begin
            fseek(0,Position,begin);
            break;
        end
    end
    writeblock(temp_buffer);
    Position = ftell(0);
end
//Read a segment from input file and place in temp_buffer. Look for
//"IEA" segment tag. If found, reset file pointer to where it was
//before the "IEA" segment was read. Write contents of temp_buffer
//to output file. Set "Position" = the current file pointer
//position.
```

get

Introduction The `get` function enables you to access individual components of a datetime variable.

Syntax `integer_variable = get datetime_component (datetime_variable);`

where:

<code>integer_variable</code>	=	integer variable
<code>datetime_component</code>	=	individual component of the datetime variable
<code>datetime_variable</code>	=	datetime variable of which you want to access a component part

Example

```
integer temp_days;
integer temp_hours;
datetime d;
temp_days = 0;
temp_hours = 0;
d = '12/25/2001 12:15:30'; //A fields value in the map can be
// assigned to the datetime variable "d"
//or a hard coded value can be assigned.

temp_days = get days (d);
temp_hours = get hours (d);

//Accesses the days from the datetime variable "d" and loads into
//variable " temp_days ". Accesses the hours from the datetime
//variable "d" and loads into variable "temp_hours".
```

getiid

Introduction The **getiid** function enables you to obtain the unique identifier for an interface, by using the string-character name of the interface to return the globally unique identifier that is used by software to run the interface.

Syntax This creates an instance of the default interface of an ActiveX Automation Server.

```
string_variable = getiid("ProgID");
```

This looks up the InterfaceID (IID) of an interface.

```
string_variable = getiid("ProgID", "Interface_Name or {Interface ID}");
```

Example

```
object ob;
string[50] iid;
iid = getiid("InternetExplorer.Application", "IWebBrowser2");
ob = createobject("InternetExplorer.Application", iid);
ob.Visible = 1;
//Displays the Internet Explorer on the desktop by setting a
//property value in an ActiveX Automation Server.
```

if then...else

Introduction

The **if**, **then**, and **else** keywords allow the use of conditional logic. Gentran:Server uses conditional logic to test conditions and then, depending on the results of the test, perform different operations. Conditions can be nested to any level. You can use the **if/then** keywords to execute one or more statements conditionally. The condition is typically a comparison, but it can be any expression that concludes with a numeric value. Gentran:Server interprets the value as either *true* or *false*. The system interprets a zero value as false and a nonzero value as true.

Gentran:Server evaluates the **if/then** condition, and if it is true, the system executes all the statements that follow the **then** keyword. If the condition is false, none of the statements following **then** are executed.

You can use the **else** keyword in conjunction with **if/then** to define several blocks of statements, one of which is executed. Gentran:Server tests the first **if/then** condition. If the condition is false, the system proceeds to test each sequential condition until it finds one that is true. The system executes the corresponding block of statements for the true condition. If none of the **if/then** conditions are true, the system executes the statements following the **else** keyword.

Note

Do not end conditions with a semicolon (;) – this terminating syntax is necessary for statements *only*.

Example

```
if condition then
begin
    statement1;
    statement2;
end
else
begin
    statement3;
    statement4;
end
```

index




Introduction The **index** function enables you to determine which instance of a particular loop the translator is currently accessing.

Syntax `index(integer_variable);`

where: `integer_variable` = integer variable that indicates the hierarchical level for which you want to determine the loop count

Example

```
index(1)
//This extended rule is located on the ILD group in the diagram
//below. This determines the current loop count for the first level
//(ODD) in the hierarchical structure.
```

```
    INVOIC -M- INVOICE DETAILS 
      ODD -M- 999999 
        ILD -M- 999999 
```

insert

Introduction

The *insert* function allows information in the database tables to be updated.

Syntax

```
insert into tablename [ (fieldlist) ] [ (valuelist) ];
```

where: tablename DivisionLookup | PartnerLookup | DivisionLocation
 PartnerLocation | DivisionXref | PartnerXref

 fieldlist (fieldname [, fieldname])

 fieldname The name of one of the fields in the table.

 valuelist (String [, String])

Notes

- A list of the tables is available later in this chapter.
 - The fieldlist lists one or more fieldnames to which data is to be added.
 - The fieldnames can be listed in any order.
 - The valuelist must be in the same order as the field list.
-

Example

```
updateStatus = update PartnerLookup
set Description = "Lookup Update
Test",Text1="Text1Updated",Text2="Text2Updated",Text3="Text3Updated",T
ext4="Text4Updated"
where TableName = "PartLkp" and Item = "1";

if updateStatus = 1 then
begin
  messagebox("Update PartnerLookup: Record Not Found,Attempting
  ^Insert...",1);
  insertStatus = insert into PartnerLookup( PartnerKEY, TableName,
  ^Item,Description,Text1,Text2,Text3,Text4)
  ^values ("PETZONE","PartLkp","1","Lookup Insert
  ^Test","Text1","Text2","Text3","Text4");
  if insertStatus = 0 then
  begin
    messagebox("Insert PartnerLookup: Failed",1);
  end
end

if updateStatus = 2 then
begin
  messagebox("Update PartnerLookup Failed with OtherError",1);
end
//Example assumes that a lookup table named "PartLkp" has been
//created.
```

left

Introduction The **left** function extracts a specified number of character from the left side of a string variable or field and returns the result as a string.

The syntax is as follows:

```
string_variable = left(string_variable,num_char)
```

where: num_char = integer variable

Note

You may not use an ActiveX property as the first parameter of the left function because the length of the property is unknown prior to compilation.

Example

```
string [25]name;  
string [5]temp_variable;  
name = "Acme Shipping Company"  
temp_variable = left(name,4);  
// "temp_variable" would contain "Acme"
```

len

Introduction The **len** function is a numerical function that counts and returns the number of characters in a string. The numerical functions enable you to convert one data type to another.

Syntax `number_char = len(string);`

where: `num_char = integer variable`

Example

```
integer a;  
a = 0;  
a = len("hello");  
// "a" contains the value 5
```

messagebox

Introduction

The **messagebox** function enables you to display a message box for which you have designated the format and content. You can specify the number and type of buttons on the message box, the message icon (e.g., hand, question mark, exclamation point, or asterisk), and the message displayed. You can also issue a return value based on the chosen action.

Note

Only String values can be displayed in a message box.

Syntax

```
messagebox("message",defined_number)
```

where: message = message string
 defined_number = defined number of the desired buttons plus the
 defined number of the desired icon (if used)

Example

```
if messagebox("Do you really want to delete this object?",36) = 6
begin
.
.
.
end

//Displays a message box with the given string as the message, Yes
//and No buttons (4) and a question mark icon (32). The number and
//type of buttons (4) plus the icon (32) equals the defined_number
//(36). If the user clicks the Yes button (return value "6"), the
//statements in the begin/end loop are executed.
```

Defined_ numbers

This table lists the defined_numbers for the button and icon types.

Defined_Number	Button or Icon Types
0	OK button only
1	OK and Cancel buttons
4	Yes and No buttons
16	Icon Hand
32	Icon Question Mark
48	Icon Exclamation Point
64	Icon Asterisk

**Message box
return values**

This table lists the message box return values.

Return Value	Action Selected
1	OK selected
2	Cancel selected
6	Yes selected
7	No selected

mid

Introduction The **mid** function extracts from a specified position in a string, either to the end of the string or for a specified number of characters and returns the resultant string. This function is zero-based.

Syntax `string_variable = mid(string_variable,start_pos,num_char)`

where: `start_pos` = integer variable
 `num_char` = integer variable

Note

You may not use an ActiveX property as the first parameter of the mid function because the length of the property is unknown prior to compilation.

Example

```
string [25]name;  
string [10]temp_variable;  
name = "Acme Shipping Company"  
temp_variable = mid(name,5,8);  
// "temp_variable" will contain "Shipping"
```

ntoa

Introduction

The **ntoa** function is a numerical function that converts integers and real numbers into strings. The numerical functions enable you to convert one data type to another.

Syntax

```
string = ntoa(integer/real,string);
```

where: integer/real = integer or real number variable
 string = string variable

Example

```
Integer b;  
string[8] s;  
b = 5.5;  
ntoa(b,s);  
//The variable "s" contains the string "5.5".
```

param

Introduction

The **param** function is used to read the value of the PARAM(n) variable. The extended rule allows you to reference values that have been passed into the translator via the command line. The -u switch can be used during the invocation of the translator to pass values in so they can be referenced by an extended rule. Typically, the param extended rule is not used by maps that are running within the Gentran:Server for Windows environment because Gentran:Server does not use the -u switch when invoking the translator. However, if you are invoking the translator to perform translation outside of Gentran:Server (tx32 -f <inputfile> <templatefile> <outputfile> <reportfile>), you have the option to pass values into the translator via the -u switch and reference those values using the param extended rule.

Syntax

param(integer_PARAM_number);
where: interger_PARAM_number = the number of the pre-defined variable

Example

For example, if you invoke TX32.EXE in the following manner:

```
tx32.exe -f input.txt 850.tpl output.txt out.rpt -u  
"PurchaseOrderNumber" -u 12345 -u 500.25
```

Then you could write an extended rule in the \\GENSRVNT\Tutorial\Pet_850.map that looks like the following:

```
string [32] strDescription;  
string [32] strPONbr;  
string [32] strTotalCost;  
real nTotalCost;  
  
strDescription = param(0);  
strPONbr = param(1);  
strTotalCost = param(2);  
nTotalCost = aton(strTotalCost);  
  
if nTotalCost > 200 then  
begin  
    messagebox("Get approval from boss",0);  
end
```

You can run this rule from any scope in your map (e.g., pre-session, post-session, group onbegin, field, etc.).

queryobject

Introduction The **queryobject** function is used to request a different interface on an existing object.

Syntax `object2 = queryobject(object1, "{IID}");`

where:

- `object2` = is the result that contains the requested interface to the object
- `IID` = is the interface identifier of the requested interface
- `object1` = an existing object

Example

```
object ob, ob2;
ob = createobject("InternetExplorer.Application");
ob2 = queryobject(ob, "{EAB22AC1-30C1-11CF-A7EB-Å0000C05BAE0B}");
//Uses the Interface ID of the desired interface to obtain another
//(different) interface of the existing object (object1).
```

readblock

Introduction

The **readblock** function reads a block of data (segment or record) from the input file and places it into the argument of a string variable. The **readblock** and **writeblock** functions are used in conjunction with each other to pass a block of data from the input file to the output file without compliance checking or testing for proper EDI syntax. Together these functions provide a more efficient alternative of using “wildcard” segments, which are typically implemented in build and break maps.

Note

See the “XML Build and Break Maps” appendix in the *XML User's Guide* for special considerations when using this function with XML data.

Syntax

```
readblock(string_variable);
```

Note

You may not use an ActiveX property as the first parameter of the **readblock** function because the length of the property is unknown prior to compilation.

Example

```
while readblock(temp_buffer) do
begin
    if left(temp_buffer,3) = "IEA" then
        begin
            fseek(0,Position,begin);
            break;
        end
    writeblock(temp_buffer);
    Position = ftell(0);
end
//Read segment from input file and place in temp_buffer. Look for
//"IEA" segment tag. If found, reset file pointer to where it was
//before the "IEA" segment was read. Write contents of temp_buffer
//to output file. Set "Position" = the current file pointer
//position.
```

readbytes

Introduction

The **readbytes** function reads a number of bytes from the input file. This function is used in conjunction with the writebytes function. Used together, the readbytes and writebytes function provide an efficient method of passing data through a map if the data does not need to be compliance checked or altered in any way.

Readbytes is similar to the readblock function, but readblock only works with entire blocks (i.e., an entire segment or record), and readbytes works with any quantity of data, whether it is smaller or larger than a block.

The readbytes function uses two parameters, first the string variable into which the data being read will be stored, and second the number of bytes to read.

Syntax

```
readbytes(read_from_buffer, num_bytes);
```

where: read_from_buffer= string variable into which the data being read will be stored
num_bytes = integer value representing the number of bytes to read from the input file.

Note

Readbytes returns the number of bytes it was actually able to read.

Example

```
string [1024] tempBuffer;  
while readbytes(tempBuffer,1024) do  
begin  
    writebytes(tempBuffer,1024);  
End  
//Read 1024 bytes from input file and place in string variable  
//named tempBuffer.  
  
writebytes("^0D^0A",2);  
//Appends a CRLF to the end of the output file.
```

right

Introduction The **right** function extracts a specified number of characters from the right side of a string variable or field.

Syntax `string_variable = right(string_variable,num_char)`

where: `num_char` = integer variable

Note

You may not use an ActiveX property as the first parameter of the right function because the length of the property is unknown prior to compilation.

Example

```
string [25]name;  
string [10]temp_variable;  
name = "Acme Shipping Company"  
temp_variable = right(name,7);  
// "temp_variable" would contain "Company"
```

select

Introduction

The **select** function allows information to be retrieved from the database tables. Only the tables and fields available in the Select standard rule are available for the select extended rule. In the command syntax “expression” and “receiverlist” can be a string field, string variable, or string literal. It is important to note that the table and field names for the select extended rule are slightly different than those depicted in the standard rule. A listing of these table and field names can be found at the end of this chapter.

Syntax

select *fieldname* into *receiverlist* from *tablename* where *key* = *expression* [and *key* = *expression*];

Example

```
string[10]temp_var1,temp_var2,temp_var3;
select Text1, Text2, Text3 into temp_var1, temp_var2, temp_var3 Åfrom
Partnerlookup where Tablename = "Test" and Item ="widgit"
//Example assumes that a Lookup table named "Test" has been
//created.
```

set

Introduction The **set** function enables you to define individual components of a datetime variable.

Syntax set datetime_component (datetime_variable, integer_variable);

where: datetime_component = individual component of the datetime variable
 datetime_variable = datetime variable of which you want to access a component part
 integer_variable = integer variable

Example

```
integer temp_days;
datetime d;
temp_days = 15;
d = '12/25/2001'; //A fields value in the map can be
// assigned to the datetime variable "d"
//or a hard coded value can be assigned.

set days (d,temp_days);

//Defines the days of the datetime variable "d" from variable
//"temp_days".
```

strdate

Introduction The **strdate** function converts a datetime type into a string using a format that you specify. This function allows you to include static characters such as a slash (/), which gives you access to full date support.

Syntax `strdate(datetime,"format",string);`

where: `datetime` = datetime variable (month specified as 1-12)
 `format` = desired date format
 `string` = string variable

Example

```
datetime d;
string[8] s;

d = #DateField;

strdate(d, "%y/%m/%d", s);

//Converts a datetime variable into an eight character string in
//the format "year/month/day".
```

Format specifiers

This table lists the format specifiers.

Format Specifier	Description
%8	ISO-8601 date format Format YYYYMMDDTHHMMSS.mmmZ Four-digit year, two-digit month, two-digit day, T (time) indicator, two-digit hour, two-digit minutes, two-digit seconds in Universal Time (also called Zulu Time or Greenwich Mean Time), Z (Zulu time) indicator (example: 20031209T123000.000Z) Note This date format cannot be combined with any other format specifier.
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name

(Continued on next page)

(Contd) Format Specifier	Description
%d	Day of the month as a decimal number (01 – 31)
%H	Hour in 24-hour format (00 – 23)
%I	Hour in 12-hour format (01– 12)
%j	Day of the year as a decimal number (001 – 366)
%m	Month as a decimal number (01 – 12)
%M	Minute as a decimal number (00 – 59)
%S	Second as a decimal number (00 – 59)
%U	Week of the year as a decimal number, with Sunday as the first day of the week (00 – 51)
%w	Weekday as a decimal number (0 – 6, with Sunday as “0”)
%W	Week of the year as a decimal number, with Monday as the first day of the week (00 – 51)
%y	Year without the century as a decimal number (00 – 99)
%Y	Year with the century as a decimal number
%%	Percent sign

strstr

Introduction

The **strstr** function finds a substring inside a string. This function returns the position of the first instance of the designated substring within the specified string. If this function does not find the specified substring in the string, it returns a value of -1. This function is zero-based.

Syntax

```
integer = strstr("string","substring");
```

where:

integer	=	integer variable
string	=	string
substring	=	part of the string

Example

```
integer d;  
d = 0;  
  
d = strstr("mississippi","is");  
  
//Finds the first instance of the substring "is" inside the string  
//"mississippi" and returns the position of that first substring.
```

unreadblock

Introduction

The **unreadblock** function provides a method of moving the input file-pointer back one block (a block of data is equivalent to one EDI segment or one positional record). This function unread the block of data that was just processed by the readblock function.

Note

Unreadblock should only be used in conjunction with the readblock function. The unreadblock function will only unread the most recent block of data processed.

Unreadblock is provided as an alternative to using the fseek and ftell functions, and is the preferred method of moving the file pointer back one block of data. Unreadblock allows the translator to correctly track the number of bytes read and number of segments read during the translation process by moving the file-pointer back and decrementing the segment and byte counts accordingly.

Syntax

```
unreadblock();
```

Example

```
string[1024] tempBuffer;

while readblock(tempBuffer) do
begin
  if left(tempBuffer,3) = "IEA" then
  begin
    unreadblock();
    break;
  end
  writeblock(tempBuffer);
end

//Read segment from input file and place in tempBuffer variable.
//Look for the "IEA" segment tag and, if found, reset the file-
//pointer to where it was located before the "IEA" segment was
//read. Write contents of tempBuffer to output file.
```

update

Introduction

The **update** function allows information in the database tables to be updated. This function is similar to the Update standard rule, except that it provides more flexibility. Only the tables and fields available in the Update standard rule are available for the update extended rule. In the command syntax “expression” can be a string field, string variable, or string literal. It is important to note that the table and field names for the update extended rule are slightly different than those depicted in the standard rule. A listing of these table and field names can be found at the end of this chapter.

This table describes how the update function is supported in Gentran:Server.

IF the map/form is of type...	THEN the update function is...
Export	Only valid on the input side of the map.
Import	Valid on the input or output side of the map.
Break (Interchange, Group, or Transaction Set)	Only valid on the input side of the map.
Build (Interchange, Group, or Transaction Set)	Only valid on the input side of the map.
Print	Not supported.
Screen entry	Not supported.

Syntax

`update tablename set fieldname =expression [fieldname =expression] where key = expression [and key = expression];`

Example

```
string[15]temp_variable;
temp_variable = "5 Pound Hammer";
update PartnerXref set Text1 = temp_variable where MyItem = " and
ATablename = "Tools";
//Example assumes that a cross-reference table named "Tools" has
//been created.
```

while...do

Introduction

The **while...do** function executes a statement repeatedly until the specified termination condition evaluates to zero. The system tests the terminating condition *before* each iteration of the loop, so a **while** loop executes zero or more times depending on the value of the termination expression.

Note

Do not end conditions with a semicolon (;) – this terminating syntax is necessary for statements *only*.

Example

```
integer i
  i = 0;

While i<10 do
Begin
  #Total = Total + 50;
  i = i + 1;

  If #Total > 100000
    Break;
  Else
    Continue;
End

//While "i" is less than 10 execute the loop.
```

winexec

Introduction

The **winexec** function enables you to execute another program while running the translator. This program is executed asynchronously. You specify the program and determine how you want the program window displayed. You can also return an error code, if desired. If the error code is greater than 32, the program ran without errors. If the error code is less than 32, the program did not run because of an error. If the error code is "0," the system is out of memory. If the error code is "2," you didn't specify a file name. The error code is *not* the return value from the program you executed.

Note

- If you specify a program on another machine or in another domain, you must have the appropriate permission to access the specified folder.
- If translation is executing from an unattended process control command, the user ID under which that service is running must have the appropriate permission to access the specified file.

Syntax

```
winexec("program",window_display)
```

where: program = executable program name string (if necessary, including UNC or direct file path)
 window_display = number that indicates how you want the program window displayed

Example

```
winexec("program.exe", 3)

//Exits Gentrans:Server and executes the "program.exe"
//program asynchronously. The system displays the program window
//maximized (3).
```

Window display numbers

The window_display numbers that control the appearance of the program window are as follows (you must use the *number* to indicate how you want the program window displayed, not the window_display value):

Number	Window_Display	Definition
0	SW_HIDE	Hides the window and activates another window.
1	SW_SHOWNORMAL	Activates and displays a window. If the window is minimized or maximized, it is restored to the original size and position. This flag should be specified when displaying a window for the first time. (Continued on next page)

(Contd) Number	Window_Display	Definition
1	SW_NORMAL	Activates and displays a window in the original size and position.
2	SW_SHOWMINIMIZED	Activates the window and displays it as a minimized windows.
3	SW_SHOWMAXIMIZED	Activates the window and displays it as a maximized window.
3	SW_MAXIMIZE	Maximizes the window.
4	SW_SHOWNOACTIVATE	Displays the window in its most recent size and position, but does not activate it (the current active window remains active).
5	SW_SHOW	Activates the window and displays it in its current size and position.
6	SW_MINIMIZE	Minimizes the window.
7	SW_SHOWMINNOACTIVE	Displays the window as a minimized window without activating it (the current active window remains active).
8	SW_SHOWNA	Displays the window in its most recent size and position without activating it (the current active window remains active).
9	SW_RESTORE	Activates and displays the window. If the window was minimized or maximized, it is restored to its original size and position.
10	SW_SHOWDEFAULT	Activates the window and allows Windows to determine the size and position.

writeblock

Introduction

The **writeblock** function writes the data contains in the argument of a string variable to the output file. The readblock and writeblock functions are used in conjunction with each other to pass a block of data from the input file to the output file without compliance checking or testing for proper EDI syntax. Together these functions provide a more efficient alternative of using “wildcard” segments, which are typically implemented in build and break maps.

Note

See the “XML Build and Break Maps” appendix in the *XML User's Guide* for special considerations when using this function with XML data.

Syntax

```
writeblock(string_variable);
```

Example

```
while readblock(temp_buffer) do
begin
  if left(temp_buffer,3) = "IEA" then
  begin
    fseek(0,Position,begin);
    break;
  end
  writeblock(temp_buffer);
  Position = ftell(0);
end
//Read segment from input file and place in temp_buffer. Look for
//"IEA" segment tag. If found, reset file pointer to where it was
//before the "IEA" segment was read. Write contents of temp_buffer
//to output file. Set "Position" = the current file pointer
//position.
```

writebytes

Introduction

The **writebytes** function writes a specified number of bytes to the output file. This function is used in conjunction with the readbytes function. Used together, the readbytes and writebytes function provide an efficient method of passing data through a map if the data does not need to be compliance checked or altered in any way.

Note

The system does not append a segment terminator to the end of the string value written to the output file.

Writebytes is similar to the writeblock function, but writeblock only works with entire blocks (i.e., an entire segment or record), and writebytes works with any quantity of data, whether it is smaller or larger than a block.

The writebytes function uses two parameters, first the string variable containing the data to be written to the output file, and second the number of bytes to write.

Syntax

```
writebytes(write_to_buffer, num_bytes);
```

where: write_to_buffer = string variable containing the data to be written to the output file
num_bytes = integer value representing the number of bytes to read from the write_to_buffer variable and write to the output file

Note

If hex values are used in constants, use \0x instead of ^.

Example

```
string [1024] temp_buffer;  
while readbytes(temp_buffer,1024) do  
begin  
    writebytes(temp_buffer,1024);  
End  
writebytes("^0D^0A",2);  
  
//Read 1024 bytes from input file and place in string variable Snamed  
temp_buffer.  
//Appends a CRLF after the while loop finishes executing.
```

Select and Update Available Options

Introduction

The following section contains the database table names and the associated field names that are available when using the **select** and **update** extended rules. Additional keys are also available for certain tables. Where applicable, a description of the key follows the field name.

Note

When you use extended rules to reference a Gentran:Server database table, the syntax used is different from the actual table name. The appropriate referencing syntax is outlined for each table in this section.

Example

To refer to the Document_tb in an extended rule, you would reference Document.

Document

These are the field names that are available when using the select or update extended rules on the Document Table (Document_tb).

Note

This sub-table can only be accessed from the following map types; Transaction Break/Build, Import, Export, Print, Screen, Group Build, and Interchange Build.

The information that is accessed will be the information of the last document that was processed at the table level accessed by the rules. This must be taken into consideration when using rules that access these tables.

See *Table access examples* on page 3 - 8 for examples.

Note

Refer to this table as **Document**.

- ▶ TESTMODE
 - ▶ AGENCY
 - ▶ VERSION
 - ▶ TRANSACTIONSETID
 - ▶ RELEASE
 - ▶ DOCUMENTTYPE
 - ▶ REFERENCEDATA
 - ▶ DOCUMENTNAME
 - ▶ APPFIELD1
 - ▶ APPFIELD2
 - ▶ APPFIELD3
 - ▶ APPFIELD4
 - ▶ APPFIELD5
-

(Continued on next page)

-
- Document (contd)**
- ▶ APPFIELD6
 - ▶ DOCUMENTPARTNERKEY
 - ▶ CONTROLNUMBER
 - ▶ PARTNERKEY
 - ▶ TABLEKEY (Select only)

Partner These are the field names that are available when using the select or update extended rules for a non-division partner on the Partner Table (Partner_tb).

Note

Refer to this table as **Partner**.

- ▶ PARTNERNAME
- ▶ EDICODE
- ▶ APPLICATIONPARTNERKEY

These are the additional keys that are available for the Partner table.

- ▶ PARTNERKEY
- ▶ ALTERNATEKEY (refers to Application Key)
- ▶ APPLICATIONPARTNERKEY (refers to Application Key)

Division These are the field names that are available when using the select or update extended rules to reference the division partner on the Partner Table (Partner_tb).

Note

Refer to this table as **Division**.

- ▶ PARTNERNAME
 - ▶ EDICODE
 - ▶ APPLICATIONPARTNERKEY
-

PartnerLocation

These are the field names that are available when using the select or update extended rules for a non-division location on the Location Table (Location_tb).

Note

Refer to this table as **PartnerLocation**.

- ▶ CONTACTNAME
- ▶ NAME
- ▶ ADDRESS1
- ▶ ADDRESS2
- ▶ ADDRESS3
- ▶ CITY
- ▶ STATE
- ▶ ZIP
- ▶ COUNTRY
- ▶ TELEPHONE
- ▶ PRIMARYREFCODE
- ▶ SECONDARYREFCODE
- ▶ FAX

DivisionLocation

These are the field names that are available when using the select or update extended rules to reference a division location on the Location Table (Location_tb).

Note

Refer to this table as **DivisionLocation**.

- ▶ CONTACTNAME
 - ▶ NAME
 - ▶ ADDRESS1
 - ▶ ADDRESS2
 - ▶ ADDRESS3
 - ▶ CITY
 - ▶ STATE
 - ▶ ZIP
 - ▶ COUNTRY
 - ▶ TELEPHONE
 - ▶ PRIMARYREFCODE
 - ▶ SECONDARYREFCODE
 - ▶ FAX
-

PartnerLookup

These are the field names that are available when using the select or update extended rules for a non-division lookup on the Lookup Table (Lookup_tb).

Note

Refer to this table as **PartnerLookup**.

- ▶ ITEM
- ▶ DESCRIPTION
- ▶ TEXT1
- ▶ TEXT2
- ▶ TEXT3
- ▶ TEXT4

This is an additional key that is available for the PartnerLookup table.

- ▶ TABLENAME
-

DivisionLookup

These are the field names that are available when using the select or update extended rules to reference a division lookup on the Lookup Table (Lookup_tb).

Note

Refer to this table as **DivisionLookup**.

- ▶ ITEM
- ▶ DESCRIPTION
- ▶ TEXT1
- ▶ TEXT2
- ▶ TEXT3
- ▶ TEXT4

This is an additional key that is available for the DivisionLookup table.

- ▶ TABLENAME
-

PartnerXref These are the field names that are available when using the select or update extended rules for a non-division cross-reference on the Cross-reference Table (CrossReference_tb).

Note

Refer to this table as **PartnerXref**.

- ▶ MYITEM
- ▶ PARTNERITEM
- ▶ DESCRIPTION
- ▶ TEXT1
- ▶ TEXT2
- ▶ TEXT3
- ▶ TEXT4

This is an additional key that is available for the PartnerXref table.

- ▶ TABLENAME

DivisionXref These are the field names that are available when using the select or update extended rules to reference a division cross-reference on the Cross-reference Table (CrossReference_tb).

Note

Refer to this table as **DivisionXref**.

- ▶ MYITEM
- ▶ PARTNERITEM
- ▶ DESCRIPTION
- ▶ TEXT1
- ▶ TEXT2
- ▶ TEXT3
- ▶ TEXT4

This is an additional key that is available for the DivisionXref table.

- ▶ TABLENAME
-

Interchange

These are the field names that are available when using the select or update extended rules on the Interchange Table (Interchange_tb).

Note

This sub-table can only be accessed from the following map types: Interchange Break or Build, Group Break, Transaction Break, and Export.

The information that is accessed will be the information of the last interchange that was processed at the table level accessed by the rules. This must be taken into consideration when using rules that access these tables.

See *Table access examples* on page 3 - 8 for examples.

Note

Refer to this table as **Interchange**.

- ▶ TESTMODE
 - ▶ CONTROLNUMBER
 - ▶ VERSION
 - ▶ APPFIELD1
 - ▶ APPFIELD2
 - ▶ APPFIELD3
 - ▶ APPFIELD4
 - ▶ APPFIELD5
 - ▶ APPFIELD6
 - ▶ AGENCY
 - ▶ PARTNERKEY
 - ▶ TABLEKEY (Select only)
-

Group These are the field names that are available when using the select or update extended rules on the Group Table (Group_tb).

Note

Can only be accessed from the following map types: Group Break or Build, Transaction Break, Export, and Interchange Build.

The information that is accessed will be the information of the last group that was processed at the table level accessed by the rules. This must be taken into consideration when using rules that access these tables.

See *Table access examples* on page 3 - 8 for examples.

Note

Refer to this table as **Group**.

- ▶ TESTMODE
 - ▶ CONTROLNUMBER
 - ▶ FUNCTIONALGROUPID
 - ▶ VERSION
 - ▶ APPFIELD1
 - ▶ APPFIELD2
 - ▶ APPFIELD3
 - ▶ APPFIELD4
 - ▶ APPFIELD5
 - ▶ APPFIELD6
 - ▶ AGENCY
 - ▶ TABLEKEY (Select only)
-

**GenericEnvelope
Segment**

These are the field names that are available when using the select extended rule on the Generic Envelope Segment Table (GenericEnvelopeSegment_tb).

Note

Refer to this table as **GenericEnvelopeSegment**.

- ▶ CONTROLNUMBER
- ▶ SUBCOUNT
- ▶ FIELD1
- ▶ FIELD2
- ▶ FIELD3
- ▶ FIELD4
- ▶ FIELD5
- ▶ FIELD6
- ▶ FIELD7
- ▶ FIELD8
- ▶ FIELD9
- ▶ FIELD10
- ▶ FIELD11
- ▶ FIELD12
- ▶ FIELD13
- ▶ FIELD14
- ▶ FIELD15
- ▶ FIELD16
- ▶ FIELD17
- ▶ FIELD18
- ▶ FIELD19
- ▶ FIELD20
- ▶ FIELD21
- ▶ FIELD22
- ▶ FIELD23
- ▶ FIELD24
- ▶ FIELD25
- ▶ FIELD26
- ▶ FIELD27
- ▶ FIELD28
- ▶ FIELD29
- ▶ FIELD30

(Continued on next page)

**GenericEnvelope
Segment (contd.)**

- ▶ FIELD31
 - ▶ FIELD32
 - ▶ FIELD33
 - ▶ FIELD34
 - ▶ FIELD35
 - ▶ FIELD36
 - ▶ FIELD37
 - ▶ FIELD38
 - ▶ FIELD39
 - ▶ FIELD40
-

Error Messages

Contents

- ▶ Overview A - 2
 - ▶ Compile Error Messages A - 3
 - ▶ Gentran:Server Error Messages A - 16
 - ▶ Translator Report Error Messages. A - 22
-

Overview

Introduction

The Gentran:Server error messages and other informational messages are noted in the Compile Errors dialog box when you compile the translation object or in the Error section of the Extended Rules dialog boxes when you compile an extended rule containing errors prior to compiling the translation object, and when you commit an erroneous action in Gentran:Server.

The informational messages are dependent on the context of the program, and are intended to be self-explanatory. The error messages are described in the topics accessed below, along with the actions you can take to correct the problem.

In this appendix

This appendix explains how to determine the action you should take when you receive an error message.

Compile Error Messages

Introduction The Compile Error Messages are displayed in the Compile Errors dialog box if you compile a translation object with errors. Or, error messages are displayed in the Error section of an Extended Rule dialog box if you compile the extended rule containing errors prior to compiling the translation object. After you correct the cause of the errors, click the **Compile** button again to ensure that the rule is error-free.

Messages The compile error messages are listed by the four- or five-digit message number and the error message text. The error definitions contain the actions that you can take to correct the problem (if appropriate) and a description that includes possible causes of the error.

Msg ID	Message Text	Explanation/Your Action
1000	expected '.'	<p>Explanation The rule does not have the required “.” between a group name and a field name.</p> <p>Your action Insert a “.” between the group name and field name.</p>
1001	no statement to compile	<p>Explanation The rule is does not contain any statements.</p> <p>Your action Add a statement or statements to the rule.</p>
1002	unexpected end of program	<p>Explanation The rule was not complete.</p> <p>Your action Finish the rule.</p>
1003	expected ','	<p>Explanation The rule does not have the required “,” between parameters.</p> <p>Your action Insert a “,” between parameters.</p>
1004	expected ';'.	<p>Explanation A statement was not terminated properly.</p> <p>Your action Terminate the statement in an appropriate manner.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
1006	no statements to compile	<p>Explanation The body of an IF/ELSE or WHILE condition was empty.</p> <p>Your action Complete the body of the unfinished condition.</p>
1007	syntax error ...	<p>Explanation The map component contains a syntax error.</p> <p>Your action Correct the syntax error.</p>
1008	expected '#'	<p>Explanation The rule does not have the required “#” before a field name.</p> <p>Your action Insert a “#” before the field name.</p>
2000	group ... undefined	<p>Explanation The rule references a group that does not exist.</p> <p>Your action Change the reference to an existing group or delete the reference.</p>
2001	... is not a member of ...	<p>Explanation The rule references a field that does not belong to the specified group.</p> <p>Your action Change the reference to an existing field in the specified group.</p>
2002	insufficient indices to access group ...	<p>Explanation The rule does not give the full addressing for a group.</p> <p>Your action Complete the addressing for the group.</p>
2003	too many indices to access group...	<p>Explanation The rule uses too many addresses for the group.</p> <p>Your action Address the group correctly.</p>
2004	... has not been defined	<p>Explanation The rule references an undefined variable.</p> <p>Your action Define the variable in the declarations section.</p>

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
2005	out of temporary variables	<p>Explanation The rule could not be compiled because some expressions are too complex.</p> <p>Your action Simplify the expressions and compile the rule again.</p>
2008	field type unknown	<p>Explanation The compiler was unable to determine the type of a field.</p> <p>Your action Verify that a Data Type is selected for this field.</p>
2009	cannot reference a local field with no current group	<p>Explanation The rule (probably pre- or post-session) references a local field, but is not associated with any group.</p> <p>Your action Reference the field using the proper addressing.</p>
2010	instances of '%1' are transient and cannot be accessed	<p>Explanation The rule references an output group using full addressing. This form of addressing is only appropriate for input groups.</p> <p>Your action Reference the output group with the proper address.</p>
2011	no field specified	<p>Explanation The rule omits a field reference.</p> <p>Your action Add the field reference to the rule.</p>
2012	group ... is promoted and cannot be accessed in this manner	<p>Explanation You attempted to access a promoted group in a COUNT or DELETE rule.</p> <p>Your action Do not COUNT or DELETE a promoted group.</p>
2100	... is not an array variable	<p>Explanation The rule uses array indexing for a variable that is not an array.</p> <p>Your action Use the proper indexing for the variable.</p>

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
2101	...: array index required	<p>Explanation The rule uses an array variable without using the necessary array indexing.</p> <p>Your action Add the necessary indexing to the array variable.</p>
2102	...: array overflow	<p>Explanation The rule uses an invalid array index.</p> <p>Your action Use the proper array index for the array variable.</p>
2103	only one wildcard index is permitted	<p>Explanation The rule uses more than one wildcard index.</p> <p>Your action Only one wildcard index is permitted per rule.</p>
2104	a wildcard index must be specified	<p>Explanation The rule does not specify a wildcard index when required.</p> <p>Your action Add a wildcard index where necessary.</p>
2200	expected a string	<p>Explanation A required string or string variable is not supplied.</p> <p>Your action Add the required string or string variable.</p>
2201	string overflow	<p>Explanation A string overflow occurred.</p> <p>Your action Verify that the size of a target string is equal to or greater than the source string.</p>
3000	array size expected in declaration of ...	<p>Explanation Invalid array declaration.</p>
3001	declaration of ... missing ']'	<p>Explanation Invalid array declaration.</p>
3002	string size expected	<p>Explanation Invalid string declaration.</p>
3003	string size missing ']'	<p>Explanation Invalid string declaration.</p>

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
3004	variable name expected	Explanation Invalid variable declaration.
3005	... already defined	Explanation Two variables with the same name are defined at the same scope. Your action Rename one of the two variables.
3006	expected an accumulator number, found ...	Explanation Invalid accumulator reference.
3007	... is not a valid accumulator number	Explanation Invalid accumulator reference.
4000	expected a numeric expression, found ...	Explanation You specified something other than the expected numeric expression. Your action Specify the correct numeric expression.
4001	expected a term, found ...	Explanation You specified something other than the expected term. Your action Specify the correct term.
4002	expected '+' or '-'	Explanation You specified something other than the expected "+" or "-". Your action Specify "+" or "-".
4003	expected '*' or '/'	Explanation You specified something other than the expected "*" or "/". Your action Specify "*" or "/".
4004	expected ')'	Explanation You specified something other than the expected ")". Your action Specify ")".

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
4005	expected a factor, found ...	Explanation Invalid numeric expression.
4006	expected '('	Explanation You specified something other than the expected “(”. Your action Specify “(”.
4007	... is of incorrect type	Explanation The specified expression is of an incorrect type. Your action Specify the correct type for the expression.
4008	expected a relational operator	Explanation You specified something other than the expected relational operator. Your action Specify the correct relational operator.
4009	missing argument	Explanation A required parameter was omitted. Your action Add the required parameter.
4010	assignment expected	Explanation The assignment operator was omitted from an assignment statement. Your action Add the correct assignment operator to the statement.
4011	operator ... requires two arguments	Explanation Only one parameter was supplied for a binary operator. Your action Supply a second parameter for the binary operator.
4012	converting real to integer may lose significant digits	Explanation You are converting a real number to an integer. Your action Verify that losing decimal places is acceptable in this instance.

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
4013	expression too complex, use sub-expressions	Explanation A mathematical expression contains too many terms. Your action Group some of the terms of the mathematical expression in parentheses.
4014	expected an integer, found ...	Explanation The compiler expected to find an integer instead of [...]. Your action Supply the correct value for [...].
4015	no valid condition specified	Explanation You did not specify a valid IF or WHILE condition. Your action Supply a valid condition in the IF or WHILE statement.
4100	expected a date, found ...	Explanation A date is required but not supplied. Your action Specify a date.
4101	expected a date modifier, found ...	Explanation A date modifier was required but not supplied. Your action Specify a date modifier.
4102	... is not a date	Explanation Invalid date expression.
5000	THEN expected	Explanation The compiler expected a THEN condition.
5001	DO expected	Explanation The compiler expected a DO condition.
5002	END expected	Explanation The compiler expected a END condition.
5004	FROM expected	Explanation The compiler expected a FROM condition.
5005	INTO expected	Explanation The compiler expected a INTO condition. (Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
5006	END without BEGIN	<p>Explanation An END statement was found without a corresponding BEGIN statement.</p> <p>Your action Insert a BEGIN statement in the correct location.</p>
5016	misplaced 'BREAK'	<p>Explanation The compiler found BREAK outside of a loop.</p> <p>Your action Remove the BREAK or place it inside a loop.</p>
5017	misplaced 'CONTINUE'	<p>Explanation The compiler found CONTINUE outside of a loop.</p> <p>Your action Remove the CONTINUE or place it inside a loop.</p>
5018	too many parameters	<p>Explanation Too many parameters were supplied for a function.</p> <p>Your action Remove the unnecessary parameters.</p>
5019	too few parameters	<p>Explanation Too few parameters were supplied for a function.</p> <p>Your action Add the necessary parameters.</p>
6000	expected a database table name	<p>Explanation The compiler expected a database table name.</p> <p>Your action Insert a database table name in the necessary location.</p>
6001	expected a database column name	<p>Explanation The compiler expected a database column name.</p> <p>Your action Insert a database column name in the necessary location.</p>
6002	too many column receivers specified	<p>Explanation The number of columns specified did not match the number of receivers in a SELECT statement.</p> <p>Your action Correlate the number of columns specified in the SELECT statement to the number of receivers.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
6003	too few column receivers specified	<p>Explanation The number of columns specified did not match the number of receivers in a SELECT statement.</p> <p>Your action Correlate the number of columns specified in the SELECT statement to the number of receivers.</p>
6004	WHERE expected	<p>Explanation The compiler expected a WHERE statement.</p>
6005	expected a database key	<p>Explanation An invalid database key was specified in a WHERE statement.</p> <p>Your action Specify the correct database key in the WHERE statement.</p>
6006	WHERE not allowed with this database table	<p>Explanation A WHERE statement is not necessary to access the specified database table.</p> <p>Your action Remove the WHERE statement.</p>
6007	invalid key combination	<p>Explanation The combination of keys specified in the WHERE statement is not a valid unique compound key to the table.</p> <p>Your action Specify a valid combination of keys in the WHERE statement.</p>
7000	... is not a valid seek type	<p>Explanation You used an invalid seek type in FSEEK.</p> <p>Your action Use BEGIN, END, or CURRENT.</p>
20001	Record ..., Field ... : date field missing date format	<p>Explanation The specified field does not have a date format.</p> <p>Your action Edit the field and choose a date format.</p>
20003	Field ... : constant used in standard rule does not exist	<p>Explanation The standard rule for the specified field uses an invalid constant.</p> <p>Your action Correct the standard rule or create the constant.</p>

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
20004	Field ... : code list used in standard rule does not exist	<p>Explanation The standard rule for the specified field uses an invalid code list.</p> <p>Your action Correct the standard rule or create the code list.</p>
20005	Field ... : the qualifier field specified in a use constant standard rule is invalid	<p>Explanation The standard rule for the specified field uses an invalid qualifier.</p> <p>Your action Correct the standard rule.</p>
20006	Field ... : the field specified to store the code description in a use code standard rule is invalid	<p>Explanation The standard rule for the specified field designates an invalid field for the description.</p> <p>Your action Correct the standard rule.</p>
20007	Record ... : the specified key field ... : uses an undefined constant	<p>Explanation The key field for the specified record uses an invalid constant.</p> <p>Your action Correct the key field.</p>
20008	Record ... : the specified key field ... : uses an undefined code list	<p>Explanation The key field for the specified record uses an invalid code list.</p> <p>Your action Correct the key field.</p>
20010	Record ... : the specified key field ... : is inactive	<p>Explanation The key field for the specified record is inactive.</p> <p>Your action Activate the key field.</p>
20700	only one binary data and one binary length field are permitted	<p>Explanation You have more than one binary data and/or more than one binary length element in one segment.</p> <p>Your action Remove the additional binary data and/or binary length elements from the segment.</p>

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
20701	binary length must precede binary data	<p>Explanation The binary length element must be sequenced <i>before</i> the binary data element in the segment so the translator knows how much data to expect.</p> <p>Your action Move the binary length element to before the binary data element.</p>
20702	incomplete binary data	<p>Explanation You marked a segment as binary, but did not include either a binary length element or a binary data element (or both).</p> <p>Your action Add a binary length element and/or a binary data element to the segment.</p>
20703	group ... has no active child objects	<p>Explanation You tried to compile the translation object, but the specified group is empty.</p> <p>Your action Activate at least one child object in the group.</p>
20704	Element ..., Attribute ...: enumerated attribute declared without accompanying standard rule	<p>Explanation The specified XML attribute is configured to use an enumeration but there is no code list standard rule that defines the allowed values.</p> <p>Your action Define a use code list standard rule for the specified attribute.</p>
20705	Element ..., Attribute ...: code list used in enumerated attribute does not exist.	<p>Explanation The code list for the specified enumerated XML attribute does not exist.</p> <p>Your action Define the code list.</p>
20706	Element ..., Attribute ...: default value is not valid ...	<p>Explanation The specified default value is not in the code list for this XML attribute.</p> <p>Your action Ensure that the specified default value is in the code list.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
20707	Codelist ..., Attribute ...: value used in enumerated attribute code list does not match XML NMTOKEN production	<p>Explanation A value in the enumeration code list is not valid for XML.</p> <p>Your action Verify that all the value specified in the code list are valid (legal) for an XML attribute.</p>
20708	Entity ... : Entity value is invalid	<p>Explanation The value of the specified entity is not valid (legal) in XML.</p> <p>Your action Correct the entity value.</p>
20709	... : Illegal use of reference character	<p>Explanation An entity reference was not terminated properly.</p> <p>Your action Correct the entity reference.</p>
20710	... : Malformed character reference encountered	<p>Explanation An XML character reference was not terminated properly.</p> <p>Your action Correct the character reference.</p>
20711	... : Invalid character referenced	<p>Explanation An XML character reference is invalid.</p> <p>Your action Correct the character reference.</p>
20712	... : Reference to undefined entity encountered	<p>Explanation The referenced entity is not defined.</p> <p>Your action Define the correct entity.</p>
20713	... : Circular entity references encountered	<p>Explanation An entity references an entity that in turn references the original entity.</p> <p>Your action Do not reference the original entity in a circular manner.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
20714	...: default does not match the attribute type	Explanation The default value is the wrong type for the attribute. Your action Either correct the type of the attribute or the default value.
20715	...: Contains illegal character ('<')	Explanation The default value contains the illegal character '<'. Your action Correct the default value.

Gentran:Server Error Messages

Introduction The Gentran:Server error messages are displayed when you commit an erroneous action. When the system displays an error message, you must first acknowledge the message by either clicking **OK**, and then take the appropriate action.

Messages The error messages are listed alphabetically below by the first letter of the error message text. The error definitions contain the actions that you can take to correct the problem (if appropriate) and a description that includes possible causes of the error.

Message Text	Explanation/Your Action
A code list entry must have a code value	Explanation You attempted to add a code without an associated code value.
A code list must have an element id	Explanation You attempted to create a code list without an element ID.
A code list for this element already exists. You must use another element id or delete the original code list first	Explanation You attempted to create a code list with the same element ID as an already existing code list. Your action Either change the element ID or delete the original code list.
A condition field is required	Explanation You established a conditional relationship that requires a condition field, but did not specify a condition field.
A data type is required.	Explanation You did not specify the Data Type of a field.
A field must have a name that is unique within its parent group	Explanation You tried to give a field a name that is already in use by another field within the same group (not necessarily the same record).
A group must have a unique name	Explanation You tried to give a group a name that is already used by another group or record. (Continued on next page)

Message Text	Explanation/Your Action
A group or record with a maximum usage of 1 cannot be split or promoted	<p>Explanation You attempted to split or promote a single-occurrence group or record. This is not a valid action.</p>
A key field has been selected but no key value has been specified.	<p>Explanation You established a key field without specifying a key value.</p>
A Memory Exception has occurred.	<p>Explanation A system error occurred.</p> <p>Your action Exit Gentran:Server and restart Windows.</p>
A name must be entered	<p>Explanation You did not specify a name for an object.</p>
A problem occurred while attempting to close loop This is probably due to incorrect standards.	<p>Explanation An error occurred when reading from the standards.</p>
A record must have a unique name	<p>Explanation You tried to give a record a name that is already used by another group or record.</p>
A Resource Exception has occurred.	<p>Explanation A system error occurred.</p> <p>Your action Exit Gentran:Server and restart Windows.</p>
A serious error was encountered whilst accessing the clipboard and the action was abandoned	<p>Explanation The system aborted a cut, copy, or paste operation because a serious error occurred.</p> <p>Your action Perform the cut, copy, or paste operation again.</p>
A system error was encountered while compiling the translation object	<p>Explanation While compiling a translation object, a system-related problem, such as lack of memory, prevented the compile from completing.</p> <p>Your action Close unnecessary applications to free memory, and reboot your computer (if necessary), and then recompile the translation object.</p> <p style="text-align: right;">(Continued on next page)</p>

Message Text	Explanation/Your Action
A TDF could not be generated due to a lack of available memory. Either restart Windows or close down other applications, then try again	<p>Explanation While generating a TDF, the system ran out of memory.</p> <p>Your action Exit Gentran:Server, restart Windows and regenerate the TDF.</p>
An accumulator must be chosen for this entry	<p>Explanation You did not specify the accumulator to be used in a Use Accum standard rule.</p>
An invalid usage count has been entered	<p>Explanation You entered an invalid usage count for a record or group.</p>
No actions have been chosen for this accumulator entry	<p>Explanation You selected an accumulator but did not specify any actions for it in a Use Accum standard rule.</p>
No alternate accumulator has been selected	<p>Explanation In a Use Accum standard rule, you attempted to create an accumulator entry that used an alternate accumulator, but did not specify which alternate accumulator should be used.</p>
No character ranges have been specified for this token.	<p>Explanation You did not specify the characters allowed for a syntax token.</p>
No fields have been used in this relationship	<p>Explanation You established a conditional relationship but did not specify the fields involved.</p>
No more accumulator entries can be created	<p>Explanation You attempted to create more than six accumulator entries in a Use Accum standard rule.</p>
No more field mappings can be created	<p>Explanation You attempted to create more than eight field mappings for a Select standard rule.</p> <p style="text-align: right;">(Continued on next page)</p>

Message Text	Explanation/Your Action
The constant ID must be unique.	<p>Explanation You entered a literal constant identifier in the ID field that is already used in an existing constant.</p> <p>Your action Type a unique literal constant identifier in the ID field.</p>
The contents of the clipboard cannot be pasted into the translation object at this point	<p>Explanation Either the Clipboard does not contain copied or cut data, or you are trying to paste XML information into a positional file format or paste position information into an XML file format.</p>
The file format cannot be deleted	<p>Explanation You tried to delete a file format object.</p>
The Maximum Usage must be greater than zero and not less than the Minimum Usage.	<p>Explanation You entered an invalid maximum usage count.</p>
The number of entries to be split must be greater than zero and less than the maximum number of entries in the original	<p>Explanation You entered an invalid number in the Split dialog box. The number to be split must be greater than zero and less than the maximum number of entries in the original.</p>
The translation object could not be compiled	<p>Explanation The translation object is not ready to be compiled.</p>
The token code must be unique.	<p>Explanation You attempted to create a syntax token with the same token identifier as an existing syntax token.</p> <p>Your action Type a unique token identifier in the Token field.</p>
These fields cannot be linked because the input field is deeper than the output field. The translator would be unable to address the input field.	<p>Explanation You are attempting to create an invalid link. A valid link involves an Input field and an Output field that are at the same level.</p>
This code already exists. You must use another code or delete the original code first	<p>Explanation You attempted to add a code to a code list that already contained that code.</p> <p style="text-align: right;">(Continued on next page)</p>

Message Text	Explanation/Your Action
This field cannot be linked to another field	<p>Explanation Due to an unknown error, Gentran:Server was unable to begin creating the link.</p>
This file is not a valid Gentran translation object	<p>Explanation You attempted to open (load) a file that is not a translation object.</p>
You have entered an invalid character or character code in Record Delimiter 1	<p>Explanation You entered an invalid Record Delimiter 1 on the Positional File Format Properties dialog box.</p>
You have entered an invalid character or character code in Record Delimiter 2	<p>Explanation You entered an invalid Record Delimiter 2 on the Positional File Format Properties dialog box.</p>
You have entered an invalid decimal separator or character code.	<p>Explanation You entered an invalid decimal separator.</p>
You have entered an invalid element delimiter character or character code	<p>Explanation You entered an invalid element delimiter.</p>
You have entered an invalid Pad character or character code	<p>Explanation You entered an invalid pad character for a positional field.</p>
You have entered an invalid release character or character code	<p>Explanation You entered an invalid release character.</p>
You have entered an invalid sub-element delimiter character or character code	<p>Explanation You entered an invalid sub-element delimiter.</p>
You have entered an invalid tag delimiter character or character code	<p>Explanation You entered an invalid tag delimiter.</p>
You must enter a description of the translation object.	<p>Explanation You attempted to save the translation object without entering the translation object description on the Translation Object Details dialog box.</p> <p>Your action Type the translation object description in the Description field on the Translation Object Details dialog box. This field must be unique because it is used by the system to identify the map.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

Message Text	Explanation/Your Action
You must enter a subtable name.	<p>Explanation For a Select or Update standard rule, the selected table requires you to specify a table name in the Sub Table field.</p>
You must enter a valid character range.	<p>Explanation You entered an invalid character range for a syntax token.</p>
You must enter a valid syntax token.	<p>Explanation You did not enter a valid token identifier in the Token field when creating or editing a syntax token.</p>
You must enter a value for this constant.	<p>Explanation You did not specify a value for the constant in the Value field when creating or editing a syntax token.</p>
You must enter an ID.	<p>Explanation You did not specify a literal constant identifier for the constant in the ID field when creating or editing a syntax token.</p>
You must enter the name of the author of this translation object	<p>Explanation You attempted to save the translation object without entering the name of the translation object author on the Translation Object Details dialog box.</p>
You must select a constant type.	<p>Explanation You did not select a constant type from the Type list when creating or editing a constant.</p>
You need to specify Input and Output File Types for the new translation object	<p>Explanation You clicked OK on the Create New Translation Object dialog box without selecting both the Input Format Type and Output Format Type.</p>

Translator Report Error Messages

Introduction The Document and Interchange Translator Report error messages are displayed on the Document Translator Report and the Interchange Translator Report under the Message Number and Message columns.

Message number column The Message Number column on the translator report contains a prefix (INF, EDI, or POS), a dash (“-”), and a four digit number that identifies the error. The prefixes are described in the following table

Prefix	Description
INF	Used only with information messages, which are not defined in this chapter because they are intended to be self-explanatory.
EDI	Used with all the messages listed below that are not informational. It is used if the error is related to an EDI file.
POS	Used with all the messages listed below that are not informational. It is used if the error is related to a positional flat file.

Message column The Message column on the translator report contains the actual error message text.

Translator error messages

The translator report error messages are listed below by the last three digits of the message number and the error message text.

Msg ID	Message Text	Explanation/Your Action
100	Mandatory Element Missing	<p>Explanation An element that the translation object designated as “Mandatory” was not created in an Outbound document or was not received in an Inbound document.</p> <p>Your Action Use the Segment/Record ID, Sequence, and Element fields on the Translator Report to determine which mandatory element in the document is missing.</p> <p><u>Outbound</u> If the document was entered using the Document Editor, open the document and complete the missing field. If you imported the document into your system, delete the document and then import that document after the import file has been corrected. See Using Documents in the <i>User’s Guide</i> for more information on opening and deleting, and importing documents.</p> <p><u>Inbound</u> Contact your trading partner and determine what action you should take.</p>
110	Incorrect Element Format	<p>Explanation An element was entered or received with an incorrect format. Some examples of incorrect format are: a numeric field that contains non-numeric characters, and a field that exceeds the maximum length or is less than the minimum length (as defined in the standard) and invalid dates.</p> <p>Your Action Use the Segment/Record ID, Sequence, and Element fields on the Translator Report to determine which element in the document is invalid.</p> <p><u>Outbound</u> Correct the data source.</p> <p><u>Inbound</u> Contact your trading partner and determine what action you should take.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
120	Too Many Components in Composite	<p>Explanation A composite element in a document you received has more component elements (sub-elements) than allowed by the standard.</p> <p>Your Action Use the Segment/Record ID, Sequence, and Element fields on the Translator Report to determine which element in the document is invalid.</p> <p><u>Outbound</u> If the document was entered using the Document Editor, open the document and correct the invalid field. If you imported the document into your system, delete the document, correct the data, and then import that document again. See Using Documents in the <i>User's Guide</i> for more information on opening and deleting, and importing documents.</p> <p><u>Inbound</u> Contact your trading partner and determine what action you should take.</p>
130	Invalid Conditional Relationship	<p>Explanation A conditional relationship in the document is not valid.</p> <p>Your Action Use the translator report to determine where in the document the error occurred.</p> <p><u>Outbound</u> If the document was entered using the Document Editor, open the document and correct the conditional relationship. If you imported the document into your system, delete the document and then import that document again.</p> <p><u>Inbound</u> Contact your trading partner and determine what action you should take.</p>
140	Implicit Rule Failure	<p>Explanation A validation rule set up against this field failed in the translator. Typically, this occurs when the Exclusive flag is set for a standard rule, and the field value does not match the data table.</p> <p>Your Action Check the data value that you received against the valid data that is allowed for the field.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
200	Mandatory Component Missing	<p>Explanation A component (sub-element) of a composite element that the translation object designated as “Mandatory” was not created in an Outbound document or not received in an Inbound document.</p> <p>Your Action Use the Segment/Record ID, Sequence, Element, and Composite fields on the Translator Report to determine which mandatory component in the document is missing.</p> <p><u>Outbound</u> If the document was entered using the Document Editor, open the document and complete the missing field. If you imported the document into your system, delete the document and then import that document again.</p> <p><u>Inbound</u> Contact your trading partner and determine what action you should take.</p>
210	Incorrect Component Format	<p>Explanation A component (sub-element) of a composite element that the translation object designated as “Mandatory” was entered (Outbound) or received (Inbound) with an incorrect format. Some examples of incorrect format are: a numeric field that contains non-numeric characters, and a field that exceeds the maximum length or is less than the minimum length (as defined in the standard).</p> <p>Your Action Use the Segment/Record ID, Sequence, Element, and Composite fields on the Translator Report to determine which element in the document is invalid.</p> <p><u>Outbound</u> If the document was entered using the Document Editor, open the document and correct the invalid field. If you imported the document into your system, delete the document, correct the data, and then import that document again.</p> <p><u>Inbound</u> Contact your trading partner and determine what action you should take.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
220	Component Delimiter	<p>Explanation A component delimiter was encountered instead of the expected element.</p> <p>Your Action Contact either/or your trading partner or the translation object creator and determine what action you should take.</p>
300	Mandatory Segment	<p>Explanation A segment that the translation object designated as “Mandatory” was not created in an Outbound document or was not received in an Inbound document.</p> <p>Note This error can be generated in a variety of circumstances. The most common is that the input data sequence does not correspond to the data sequence defined in the translation object used to translate the data. If this is the case, the information provided with the message may indicate a segment in the data.</p> <p>Your Action Use the Segment/Record ID field on the Translator Report to determine which mandatory segment in the document is missing.</p> <p><u>Outbound</u> If the document was entered using the Document Editor, open the document and key data into the fields that are necessary to generate the segment. If you imported the document into your system, delete the document, add the data that is necessary to generate the segment, and then import that document again.</p> <p><u>Inbound</u> Contact your trading partner and determine what action you should take.</p>
310	Invalid Loop Start/ End Structure	<p>Explanation An invalid Loop Start/Loop End was found in an Inbound document.</p> <p>Your Action Use the information in the translator report to determine which LS/LE pairing is invalid. Contact your trading partner and determine what action you should take.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
315	Invalid Segment or Record Structure	<p>Explanation A segment (in an EDI file) or a record (in a positional flat file) in an inbound file did not match what the translation object was expecting.</p> <p>Your Action From viewing the information in the translator report and viewing the “Raw EDI” interchange, determine which segment or record is invalid. Contact your trading partner to determine what action you should take. See How to View Interchanges in the <i>User’s Guide</i> for more information.</p>
405	Unknown Partner	<p>Explanation An Interchange was received but the system cannot determine which partner sent it.</p> <p>Your Action From viewing the information in the translator report and viewing the “Raw EDI” interchange, determine which partner sent you the interchange. If the partner is not listed on your system, create the partner and a relationship, and attach the interchange to that partner. If the partner already exists on your system, attach the interchange to that partner and then determine why the system did not automatically identify the partner.</p>
410	Header/Trailer Control Numbers do not match	<p>Explanation The control numbers on the header and trailer do not match, as specified by the standard.</p> <p>Your Action Check the “Raw EDI” view to determine which control numbers are in the EDI file, and then contact your trading partner to determine what action you should take. See How to View a Document in the <i>User’s Guide</i> for more information on viewing a document.</p>
415	Control Total Incorrect	<p>Explanation The EDI control total in the Segment Identified field of the translator report does not equal the value that was calculated by the Compliance Checker.</p> <p>Your Action Check the “Raw EDI” view to determine what the control total should be, and then contact your trading partner to determine what action you should take. See How to View a Document in the <i>User’s Guide</i> for more information on viewing a document.</p> <p style="text-align: right;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
420	Unknown Relationship	<p>Explanation A document was received but the Partner Profile for that partner does not include a corresponding Inbound Relationship.</p> <p>Your Action From the viewing the information in the translator report and viewing the “Raw EDI” interchange, determine which relationship the document requires and create that inbound relationship for the partner. See How to View Interchanges in the <i>User’s Guide</i> for more information on viewing interchanges. See How to Create a New Inbound Relationship in the <i>User’s Guide</i> for more information on creating inbound relationships.</p>
460	Invalid Test Mode Flag	<p>Explanation The partner relationship was found but the test mode flag in the data did not match the test mode defined in the inbound partner relationship.</p> <p>Your Action Change the test mode of the inbound partner relationship to match the test mode of the data. See How to Create a New Inbound Relationship in the <i>User’s Guide</i> for more information on creating inbound relationships. Then, ask your trading partner to change the test mode of the data they are sending to match that defined by the inbound relationship.</p>

User Level Error Messages

User-level type errors can occur when extended rules are used in the map.

The user-level error messages are listed below in numerical order based on their Message ID. The table also includes an explanation of the error and an action you can take to correct the error.

Msg ID	Message Text	Explanation/Your Action
800	Batch File Header Section Missing	<p>Explanation The system cannot find a header section record within the batch file.</p> <p>Your action Contact your Trading Partner.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
801	Unexpected Header Segment ID	<p>Explanation A value other than “00” is in the header section record segment Identifier field (701).</p> <p>Your action Contact your Trading Partner.</p>
802	Unexpected Transmission Type	<p>Explanation A transmission value other than T (Transaction), R (Response), or E (Error) is in the transmission type (880-K6) field of the header section record.</p> <p>Your action Contact your Trading Partner.</p>
803	Sender ID Missing	<p>Explanation The system cannot find a value in the sender ID (880-K1) field in the Header Section Record.</p> <p>Your action Contact your Trading Partner.</p>
804	Invalid Batch Number	<p>Explanation The batch number field (806-5C) contains a format other than the YYDDD format, where:</p> <ul style="list-style-type: none"> ▶ YY=Year ▶ DDD=Julian date <p>Your action Contact your Trading Partner.</p>
805	Unexpected File Type	<p>Explanation A value other than P (Production) or T (Test) is in the File Type field (702).</p> <p>Your action Contact your Trading Partner.</p>
806	Receiver ID Missing	<p>Explanation The system cannot find a value in the receiver ID field (880-K7) of the header section.</p> <p>Your action Contact your Trading Partner.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
807	Batch File Detail Data Record Missing	<p>Explanation The system cannot find a Detail Data Record within the batch file.</p> <p>Your action Contact your Trading Partner.</p>
808	Unexpected Data Segment ID	<p>Explanation A value other than “G1” is in the segment identifier field of the Detail Data Record (701).</p> <p>Your action Contact your Trading Partner.</p>
809	Transaction Reference Number Missing	<p>Explanation The system cannot find a value in the transaction reference number field of the Detail Data Record (880-K5).</p> <p>Your action Contact your Trading Partner.</p>
810	Batch File Trailer Record Missing	<p>Explanation The system cannot find the trailer record within the batch file.</p> <p>Your action Contact your Trading Partner.</p>
811	Invalid Trailer Segment ID	<p>Explanation A value other than “99” is in the segment identifier field (701) of the trailer record.</p> <p>Your action Contact your Trading Partner.</p>
812	Unexpected Batch Number	<p>Explanation The batch number field (806-5C) of the trailer record does not match the batch number field (806-5C) of the header record.</p> <p>Your action Contact your Trading Partner.</p>
813	Invalid Record Count	<p>Explanation The record count field (751) of the trailer record contains an invalid number.</p> <p>Your action Contact your Trading Partner.</p>

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
814	Processed DDRs Mismatch Record Count	<p>Explanation The actual count of Detail Data Records is different from the value in the record count field (751) of the trailer record.</p> <p>Your action Contact your Trading Partner.</p>
815	Duplicate Segment Within Transaction	<p>Explanation A segment occurs more than one time within the transaction.</p> <p>Your action Contact your Trading Partner.</p>
816	Unexpected Transaction Code	<p>Explanation A value other than a value listed below is in the transaction code field (103-A3) of the request transaction header segment or response header segment.</p> <ul style="list-style-type: none"> ▶ E1 = Eligibility Verification ▶ B1 = Billing ▶ B2 = Reversal ▶ P1 = Re-bill ▶ P2 = P.A. Request & Billing ▶ P3 = P.A. Inquiry ▶ P4 = P.A. Request Only ▶ N1 = Information Reporting ▶ N2 = Information Reporting Reversal ▶ N3 = Information Reporting Re-bill ▶ C1 = Controlled Substance Reporting ▶ C2 = Controlled Substance Reporting Reversal ▶ C3 = Controlled Substance Reporting Re-bill <p>Your action Contact your Trading Partner.</p> <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Msg ID	Message Text	Explanation/Your Action
817	Invalid Transaction Count	<p>Explanation A value other than a value listed below is in the transaction count field (109-A9) of the request transaction header segment or the response transaction header segment.</p> <ul style="list-style-type: none"> ▶ Blank = not specified ▶ 1 = One Occurrence (Default) ▶ 2 = Two Occurrence (Except Transaction Code E and P) ▶ 3 = Three Occurrence (Except Transaction Code E and P) ▶ 4 = Four Occurrence (Except Transaction Code E and P) <p>Your action Contact your Trading Partner.</p>
818	Processed Transactions Mismatch Transaction Count	<p>Explanation The actual count of the transactions is different from the value in the transaction count field (109-A9) of the response header segment.</p> <p>Your action Contact your Trading Partner.</p>
819	Unexpected Header Response Status Code	<p>Explanation A value other than A (Accepted) or R (Rejected) is in the header response status field (501-F1) of the response header segment.</p> <p>Your action Contact your Trading Partner.</p>
820	Unexpected Transaction Response Status Code	<p>Explanation A value other than a value listed below is in the transaction response status field (112-AN).</p> <ul style="list-style-type: none"> ▶ A = Approved ▶ C = Captured ▶ D = Duplicate of Paid ▶ F= PA Deferred ▶ P = Paid ▶ Q = Duplicate of Capture ▶ R = Rejected ▶ S = Duplicate of Approved <p>Your action Contact your Trading Partner.</p>

Core Translator Error Messages

Core translator type errors can occur when you are receiving or sending data.

The core translator error messages are listed below in numerical order based on their Message ID. The table also includes an explanation of the error and an action you can take to correct the error.

Msg ID	Message Text	Explanation/Your Action
830	Null Character	<p>Explanation The data file contains binary data.</p> <p>Your action Contact your Trading Partner.</p>
833	Unexpected End of File	<p>Explanation A field of a specified length is incomplete.</p> <p>Your action Contact your Trading Partner.</p>
835	Unexpected Start of Text	<p>Explanation The record does not contain End of Text character.</p> <p>Your action Contact your Trading Partner.</p>
836	Unexpected End of Text	<p>Explanation The record does not contain Start of Text character.</p> <p>Your action Contact your Trading Partner.</p>
837	Missing Stream Separator1	<p>Explanation The system is reading non-streamed data with a template from a map where “stream segments” box is unchecked.</p> <p>Your action Contact your Trading Partner.</p>
838	Missing Stream Separator2	<p>Explanation The system reading non-streamed data with a template from a map where “stream segments” box is unchecked.</p> <p>Your action Contact your Trading Partner.</p>
839	Unexpected Separator	<p>Explanation A group separator, segment separator, field separator, STX, or ETX is present in the data.</p> <p>Your action Contact your Trading Partner.</p>

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
840	Unexpected Field Separator	<p>Explanation Based on the map, two field separators occur consecutively or a different separator is expected.</p> <p>Your action Contact your Trading Partner.</p>
841	Unexpected Group Separator	<p>Explanation Based on the map, a group separator occurs when none was specified.</p> <p>Your action Contact your Trading Partner.</p>
842	Missing Group Separator	<p>Explanation Based on the map, a group separator is missing.</p> <p>Your action Contact your Trading Partner.</p>
843	Unexpected Stream Separator1	<p>Explanation Based on the map, a stream separator1 is occurs when none was specified.</p> <p>Your Action Contact your Trading Partner.</p>
845	Positional Tag Defined Beyond EOS	<p>Explanation The position of the transaction code (Positional Record Tag) is specified to start beyond the end of the segment.</p> <p>Your action Change the value of the positional segment under the NCPDP Properties tag to start before the end of the segment.</p>
855	Delimited Segment Has No Fields	<p>Explanation The delimited segment must have at least the mandatory fields.</p> <p>Your action Contact your Trading Partner.</p>
856	Missing Delimited FS After SS	<p>Explanation The delimited field separator is not present after the segment separator.</p> <p>Your action Contact your Trading Partner.</p>

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
857	Field ID Data Does Not Match Map Field ID	<p>Explanation The field ID data is different from the defined delimited field ID tag.</p> <p>Your action Contact your Trading Partner.</p>
858	Unexpected Char(s) in Delimited Field ID	<p>Explanation A separator was found in the delimited field ID data.</p> <p>Your action Contact your Trading Partner.</p>
859	Null Character in Delimited Field ID	<p>Explanation The delimited field ID contains binary data.</p> <p>Your action Contact your Trading Partner.</p>
860	Incomplete Delimited Field ID	<p>Explanation The field ID data is shorter than expected.</p> <p>Your action Contact your Trading Partner.</p>
861	Missing Delimited Field Data	<p>Explanation No data is found after a delimited field ID.</p> <p>Your action Contact your Trading Partner.</p>
862	Null Character in Delimited Field Data	<p>Explanation The delimited field data contains binary data.</p> <p>Your action Contact your Trading Partner.</p>
865	Segment ID Data Does Not Match Map Segment ID	<p>Explanation The segment ID data is different from the defined delimited segment ID field.</p> <p>Your action Contact your Trading Partner.</p>
866	Unexpected Char(s) in Delimited Segment ID	<p>Explanation A separator was found in the delimited segment ID.</p> <p>Your action Contact your Trading Partner.</p>

(Continued on next page)

(Contd) Msg ID	Message Text	Explanation/Your Action
867	Null Character in Delimited Segment ID	<p>Explanation The delimited segment ID contains binary data.</p> <p>Your action Contact your Trading Partner.</p>
868	Incomplete Delimited Segment ID	<p>Explanation The delimited segment ID is shorter than expected.</p> <p>Your action Contact your Trading Partner.</p>
869	Field ID Size Differs From Global Field ID Size	<p>Explanation The computed size of the segment/field ID is different from the value of the global field ID length.</p> <p>Your action Contact your Trading Partner.</p>
885	No Data after STX in Batch Segment	<p>Explanation The batch record does not contain any data.</p> <p>Your action Contact your Trading Partner.</p>
886	Unexpected STX in Batch Segment	<p>Explanation The batch record does not contain an ETX character.</p> <p>Your action Contact your Trading Partner.</p>
887	Missing ETX in Batch Segment	<p>Explanation The batch record does not contain an ETX character.</p> <p>Your action Contact your Trading Partner.</p>

**Core Translator
Error Messages:
sending batch data
using the NCPDP
Data Record field**

Core translator errors can also occur when you are sending batch data using the NCPDP Data Record field to your Trading Partner. Many times the error results because the Batch File string is missing a portion of the Batch File or a value in the string is incorrect.

To prevent errors from occurring verify that the Batch File string follows the detail data record (DDR) format:

<separator>filename<separator>StartOffset<separator>EndOffset<separator>.

The core translator error messages are listed below in numerical order based on their Message ID. The table also includes an explanation of the error and an action you can take to correct the error.

Msg ID	Message Text	Explanation/Your Action
888	Could Not Open NCPDP Data File	<p>Explanation The NCPDP data file either does not exist or it is not readable.</p> <p>Your action Verify that the file you are selecting is the correct filename or supply the information that is missing from the string field.</p>
890	Missing Separator in DDR Info String	<p>Explanation A separator is missing from the DDR string field.</p> <p>Your action Insert a “;” in the DDR string field.</p>
891	Missing Filename in DDR Info String	<p>Explanation The filename is not in the DDR string field.</p> <p>Your action Add the filename to the DDR.</p>
892	Invalid Start byte Offset in DDR Info String	<p>Explanation The system found non-numeric data at start of set.</p> <p>Your action Add a valid start of set (numeric) value.</p>
893	Invalid End Byte Offset in DDR Info String	<p>Explanation The system found non-numeric data at end of set.</p> <p>Your action Add a valid end of set (numeric) value or remove the data if data up to end of file should be processed.</p> <p style="text-align: right;">(Continued on next page)</p>

Msg ID	Message Text	Explanation/Your Action
894	Missing Start/End Byte Offset in DDR Info String	<p>Explanation The Start/End Byte Offset is not in the DDR string field.</p> <p>Your action Add the Start/End Byte Offset to the DDR string field or remove extra separators.</p>
895	Extra Data in DDR Info String	<p>Explanation The DDR string contains extra data.</p> <p>Your action Remove the extra data from the DDR string.</p>
896	Total Bytes Read Mismatch Total Bytes to Read	<p>Explanation The actual number of bytes read is less than the number of bytes specified in the set.</p> <p>Your action Verify the Start/End Byte Offsets or remove the end byte offset if data up to end of file should be processed.</p>
897	Start Byte Offset is Beyond End of File	<p>Explanation The position of the Start Byte Offset is specified to start after the end of the file.</p> <p>Your action Change the start position of the Start Byte Offset.</p>
898	Start Byte Offset is Beyond End Byte Offset	<p>Explanation The position of the Start Byte Offset starts after the position of the End Byte Offset.</p> <p>Your action Change the position of the Start Byte Offset to start before the End Byte Offset, or change the position of the End Byte Offset to start after the Start Byte Offset.</p>

User Exits

Contents

- Overview B - 2
- Definition of ActiveX Terminology B - 3
- Gentran:Server User Exit Overview B - 4
- Examples of User Exits B - 7
- Examples of Automation Servers B - 12
- How to Create a User Exit. B - 15



Overview

Assumptions

Use of this appendix assumes that you:

- Are familiar with the use of ActiveX Automation Servers and languages such as Visual Basic.
- Know the Translator Programming Language (TPL) constructs for creating, manipulating, and deleting ActiveX objects.
- Know the ProgID of the ActiveX object and all its exposed interfaces.
- Understand how and when extended rules are invoked and their scope.

Advanced feature

User exits are an *advanced feature* of Gentrans:Server that should only be used with the above prerequisites.

Restrictions

Gentrans:Server extended rules selectively support ActiveX technology.

Supported

Gentrans:Server extended rules support:

- ActiveX Automation Servers.
- Some *ActiveX Controls*, but only those that work as Automation Servers.

Not supported

Gentrans:Server does not support:

- ActiveX controls that are not ActiveX Automation Servers. Such ActiveX controls must be hosted in a graphical user interface (GUI) display, and therefore cannot be used from extended rules.
 - ActiveX Arrays (e.g., the VT_ARRAY data type modifier)
 - References (e.g., the VT_BYREF data type modifier) except for output parameters in method calls. In this instance, references are only valid for the duration of the method call.
 - Gentrans:Server does not read registry entries or type libraries when compiling extended rules to verify the accuracy of programmatic identifiers (ProgIDs) or interfaces.
 - Extended rules that compare two ActiveX properties or method results, or any combination of properties or method results. This type of comparison is invalid because property and method types are unknown prior to compilation and thus it is not possible to generate the correct comparison code.
-

Definition of ActiveX Terminology

What is ActiveX?

ActiveX is a term that encompasses a set of rules that define how applications should share information. ActiveX grew out of technologies developed by Microsoft, specifically Object Linking and Embedding (OLE) and Component Object Model (COM).

What is an ActiveX automation server?

An ActiveX automation server is an ActiveX component (a .DLL or .EXE program) that can expose part of its functionality, specifically properties and methods, via the IDispatch interface to another program on the system.

Some ActiveX controls function as automation servers.

What is the IDispatch interface?

The IDispatch interface is a standard COM interface. Automation Servers typically expose their methods and properties through this interface.

What is a method?

A method is an action or function that is performed by an object (e.g., a calculation or a search).

What is a property?

A property is a characteristic or parameter of an object (e.g., type, size, or creation date).

What are ActiveX controls?

ActiveX controls are a specifically defined method of implementing ActiveX technologies. Basically an ActiveX control is a software component that executes common tasks and can integrate into the user interface of an application that provides the necessary ActiveX control host functionality.

The ActiveX control specification enables you to build component software that interacts with your application and Gentran:Server. ActiveX controls require a user interface, so they are not appropriate for translation user exits. These controls can be developed in a variety of programming languages, including Visual Basic.

Gentran:Server User Exit Overview

Why use user exits?

User exits allow you to enhance your functionality or fulfill specific requirements that Gentran:Server does not perform during normal translation.

User exits with Gentran

Gentran:Server supports user exits in *all types of translation objects* through extended rules that enable the use of Microsoft's ActiveX technology. This feature allows you to use extended rules to invoke custom functionality that was created as an *ActiveX Automation Server*.

Specifically, you can:

- create objects,
- delete objects,
- query objects,
- access properties, and
- invoke methods.

Note

You can create your own custom functionality as an ActiveX Automation Server, or you can use third party Automation Servers.

Gentran:Server supports ActiveX indexed properties, specifically these types of index:

- variant
- numeric
- string

The index support allows you to use these syntaxes (if they are supported by your Automation server):

```
n = ob.property[1];  
n = ob.property["Count"];  
n = ob.property[ob.method()];
```

Gentran:Server also supports chaining of ActiveX method calls and properties, which simplifies extended rules for Automation servers with moderately complex object models (e.g., ADO).

Example

The following is an example of chaining statements:

```
recordset.fields.item["MessageId"].value
```

Examples of user exits

User exits may be used in the following manner.

- To access your database table to perform cross-reference or lookups instead of using the Gentran:Server tables.
- To perform complex pricing calculations (e.g., involving multiple customers, where they are located, and where the product is sold).

Data types supported

The following table lists the extended rule data types and the corresponding ActiveX data types that are supported for use with user exits:

Extended Rule Data Type	ActiveX Data Type
INTEGER	VT_14
REAL	VT_R8
DATETIME	VT_DATE
STRING	VT_BSTR
OBJECT	VT_DISPATCH

Data type conversion

The Gentran:Server translator automatically converts the extended rule data type to the ActiveX data type, whenever possible. If the conversion cannot be performed, a type mismatch error is written to the Audit Log and the extended rule is immediately terminated. An error is also written to the translator report, if one is used by the operation. Since an export operation does not generate a translator report, any conversion errors during export translation are only written to the Audit Log.

Objects

ActiveX automation servers use the datatype OBJECT, which consists of two elements: properties (a defined set of data) and methods. A method is a function that is defined by the interface of the object.

All objects that are automation servers provide the default interface IDispatch, which exposes the internal functions of an application to Gentran:Server. You can also expose other interfaces.

Example

Internet Explorer exposes the interface IWebBrowser2.

Syntax The object must be declared and then created via the CREATEOBJECT command, by pointing to the object's IDispatch interface. Then, the object's methods may be invoked using the syntax *object.method(parameters)*.

The user may specify the ProgID and an optional InterfaceID (IID). If you just specify the ProgID, Gentran:Server uses the default IDispatch interface. The IID is a machine-readable identifier that uniquely identifies the interface of an object (e.g., {00020300-00B000-C00004005300}). Interfaces can also be identified by their human-readable name (e.g., IDispatch).

Note

Each IID must be enclosed in braces {}.

Location The location of the user exit in the map depends on what the user exit needs to do and what you want the scope of the rule to be. Typically, you might:

- declare an object in the On Begin rule of the Input side of the map (Loop Level Extended Rules dialog box of Input positional file),
- create and use the object in the On Begin rule of the Input side of the map (Loop Level Extended Rules dialog box of Input positional file), and
- delete the object in the On End rule of the Output side (Loop Level Extended Rules dialog box of Output EDI file).

Note

The sample above is intended as an example only. You can declare, create, use, and delete an object in an extended rule for a map component at any hierarchical level, including field level extended rules, if appropriate.

Reference

See *Defining Extended Rules* on page 6 - 12 for more information on creating extended rules for various map components.

Examples of User Exits

Introduction

This section contains examples of the construction of user exits. In this section the following conventions apply:

- Bold typeface is used to express what you need to type.
- Lines that are too long to fit on one line in this section may be continued on the next line using a line-continuation character (`^`).
- A double slash (`//`) introduces comments.

Defining object variable

This example defines a variable that represents an ActiveX Automation Server:

```
object ob;
```

Creating a default interface

This example creates an instance of the default interface of the Internet Explorer ActiveX Automation Server:

```
object ob;  
ob = createobject("InternetExplorer.Application");  
//Creates an instance of the default interface of an ActiveX  
//Automation Server.
```

Creating a specific interface

This example creates an instance of the Internet Explorer ActiveX Automation Server and requests a specific interface:

```
object ob;  
ob = createobject("InternetExplorer.Application", "{EAB22AC1-30C1  
^A-11CR-A7EB-000C05BAE0B}");  
//Creates an instance of a specific interface of an ActiveX Automation  
//Server. In this case, the IID is known (specified in braces).  
//Note:  
//The createobject command is more efficient if you use the IID  
//instead of the interface name.
```

This example creates an instance of the Internet Explorer ActiveX Automation Server, where the interface identifier of the desired interface is unknown, but the name of the interface is `IWebBrowser`:

```
object ob;  
string[50] iid;  
iid = getiid("InternetExplorer.Application", "IWebBrowser2");  
ob = createobject("InternetExplorer.Application", iid);  
//Creates an instance of a specific interface of an ActiveX  
//Automation Server. In this case, the IID is not known, so the  
//ProgID and name of the interface (IWebBrowser) are specified and  
//Gentran:Server looks up the IID. The IID is loaded into the  
//string variable "iid" and then that value is used to create the  
//object.
```

Deleting an object

An object must be deleted before the end of the map that uses it. It is more efficient to delete the object immediately when you no longer need it, although the Gentran:Server translator deletes the object automatically at the end of the map. Also, if you assign one object to another one, both copies of the object must be deleted for that object to be properly unloaded.

This example deletes the object “ob” that was used in the previous examples:

```
deleteobject (ob);
```

Getting a property value

This example obtains a property value from an ActiveX Automation Server:

```
object ob;
string[50] iid;
iid = getiid("InternetExplorer.Application", "IWebBrowser2");
ob = createobject("InternetExplorer.Application", iid);
IF ob.Visible = 1 THEN
BEGIN
//The property value is accessed.
END
//Test to see if the system displays the Internet Explorer.
```

Setting a property value

This example sets a property value in an ActiveX Automation Server:

```
object ob;
string[50] iid;
iid = getiid("InternetExplorer.Application", "IWebBrowser2");
ob = createobject("InternetExplorer.Application", iid);
ob.Visible = 1;
//The property value is set.
//Display Internet Explorer on the desktop.
```

Passing parameters to a method

To pass parameters to a method, use the syntax *objectname.methodname(parameters)*. The Gentran:Server translator automatically converts the extended rule data type of the property to the ActiveX data type, whenever possible. If there is no directly corresponding type in the extended rules, the conversion is performed as well as possible by the operating system.

Note

Gentran:Server relies on default data types. For example, Gentran:Server converts VT_CURRENCY (which is an unknown data type to the translator programming language) to a real or numeric data type. If the conversion cannot be performed, a type mismatch error is written to the Translator Report and the extended rule is immediately terminated.

(Continued on next page)

Passing parameters to a method (contd)

Example

This example navigates to a web page by passing the URL to the Navigate method of Internet Explorer:

```
object ob;
string[50] iid;
iid = getiid("InternetExplorer.Application", "IWebBrowser2");
ob = createobject("InternetExplorer.Application", iid);
ob.Navigate("www.sterlingcommerce.com");
//Navigates to the Sterling Commerce web page by passing the URL to
//the Navigate method of Internet Explorer. Note: URL must be
//enclosed in quotation marks.
```

Returning values in output parameters

To return values in output parameters, use the syntax *objectname.methodname(InputParameter, OUT OutputParameter)*.

Note

- Input and output parameters may occur anywhere in the parameter list.
- Output parameters must be preceded by the OUT keyword.
- When an extended rule variable is used as an output parameter, the ActiveX datatype must match exactly.

Reference

See *Data types supported* on page B - 5 for a list of ActiveX data types and the analogous extended rule data type for each.

Example

This example decrypts data in a field:

```
object ob;
ob = createobject("YourCompany.DecryptionUserExit");
ob.Decrypt("EncryptionKey", OUT #Field_Name);
```

Testing successful object creation

This example tests the value of an object against zero to determine if it was correctly created:

```
object ob;
ob = createobject("YourCompany.DecryptionUserExit");
IF ob = 0 THEN
BEGIN
//Object not created properly; perform task X.
END
//Test to see if the object was created successful. If it was not
//created, perform the specified task.
```

Obtaining another interface of an existing object

This example uses the IID of an object to obtain a different interface of an existing object (object1):

```
object ob, ob2;
ob = createobject("InternetExplorer.Application");
ob2 = queryobject(ob, "{EAB22AC1-30C1-11CF-A7EB-0000C05BAE0B}");
If the IID of the desired interface is unknown, use the getiid function
to determine the correct IID and then use the queryobject function, as
this example demonstrates:
object ob, ob2;
string[50] iid;
ob = createobject("InternetExplorer.Application");
iid = getiid("InternetExplorer.Application", "IWebBrowser2");
ob2 = queryobject(ob, iid);
```

Accessing a database

This example uses a user exit to cross-reference a UPC code with a value in a database to return an SKU number. This example adds the user exit to an invoice (import) map (refer to the Gentran:Server TUTORIAL folder, PET_810.MAP for an example). The user exit code is implemented by increments in the map, with each section added to the logical map component.

On Begin Extended Rule

Type the following code into the On Begin extended rule of the INPUT positional file:

```
//Declare the object
object ob;
//Create an instance of the Visual Basic Active X Automation Server
ob = CreateObject("SCUPCSKU.SKUResolve");
//To create an instance of the C++ ActiveX Automation Server
//use the following command instead:
//ob = CreateObject("UpcSku.Application");

if ob = 0 then
begin
MessageBox("Create Object failed",0);
end
```

INVDETAIL.UPCCODE Field Extended Rule

Type the following code into the UPCCODE field extended rule on the Input side of the map (INVDETAIL record):

```
string [100] msg;
msg = "UPC Code = ";
concat(msg, #UPCCODE, 12);

//Call the automation server
#UPCCODE = ob.ResolveSKU(#UPCCODE);
concat(msg, ", SKU Code = ", 13);
concat(msg, #UPCCODE, len(#UPCCODE));

//Display message box to user listing UPC code and SKU matches
MessageBox(msg, 0);
```

(Continued on next page)

**Accessing a
database
(contd)****On End Extended Rule**

Type the following code into the On End extended rule of the INPUT positional file:

```
//Delete the object  
deleteobject (ob);
```

Note

See *Examples of Automation Servers* on page B - 12 for sample automation servers (written in Visual Basic and C++) that access a Microsoft Access Database to cross-reference a UPC code with an SKU code.

Examples of Automation Servers

Introduction

The following pages contain sample automation servers that access a Microsoft Access Database to cross-reference a UPC code with an SKU code. The first example is written in the Visual Basic programming language and the second one in C++.

Note

Both of the following automation servers could be used with the “Accessing a database” user exit extended rules on *Accessing a database* on page B - 10.

Visual Basic Automation Server

This sample automation server was written in the Visual Basic programming language. It accesses a Microsoft Access Database to cross-reference a UPC code with an SKU code. The syntax is: *ResolveSKU(UPCCode as String)*. Note that after compiling the Automation Server (SCUPCSKU.EXE), it was registered using REGSRV32.EXE because it was not self-registering.

```
//This is a Visual Basic Active X EXE project. All code shown here
//resides
//in the SKUResolve class. (Class 1 was renamed to SKUResolve).
//To access the database, the following reference was included in
//the project: Microsoft DAO 2.5/3.5 Compatibility Library.
//The following line forces the compiler to make sure all variables
//are declared prior to use within the program.
Option Explicit
//This function receives a UPC code as it's only parameter and
//cross-references it with a value in the database to get the
//corresponding SKU code.
Public Function ResolveSKU(UPCCode As String) As String
    Dim DB As Database, RS As Recordset
    Set DB = DBEngine.Workspaces(0).OpenDatabase
    A("c:\GENSRVNT\tutorial\upcsku.mdb")
    Set RS = DB.OpenRecordset("UPCSKU", dbOpenDynaset)
//Make sure there are no trailing spaces on the parameter passed
//to this function.
    UPCCode = Trim(UPCCode)
//Since the database is case-sensitive make sure that the parameter
//has all capitalized letters in it.
    UPCCode = UCase(UPCCode)
    RS.MoveFirst
//Loop until we find the UPC code or until the end of the recordset
//is found.
    While Not RS.EOF
//If we find the UPC code, return the 'SKU code associated with it.
        If RS("UPCCode") = UPCCode Then
            ResolveSKU = RS("SKUCode")
            Exit Function
        End If
        RS.MoveNext
    Wend
//No matching UPC code was found for the parameter passed to the
//function. Return a value stating that there was no SKU found.
    ResolveSKU = "No Record Found"
End Function
```

C++ Automation Server

This sample automation server was written in the C++ programming language. It performs exactly the same function as the previous Visual Basic sample user exit—accessing a Microsoft Access Database to cross-reference a UPC code with an SKU code. Note that after compiling the Automation Server (UPCSKU.DLL), it was registered using REGSRV32.EXE because it was not self-registering.

```
// This is a MFC AppWizard (dll) project with the Automation option
//selected. All code shown here resides in the CUpcSkuMain class
//which is derived from CCmdTarget. When the class was created in
//ClassWizard, the ProgID
//"UpcSku.Application" was specified in the "Createable by type
//ID:" field.
// This ProgID is different than the one used for the VB example so
//that both Automation servers can co-exist in the registry.

// To access the database, a class (CUpcSkuDaoRecordset) derived
//from CDaoRecordset is needed. DAO and Snapshot options were
//selected and the location of the database (Upcsku.mdb) was
//specified during the creation of the class in ClassWizard.
// Include the header file for the derived recordset class
#include "UpcSkuDaoRecordset.h"
// This function is called when the last reference for this
//automation object is released

void CUpcSkuMain::OnFinalRelease()
{
    CCmdTarget::OnFinalRelease();

    // Since we used DAO to access the database,
    // terminate DAO now so the DLL can unload.

    if (AfxOleCanExitApp())
        AfxDaoTerm();
}
// This function receives a UPC code as it's only parameter and
//cross-references it with a value in the database to get the
//corresponding SKU code.

BSTR CUpcSkuMain::ResolveSKU(LPCTSTR UPCCode)
{
    CString strResult;
    CString sUPCCode = UPCCode;

    CDaoDatabase DB;
    DB.Open("C:\\GENSRVNT\\Tutorial\\Upcsku.mdb", FALSE, TRUE);

    CUpcSkuDaoRecordset RS(&DB);
    RS.Open();
```

(Continued on next page)

**C++ Automation
Server
(contd)**

```
// Make sure there are no trailing spaces on the parameter passed
//to this function.

        sUPCCode.TrimRight();

// Since the database is case-sensitive make sure that the
//parameter has all capitalized letters in it.

        sUPCCode.MakeUpper();

        RS.MoveFirst();
// Loop until we find the UPC code or until the end of the recordset
//is found.

        while ( !RS.IsEOF() )
        {
// If we find the UPC code, return the SKU code
// associated with it.

                if ( RS.m_UPCCode == sUPCCode )
                {
                        strResult = RS.m_SKUCode;

                        RS.Close();
                        DB.Close();

                        return strResult.AllocSysString();
                }

                RS.MoveNext();
        }

// No matching UPC code was found for the parameter passed to the
// function. Return a value stating that there was no SKU found.

        strResult = "No Record Found";

        RS.Close();
        DB.Close();

        return strResult.AllocSysString();
}
```

How to Create a User Exit

Before you begin

The ActiveX Automation Server must be installed on the same machine as the Gentran:Server translator (where you are using the translation object that contains the user exit).

Procedure

Use this procedure to create a user exit.

Step	Action
1	Start the Gentran:Server Application Integration subsystem.
2	Open the map in which you want to apply a user exit.
3	Determine where the user exit will be located in the map (e.g., session rule or extended rule). Note When you apply a user exit to a map component, subordinate map components may also be able to execute the same user exit.
4	Construct the user exit. Reference <ul style="list-style-type: none"> ▶ See <i>Defining Extended Rules</i> on page 6 - 12 for more information on creating extended rules for various map components. ▶ See <i>Alphabetic Language Reference</i> on page 6 - 47 for more information on the <i>createobject</i>, <i>deleteobject</i>, and <i>getiid</i> commands.

Using GentranEx.DLL

Contents	▶ Introduction	C - 2
	Database Access	C - 3
	▶ Overview	C - 3
	▶ Implementing Database Access	C - 4
	Debugging	C - 5
	▶ Overview	C - 5
	▶ Implementing Debugging	C - 6
	Rules Extensions	C - 8
	▶ Overview	C - 8
	▶ Implementing Rules Extensions	C - 9

Introduction

In this appendix

This appendix describes how to use GentranEx.DLL, the Gentran:Server in-process automation server.

Note

GentranEx.DLL is automatically installed during the Gentran:Server installation process.

What is GentranEx.DLL?

GentranEx.DLL is an in-process automation server that extends the flexibility of the Gentran:Server MAPPER.EXE (Application Integration) and TX32.EXE (translator) programs by using COM technology and the IDispatch interface.

GentranEx.DLL enables you to perform several functions that exceed the scope of normal Gentran:Server operation, including:

- accessing any GentranDatabase table,
 - gathering information from a translation session and writing that information to a log file, and
 - extending the functionality of Application Integration extended rules.
-

Assumptions

This list contains the items with which this appendix assumes you are familiar.

- Gentran:Server for Windows,
- the Application Integration subsystem,
- your operating system,
- automation servers,
- how and when extended rules are invoked and their scope, and
- how to create extended rules.

Reference

See *Using Extended Rules*, chapter 6, for more information on Application Integration extended rules.

Database Access

Overview

Introduction

GentranEx.DLL allow access to any GentranDatabase table through the TSql statements UPDATES and INSERTS. The GentranEx.DLL uses ODBC and the Gentran:Server ODBC DSN (typically GentranDatabase) to connect to the database.

Note

The TSql statement must be generated by the user and passed to the Execute method. To support a connection to an Oracle[®] database the TSql statements must enclose Tablenames and Columnames in square brackets. These square brackets are automatically converted to double quotes before the statement is executed.

Implementing Database Access

Prog ID "GentranEx.DispatchDBAccess"

Interface ID {860AAE85-7DA8-11D2-ABDB-00C04FF3971C}

Methods This table contains the database access methods.

Method	Description
short ExecuteSQL (BSTR IpszSQLString)	Returns "1" for success and "-1" for failure. Errors are written to the Application Event Viewer.
BSTR GetLastError ()	Returns a string containing the error of the most recent attempt to invoke the ExecuteSQL method. If the last attempt was successful this call returns a null string.
BSTR GetLastSQL ()	Returns a string containing the most recently executed SQL statement using the ExecuteSQL (...) function.

Properties None.

Extended rule example This is an example of an extended rule using GentranEx.DLL to access the GentranDatabase.

```
Object obDBAccess;
Integer result;
String [64] szPartnerKEY;
String [1024] szMsg;

obDBAccess = CreateObject ("GentranEx.DispatchDBAccess");
szTSQL = "Update [interchange_tb] Set [AppField6]=""+#STATUS+
"" where [PartnerKEY]=""+
szPartnerKEY+
"" and [Direction]=1 and [ControlNumber]=""+#REF+"";
result = obDBAccess.ExecuteSQL (szTSQL);
if result = -1 then
  Begin
    szMsg = obDBAccess.GetLastError ();
    messagebox (szMsg, 0);
    szMsg = obDBAccess.GetLastSQL ();
    messagebox (szMsg, 0);
  End

deleteobject (obDBAccess);
```

Debugging

Overview

Introduction

GentranEx.DLL enables you to gather information from any translation session executed on the machine on which GentranEx.DLL is installed, and write that information to a log file. This function does not require any interaction from the user and thus is an efficient alternative to using the MessageBox function when debugging map.

Each entry made to the log file contains the following information, which provides clarity and readability to the log:

- date and time stamp,
- name of the log owner,
- source of the message, and
- the message.

Note

The name of the log owner, the source of the message, and the message are all supplied by the user via the OpenLog() and WriteLog() functions.

Implementing Debugging

Prog ID "GentransEx.TraceLog"

Interface ID {7DA17F77-8090-11D2-ABE4-00C04FF3971C}

Methods This table contains the debugging methods.

Method	Description
short OpenLog (BSTR PathFilename, BSTR Owner, short WriteMode)	<ul style="list-style-type: none"> ▶ Returns "1" for success, "-1" for failure, and "-2" if the file is already open. ▶ Argument 1 is the path and filename of the log file to be created. ▶ Argument 2 is the owner of the log. This value is written to the log file and is included to make the log easier to read. ▶ Argument 3 is the writemode (i.e., 1 = append (this is the default) and 2 = truncate). ▶ The "Owner" column in the generated log file is 20 characters wide — therefore Argument 2 is truncated at that value (20) when written to the log file.
short Open ()	Opens a trace log file (named "GentransTrace.log") in the GENSRVNT home directory. The file is opened in Append mode and is assigned an Owner name of "GentransEx".
short WriteLog (BSTR Source, BSTR Message)	<ul style="list-style-type: none"> ▶ Returns "1" for success and "-1" for failure. ▶ Argument 1 is the source of the message that is written. ▶ Argument 2 is the message text. Both Arguments 1 and 2 are written in separate columns in the log file for clarity. ▶ The "Source" column in the log file is 20 characters wide and thus Argument 1 is truncated at that value (20). ▶ The "Message" column in the log file is 256 characters wide and thus Argument 2 of this function is truncated at that value when written to the log file. <p style="text-align: right; color: green;">(Continued on next page)</p>

(Contd) Method	Description
BSTR GetLastError ()	Returns a string containing the error of the most recent attempt to open the log file.
short CloseLog ()	Closes the log file and returns "1" for success or "-1" for failure. Note The file should always be closed before your trace object goes out of scope.

Properties

None.

Extended rule example

This is an example of an extended rule using GentransEx.DLL for debugging.

```

Object obTrace;
Integer result;
String [32] szPartnerKEY;

obTrace = CreateObject ("GentransEx.TraceLog");
result = obTrace.OpenLog("D:\GENSRVNT\TraceTest.txt","ICLANA",2);
if result = -1 then
  Begin
    obTrace.WriteLog("Start of Log", "Trace Test");
    obTrace.WriteLog("Converted TUN#",szPartnerKEY);
  End
obTrace.CloseLog();
deleteobject (obTrace);

```

Rules Extensions

Overview

Introduction

GentransEx.DLL provides methods and properties that extend the functionality of the Application Integration extended rules logic. For example, there are methods that allow the conversion of strings from uppercase to lowercase, which allows greater control over string comparison logic. Also, there are functions available that return the current date and/or time. GentransEx.DLL also provides a property that can be referenced to determine whether the automation server is running in a non-interactive environment.

Implementing Rules Extensions

Prog ID "GentranEx.Rules"

Interface ID {387C0F34-82F5-11D2-ABEA-00C04FF3971C}

Methods This table contains the rules extension methods.

Method	Description
DATE GetDate ()	Returns the current date. The assignment operator (=) can be used to set any date/time variable or date/time type field equal to the value returned by this method.
DATE GetTime ()	Returns the current time. The assignment operator (=) can be used to set any date/time variable or date/time type field equal to the value returned by this method.
BSTR ToUpper (BSTR szMixedCase)	<ul style="list-style-type: none"> ▶ Uses the string to be converted as its argument. ▶ Returns the original string converted to uppercase. Any string variable, string field, or string literal can be supplied as the first argument.
BSTR ToLower (BSTR szMixedCase)	<ul style="list-style-type: none"> ▶ Uses the string to be converted as its argument. ▶ Returns the original string converted to lowercase. Any string variable, string field, or string literal can be supplied as the first argument.

Properties short **InteractiveState**

**Extended rule
example**

This is an example of an extended rule using GentransEx.DLL to exceed normal extended rule logic.

```
Object obRules;
String [32] szLowerCaseStr;
String [32] szUpperCaseStr;
String [32] szResultString;
date d;

//Initialize values of string variables
szLowerCaseStr = "abcdefg";
szUpperCaseStr = "WXYZ";

//Create the Rules object
obRules = CreateObject("GentransEx.Rules");

//Test string conversion methods and display results
szResultString = obRules.ToUpper(szLowerCaseStr);
messagebox(szResultString,0);
szResultString = obRules.ToLower(szUpperCaseStr);
messagebox(szResultString,0);

//Test date methods
d = obRules.GetDate();
#TimeField = obRules.GetTime();

//Test whether running interactively
if obRules.InteractiveState = 1 then
    messagebox("We're running on the desktop",0);
else
    Begin
        //use the TraceLog object to write to a log file since we're
        //running non-interactively
    End

deleteobject(obRules);
```

Using the Translator Command Line Interface

Contents	▶ Introduction	D - 2
	▶ Command Line Syntax	D - 3

Introduction

In this appendix

This appendix describes how to use translator command line interface to invoke the services of TX32.EXE, including generating a translator report in XML format.

Notes

- If you do not specify the XML format parameter, the report is generated in binary form by default.
 - This interface is currently used by all applications that invoke the services of the TX32.EXE translator.
-

Command Line Syntax

Introduction The syntax of the command line is:

```
tx32 [-n] [-q]
      [-a server]
      [-e eventid]
      [-i ipckey [hwnd] ]
      [-f infile templatefile outfile report file]
      [-u param -u param ...]
      [-t tracefile]
      [-x XMLformattedreport]
```

Syntax parameters This table describes the command line syntax parameters.

Parameter	Description	
-n	Do not move document to InDrawer after print or export	
-q	Quiet mode (does not display message boxes or status dialog box)	
-a	Server (name of audit server to which TX32.EXE connects)	
-e	EventID (identifier of an audit event to which this translator belongs)	
-i	Inter-process communication (IPC) mode	
	Subparameter	Description
	ipckey	Key to ipcfile (translator will use the IpcMsg folder and add .IPC)
	hwnd	Number representation of the parent applications's main window handle
-f	Simple-translator mode	
	Subparameter	Description
	infile	Path to file to be translated
	templatefile	Path to translation object to be used
	outfile	Path to file to write translated output
	reportfile	Path to create translator report
-u	User-defined parameters accessible from extended rules	
	Subparameter	Description
	param	Parameter value

(Continued on next page)

Parameter	Description	
-t	Trace mode	
	Subparameter	Description
	tracefile	Path to trace the file to be used
-x	XMLformattedreport	Generates the translator report in XML format Note If you do not specify this parameter the report is generated in binary form by default.

Glossary

?In Documents	This Gentran:Server browser contains a list of documents that were received by the system but failed compliance checking or that do not have an identifiable partner or transaction set.
?Out Documents	This Gentran:Server browser contains a list of documents that were imported into the system via unattended processing but are invalid.
acknowledgement	This term is used to indicate the ANSI 997 functional acknowledgement, the EDIA 999 acceptance/rejection advice, and the EDIFACT CNTRL document.
activation	This function enables you to make map components available for use. The system activates all the groups, segments, composites, and elements that are defined as “mandatory” (must be present) by the standard. The system does not enable you to deactivate the mandatory groups, segments, composites, and elements. When translating data, the system does not process groups, segments, composites, and elements (or records and fields) that are not activated. Therefore, <i>you</i> must activate the groups, segments, composites, and elements that are not defined as mandatory by the standard, but that you have determined that you need to use in mapping.
AIAG	The Automotive Industry Action Group (AIAG) is the standards-setting group for the automotive industry. The standards form a subset of the ANSI X12 standard.
ANA	Article Numbering Association.
ANSI	American National Standards Institute. ANSI sets standards for many products and services, such as safety glasses and battery capacities. The ANSI X12 committee is the chief EDI standards-setting organization for the United States.
application file	If you are creating an import or export map, you must define your application to Gentran:Server. In Gentran:Server terminology, your application file is also referred to as a <i>fixed-format file</i> or a <i>positional file</i> . Your application file must contain all the information that you either need to extract from your partner’s document (if the map is inbound) or need to send to your partner (if the map is outbound), so that your system can accurately process the data.

application system	Computer systems, outside of EDI, designed to fulfill specific business functions. These include accounting, purchasing, materials control, human resources, shipping, and other systems.
browser	A browser is a window that displays a list of items, such as documents (transaction sets), interchanges, translation objects, etc. A document browser displays only documents.
button	A button enables you to execute a command. You select a button either by clicking it or by pressing TAB until the button you want is selected and then pressing ENTER . Unavailable buttons are dimmed, and the currently selected button has a darker border than the other buttons.
check box	A check box enables you to select or clear options from a list. You can select as many options from the list as you need. A selected check box contains a checkmark. The labels of unavailable options are dimmed.
click	Click indicates that you should position your cursor on top of the designated item and press and release your primary mouse button. The primary mouse button is the button you access with your index finger, regardless of whether you are utilizing right-handed or left-handed mouse operation.
code list tables	These are used by EDI standards as repositories for lists of codes. Each EDI standard provides a code list for each element that can be further defined with a code.
colours	This feature enables you to select foreground and background colours to visually define the various map components. The use of colour is optional.
communications session	Everything sent and received to/from one telephone number in one continuous period of connection. This could include sending two or three interchange envelopes to a network, each for a different trading partner.
Communicator	This is the Gentran:Server communications software. The Communicator enables you to send, receive, resend, establish communication port definitions, establish communication profiles, establish communication scripts, and view and delete communication sessions.
compile	This function compiles the map and creates a translation object. The map that you create using Gentran:Server is a <i>source map</i> . When that source map is compiled, the result is a <i>compiled translation object</i> . This translation object must be registered with the Gentran:Server system before you can use it.

compliant	This means that the document conforms to the EDI standards as defined by the translation object.
component data element	A simple data element that belongs to a composite data element. Component data elements are also called sub-elements.
composite data element	A contains two or more component data elements or sub-elements. Composites are defined by the EDI standards that use them (EDIFACT, TRADACOMS, and certain ANSI X12 standards).
conditions	See <i>relational condition</i> on page 1 - 9.
constant	This standard rule enables you to move a literal constant value to the specified element or field, indicate a qualifying relationship with another element or field, and map the current date or time to the specified element or field.
cross-reference table	This table is created in Partner Editor and is used to convert your values to your trading partner's values during outbound processing, or your partner's values to your values during inbound processing.
date/time	This type of field or element contains a date or time. If you specify that a field or element is a date/time type, you must specify exactly how the date or time must be formatted.
deactivation	This function enables you to make map components unavailable for use by the system. The system does not enable you to deactivate the mandatory groups, segments, composites, and elements.
default	A predefined value – programs use these built-in values <i>unless</i> you specifically override them.
dialog box	A dialog box contains additional information or options that you need to act on. If you specify any options, you can select a button to execute the command. Some dialog boxes display warnings or messages indicating why a requested task cannot be accomplished. If the dialog box has a title bar, you can move it to another location on your desktop.

document One transaction set containing actual data and treated as a single entity. The amount of data does not affect whether something is a document or not, only the fact that it is treated as a single entity. For example, regardless of whether a purchase order contains one item or ten thousand, if it is one purchase order, it is one document.

If a trading partner sends you a communication containing 10 purchase orders, you have received 10 documents. If the communication contained 15 invoices, you have received 15 documents.

EANA International Article Numbering Association, Brussels, Belgium.

EDI Electronic Data Interchange (EDI) is the process by which companies can exchange business documents directly from application to application by computer without paper documents being produced.

EDI standard This is the rules for turning a business document into an EDI document.

EDIA The Electronic Data Interchange Association, formerly known as the Transportation Data Coordinating Committee (TDCC).

EDIFACT This is the standards organization of the United Nations Economic and Social Council. The acronym is short for the Electronic Data Interchange for Administration, Commerce, and Transport.

element The smallest piece of usable information defined by the standards. Examples might include a quantity, unit price, or description. An individual element can have somewhat different meanings depending on context. Therefore, elements are normally not considered to have useful meaning until they are combined into segments. An element is the EDI map component that is mapped (linked) to a corresponding application field to move data to and from the EDI file.

There are three types of data elements, as illustrated in the table below:

Data Element	Definition
Simple data element	A single piece of information defined by the standards.
Composite data element	A data element that is made up of multiple component elements.
Component data element	A simple data element that belongs to a composite data element. Component data elements are also called sub-elements.

envelope	<p>A way of separating information in transmissions for ease of processing. Each envelope contains a header segment and a trailer segment, which separate the envelope from other envelopes and provide information about the contents of the envelope.</p> <p>There are three levels of envelopes, as follows:</p> <ul style="list-style-type: none">• Transaction Set Each transaction set (business document) is contained within a transaction set envelope.• Functional Group An envelope containing related business documents. The standards define which transaction sets should be placed together into a functional group envelope.• Interchange Envelope All material being sent to one trading partner in one communication. The term Interchange Envelope is the term used by ANSI. EDIA uses the term Transmission Envelope to refer to this level of envelope. Since we use the term transmission for other uses, we refer to Interchange Envelope only.
equalize	<p>This function enables you to reinstate the two sides of the map with focus in equal dimensions. The use of the Equalize function is optional.</p>
export map	<p>This map defines how to move data from the EDI standard-formatted documents that your partners send you to your application file (flat file definition). An export map is necessary for inbound processing.</p>
export translation object	<p>See <i>export map</i> on page 1 - 5.</p>
extended rules	<p>These rules enable you to use a Gentran:Server proprietary programming language to perform virtually any mapping operation you require.</p>
field	<p>The smallest piece of information defined in the application file. A field is the application map component that is mapped (linked) to a corresponding EDI element to move data to and from your application file.</p>
fixed-format file	<p>See <i>application file</i> on page 1 - 1.</p>
font	<p>This feature enables you to globally change the font that is used in the display of all maps. The default font that Gentran:Server uses is a Sans Serif 9 point. The Font feature enables you to change the font type, style, and point size. This gives you the flexibility to shrink the font if you need to view more of the map on your monitor, enlarge the font, or change the type and style to be more easily readable by you.</p>

functional group	A group of transaction sets that the standards-setting body (such as ANSI) has defined as fitting together with other related transaction sets. An example might be a Functional Group defined as containing all purchasing transaction sets.
group	A looping structure that contains related records/segments and/or groups that repeat in sequence until either the group data ends or the maximum number of times that the loop is allowed to repeat is exhausted. If you create a group that is subordinate to another group (a sub-group), this corresponds to a nested looping structure (a loop within a loop). The application (positional) file and the EDI file are both groups and therefore, they are visually represented the same way as other groups and sub-groups in Gentran:Server.
import	This command enables you to import data from an external application file. Depending on the content of the file, the system may prompt you for partner, transaction, or translation object information.
import map	This map defines how to move data from your application file (flat file definition), which may contain multiple documents, to the EDI standard-formatted documents that your partners expect to receive from you. An import map is necessary for outbound processing.
import translation object	See <i>import map</i> on page 1 - 6.
inbound mapping	The system translates your trading partner's EDI standard formatted business document to your application file format, so you can receive documents from your partners. To translate inbound data, you need to create an export map.
In Documents	This Gentran:Server browser contains a list of documents that the system received but that have not yet been processed by the user. Once the documents are processed (printed or exported) the documents are transferred to the In Drawer.
In Drawer	This Gentran:Server browser contains a list of documents that were received and processed.
integer	An integer is a number that has an implied decimal point (e.g., "2.01" is formatted as "201").
interchange	The interchange contains all the functional groups of documents (transaction sets) sent from one sender to one receiver in the same transmission.

Interchanges browser	This Gentran:Server browser enables you to view a hierarchical list of all the interchanges that were sent or received. It contains status information about the interchanges, (e.g., whether or not a functional acknowledgement was received and the status of that acknowledgement). This browser also enables you to access the documents within the interchanges. The Interchanges browser can always be accessed.
link	This function enables you to map a field or element from the Input side of the map to a field or element on the Output side of the map. The link between two map components (hereafter referred to as fields) is visually represented with a line connecting the two fields.
literal constant	These are used by the system as a repository to store information that is used at a later point in the map. Typically, constants are used in an outbound map to generate a qualifier.
location table	This is a table created in Partner Editor that is used to contain address-related information about the partner.
lookup table	This is a table created in Partner Editor that is used to select information related to a value in inbound or outbound data.
loop count	This standard rule enables you to count the number of times a loop is repeated, if the element or field is part of a loop. If the loop is a nested loop, you can track the current loop <i>or</i> the outer loop. For example, if the Y loop is nested within the X loop, and the Y loop has cycled through 15 iterations and the X loop has cycled through 3 iterations, you can choose to count either the “15” (Y loop) or the “3” (X loop).
loop start/loop end	Certain EDI standards use Loop Start (LS) and Loop End (LE) segments. LS and LE segments differentiate between two or more loops of the same type. If the transaction contains LS and LE segments and depending on whether your map is inbound or outbound, you need to define the LS and LE segments for the loops you are using in the map in one of two different ways.
map	A set of instructions you define in the Gentran:Server Application Integration subsystem that indicates the corresponding relationship between your application file and the EDI standards, and defines how the system should translate data.
menu	A list of items from which you can choose. Each item represents a command. Unavailable items on a menu are dimmed to inform you they are disabled.

menu bar	This contains the list of menus for the application.
network	Also known as a Third Party Network or Value-Added Network (VAN) that accepts and holds transmissions from companies until it is convenient for a trading partner to accept them.
number	This type of field or element contains either an integer or real number. If you specify that a field or element is a number type, you must designate the format by specifying a format of either “N” (integer) or “R” (real) and the number of decimal places.
ODETTE	The Organization for Data Exchange by Tele-Transmission in Europe.
outbound mapping	The system translates your application file format to EDI standard formats, so you can send documents to your partners. To translate outbound data, you need to create an import map and a system import map in Gentran:Server.
Out Documents	This Gentran:Server browser contains a list of documents that are ready to be sent. After the documents are successfully sent, they are automatically transferred to the Out Drawer.
Out Drawer	This Gentran:Server browser contains a list of documents that were successfully sent out by the system.
partner	Another firm with which your company trades documents. Also referred to as a trading partner.
Partner Editor	This Gentran:Server function enables you to define, edit, and delete all partner information for your company and all of your trading partners.
positional file	See <i>application file</i> on page 1 - 1.
post	This Gentran:Server function enables you to move compliant documents from the Workspace to the Out Documents.
predefined	On a data entry translation object, a default value for a particular element. You can change the value as necessary.

printout	This is produced when data received from a trading partner passes through a print translation object. You do not have to actually print the data. The printout could be to a file on your hard disk.
promote	This function extracts one iteration (instance) of a group or repeating segment. This enables you to map unique data from your application file, and/or enter a specialized definition. Gentran:Server specifies that only one-to-one (no loops) or many-to-many (loop) mapping relationships are valid.
qualifier	This is an element that has a value expressed as a code that gives a specific meaning to the function of another element. A qualifying relationship is the interaction between an element and its qualifier. The function of the element changes depending on which code the qualifier contains.
real number	A real number has an explicit decimal point (e.g., “2.01” is formatted as “2.01”), and truncates trailing zeroes.
receive	This Gentran:Server function enables you to manually initiate a communications session to receive data from your trading partner.
record	Contains a group of related fields. A record can occur once or can repeat multiple times.
relational condition	This function enables to connect fields together for syntax or compliance reasons. For example, Field A is invalid unless Field B is present. Therefore, if you set up a condition that pairs Fields A and B, the system generates a compliance error if one of those fields is not present.
respond	This Gentran:Server function enables you to create a turnaround document (using the Turn Around translation object in the partner relationship) in response to one or more selected documents in the In Drawer.
responsible agency	An organization that develops and updates standards for EDI communications. These organizations include ANSI, EDIA, AIAG, UCS, EDIFACT, ODETTE, and VICS.

-
- segment** A group of related elements or composite data elements that combine to communicate useful data. Segments are defined by the EDI standards. A segment can occur once or can repeat multiple times. For example, a catalog price segment might consist of elements for item description, volume, price, lead time, etc. By themselves, none of these elements would communicate useful information. Together, they provide the information necessary for someone to tell if the item is what is desired, whether the price is reasonable, etc. A number of segments together form a transaction set.
-
- select** This standard rule enables you to select entries from a location table, cross-reference table, partner table, or lookup table (all tables created in the Gentran:Server Partner Editor). You can then map the fields in those tables to one or more fields in the data. The Select function uses the value of the current field to perform the selection.
-
- send** This Gentran:Server function enables you to manually start a communications session to send data to your trading partner. Only selected documents are enveloped and sent. If no documents are selected, ALL documents are sent. Successfully sent documents are moved to the Out Drawer.
-
- set** See *transaction set (document)* on page 1 - 12.
-
- simple mapping** See *link* on page 1 - 7.
-
- split** This function enables you to split (break) a group or repeating segment into two loops. You typically use this function when you need more than one instance of the same map component that still occurs multiple times.
-
- standard rules** These rules give you access to functions that are necessary for mapping operations that are more complex than simple linking, but less involved than extended rules. Each of the standard rules are mutually exclusive (you can use only one on a particular field).
-
- status bar** The status bar of an application window defines information about a selection, command, or process, defines Menu Bar items as the user highlights each item, and indicates any current keyboard-initiated modes for typing (e.g., CAP for the “Caps Lock” key or NUM for the “Num Lock” key, etc.).
-
- string** This type of field or element contains one or more printable characters. If you specify that a field or element is a string type, you must designate the format by specifying a syntax token.
-

syntax token	This function enables you to designate a “token” that defines ranges of characters and/or numbers that are allowed to be used for a string-type element or field. You can then use the syntax tokens in the Format field of the Field Properties dialog box. This enables you to define what type of characters should be used while compliance checking each element/field (i.e., alphanumeric within a certain range, numeric within a certain range, etc.).
system import map	The translator uses this map to determine which trading relationship (established in Partner Editor) corresponds to each document in the application file, so the system knows which import map to use to process the document. You need a system import map that is defined in EDIMGR.INI to translate outbound data.
system import translation object	See <i>system import map</i> on page 1 - 11.
system translation object	These translation objects control the creation and separation of interchanges, functional groups, and transaction sets. They also used to generate and reconcile functional acknowledgements. All the required system translation objects are automatically installed with the Gentran:Server system.
TDCC	See <i>EDIA</i> on page 1 - 4.
TDF	TDF is the Transaction Data File. This file serves as a filter between your document files and the Gentran:Server translator. For outbound processing, data is imported from a TDF file and translated to EDI format using a TDF Import translation object. The data is then ready to be posted and sent to your trading partner. For inbound processing, EDI documents that are received by Gentran:Server can be exported to a TDF-formatted file using a TDF Export translation object. This data file is then ready to be processed or converted to your internal application files.
third-party network	See <i>network</i> on page 1 - 8.
title bar	The title bar of an application window contains the name of the application and enables you to move the window to another position on your desktop.
toolbar	This contains buttons that graphically represent commands available in the Menu Bar.

TRADACOMS	The U. K. standards for EDI that are published by the Article Numbering Association (UK) LTD.
trading partner	Another firm with which you company trades documents. Also referred to as a partner.
transaction set (document)	A business form as defined by the standards. Examples include an ANSI 850 purchase order or an UCS 880 invoice. The standards define each transaction set in terms of the segments and elements that make up the form, the order in which they appear, and the relationships among them. This is also known as “message” in Europe.
translation object	<p>A predesigned layout set up to ensure that input or output for a particular transaction set exists and is presented in a usable fashion. You must specify which translation objects are used by each partner relationship.</p> <p>Inbound Translation Objects:</p> <p>Turn Around: This translation object is used when a document is received, to create the natural response document that contains as many elements from the received document as possible.</p> <p>Export File: This translation object indicates that when a document is received, it is exported to a specified file format.</p> <p>Print: This translation object is used to print documents.</p> <p>Outbound Translation Objects:</p> <p>Import: This translation object is used to import data from an application file.</p> <p>Print: This translation object is used to print documents.</p> <p>Data Entry: This translation object is used to enter data into the Document Editor facility.</p>
translator	This is the engine that processes data for Gentran:Server, the Application Integration subsystem, and the Forms Integration subsystem.
transmission	See <i>communications session</i> on page 1 - 2.
transmission chain	A path an EDI communication could follow, including one company, one trading partner, and one or more network services.
turnaround document	A document into which data elements from the source document have been automatically transferred using a turnaround map.

-
- turnaround map** A series of instructions that the system uses to create a turnaround document (a logical response document to the source) from an inbound (source) document, by transferring data from the source document to elements in the target document (translation object).
-
- UCS** The Uniform Communications Standard is the standard used by the grocery industry.
-
- update** This standard rule enables you to update a specific field in a document record, envelope segment, interchange, group, current partner, or document (if the map side format type is EDI), with the contents of the element or field.
- Warning**
This function updates the internal Gentran:Server database tables. We recommend that you use this function only if you are sure you really want to update the internal database tables.
- Typically, you *only* use this function if you want to update the document name and reference in the document table. Any other use of this function could have disastrous consequences!
-
- use accum** This standard rule gives you access to a set of numeric variables that you can manipulate via numeric operations, and then transfer to and from fields. This function enables you to add, change, or delete calculations for the element/field, including hash totals (used to accumulate numeric field values, i.e., quantity, price, etc.). This function also enables you to map the accumulated total into a control total field, and use accumulators. Accumulators are used generally for counting the occurrences of a specific element or generating increasing or sequential record or line item numbers.
-
- use code** This standard rule enables you to match an element or field against a predefined code table, specify whether or not a compliance error is generated if the element or field does not contain one of the values in the code table, and store a code's description in another element or field.
-
- user translation object** These translation objects control data entry, importing, exporting, document turnaround, and creating printed reports. These translation objects are created using the Application Integration or Forms Integration subsystems. The Application Integration subsystem enables you to generate import, export, and document turnaround translation objects. The Forms Integration subsystem enables you to generate data entry and print translation objects.
-
- version** Each standards-setting body updates its standards on a regular basis. Each formal update is referred to as a version.
-

VICS The Voluntary Inter-industry Communication Standards is the standards-setting body for the retail industry, a subset of ANSI X12.

Workspace This Gentran:Server browser contains a list of outbound “Work in Progress” documents. It also contains recently imported or data entry documents.

X12 The ANSI committee that sets and publishes standards for EDI.
