

Gentran:Director[®]

for Windows[®]

Version 5.4

Forms Integration

User's Guide

Sterling Commerce
An IBM Company

April 2005

© Copyright 1998–2005
Sterling Commerce, Inc.
ALL RIGHTS RESERVED

WARNING: ANY UNAUTHORIZED DUPLICATION OF THIS DOCUMENTATION SHALL BE AN INFRINGEMENT OF COPYRIGHT

Trade Secret Notice

This documentation, the 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 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 FAR 52.227-19. Further, as and when provided to any governmental entity, government contractor or subcontractor subject to DFARs, this documentation and the software it describes are provided pursuant to the customary Sterling Commerce license, as described in DFAR 227.7202 with respect to commercial software and commercial software documentation.

This documentation and the software which it describes are licensed either "AS IS" or with a limited warranty, as set forth in the applicable 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.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Gentran:Director, Commerce:Connection, and Commerce:Network are trademarks or registered trade marks of Sterling Commerce. Windows, Windows NT, ODBC, SQL Server, Microsoft, MS-DOS, and Access are trademarks or registered trademarks of Microsoft Corporation. CLEO is a proprietary product of CLEO Communications, a division of Interface Systems, Inc. HP and LaserJet are registered trademarks of Hewlett-Packard Company. Pentium is a trademark of Intel Corporation.

Contents

Chapter 1	Getting Started	1
	What's in this Manual	1
	Conventions in this Manual	3
	Getting Support	4
	Introducing Forms Integration	5
	EDI File Format Window	5
	Layout Window	7
	Introducing the Forms Integration Subsystem	8
	Navigation	8
	Keyboard Techniques	11
	Building a Form with Forms Integration	12
Chapter 2	Form Integration Basics	15
	Creating a New Form	16
	Using an EDI File Format	20
	Verifying EDI Delimiters	20
	Saving a File Definition	22
	Creating a Group	23
	Modifying Group Properties	25
	Modifying Segment Properties	26
	Modifying Composite Properties	28
	Modifying Element Properties	29
	Activating EDI Form Components	30
	Promoting Groups and Repeating Segments	32
	Splitting Groups and Repeating Segments	32
	Using Copy, Cut, and Paste	33
	Opening a Form	38
	Saving a Form	39

Naming a Form	41
Compiling a Translation Object	42
Printing a Report	44
Formatting Your Data	45
Using the String Type	45
Using the Number Type	54
Using the Dt/Tm Type	57
Customizing Your Display	60
Customizing Global Display Options	60
Customizing Global Colours	61
Customizing Global Fonts	63
Customizing Display of EDI File Format and Layout Windows	64
Setting Default Date Format	65
Selecting Which Form Components to Use	67
Cutting, Copying, and Pasting	69
Setting up the Document Name and Reference	71
Chapter 3 Customizing the Properties of Form Components	75
Customizing a Segment or Group	76
Customizing Group Properties	76
Customizing Segment Properties	77
Using Loop Start and Loop End Segments	80
Defining a Loop Start	80
Defining a Loop End	83
Customizing an Element	84
Defining and Modifying Relational Conditions	85
Customizing a Composite Element	88
Chapter 4 Using Standard Rules	89
Using the Select Standard Rule	90
Using the Update Standard Rule	94
Example	95

Using the Constant Standard Rule	98
Using System Variables	98
Using a Constant	100
Example	101
Using the Use Accum Standard Rule	105
Using the Loop Count Standard Rule	107
Using the Use Code Standard Rule	109
Example	111
Using the System Date and Time	113
Using Literal Constants and Qualifying Relationships	114
Defining Literal Constants	115
Editing Literal Constants	116
Deleting Literal Constants	117
Using Literal Constants	118
Generating Qualifiers	119
Using Partner-Related Information in Forms	121
Using Information from the Partner Definition	121
Using Information from Location Tables	122
Using Information from Lookup Tables	123
Using Information from Cross-Reference Tables	124
Using Accumulators	126
Counting Line Items	126
Calculating Hash Totals	133
Resetting and Calculating a Value Total	137
Using Code List Tables	145
Defining a Code List	146
Modifying a Code List	148
Deleting a Code List or Code List Entry	150
Importing a Code List	153
Exporting a Code List	155
Loading a Code List Table from the Standard	157
Copying and Pasting Code Lists	160
Validating Data Against Code List Tables	162
Loading Code Item Descriptions	163

Chapter 5 Using Extended Rules 165

Introducing Extended Rules 166

Declarations Section 166

Statements Section 167

When Rules are Processed 168

Defining Extended Rules 170

Extended Rule Syntax 174

Keywords and Commands 174

Operators 175

Symbols 175

Common Statements and Examples 177

Alphabetic Language Reference 185

atoi 185

aton 186

begin 186

break 187

concat 187

continue 188

count 188

delete 188

empty 189

end 189

exist 189

get 190

if...then...else 190

index 191

left 192

len 192

messagebox 193

mid 194

ntoa 194

right 195

select 195

set 196

strdate 196

strstr	197
update	198
while...do	198
winexec.	199
Select and Update Available Options.	200

Chapter 6 Customizing Screen Entry Forms 205

Customizing Field Labels.	206
Setting an Initial Value for a Field	208
Using a Constant Value for a Field	210
Creating Definitions for a List Box	212
Creating Help Text for Fields and List Boxes	214
Creating Help Text for Fields.	214
Creating Help Text for List Boxes.	216
Creating Help Text for Frames	217
Formatting Entry Fields.	218
Formatting Display-Only (Non-Editable) Fields	220
Hiding Fields	221
Formatting Numbers	223
Formatting Dates and Times	226
Formatting Strings	229
Storing EDI Code Value Descriptions.	230

Chapter 7 Customizing Print Forms 233

Customizing Field Labels.	234
Hiding Fields	235
Formatting Numbers	236
Formatting Dates and Times	239
Formatting Strings	242

Chapter 8 Formatting Screen Entry and Print Forms 243

Selecting and Grouping Fields.	244
Selecting One or More Fields or Field Labels	244
Grouping Fields and Field Labels	246

- Resizing Fields 247**
 - Using Size to Length 247
 - Resizing a Field Manually 248
 - Resizing Groups of Fields Manually 248
- Arranging Fields 249**
 - Changing Grid Settings 249
 - Using Alignment Lines 251
 - Moving Fields and Field Labels 251
 - Aligning Fields and Field Labels. 252
 - Spacing Fields and Field Labels 253
- Resizing and Positioning a Frame 254**
- Resizing Drop-Down Combo Boxes 255**
- Setting Tab Sequences. 257**
- Adding and Positioning Static Text 258**
- Modifying Frame Titles. 259**
- Modifying List Box Names 260**
- Adding Column Headings to List Boxes 261**

- Chapter 9 Error Messages 263**
 - Compile Error Messages. 264
 - Gentran:Director Error Messages. 271

- Glossary 277**

- Index 291**

Getting Started

The Gentran:Director[®] for Windows[®] Forms Integration subsystem provides you with the tools you need to design forms to facilitate the keying and printing of EDI documents.

What's in this Manual

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



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.**
- **Using the Windows Program Manager.**

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

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

- *Chapter 1, Getting Started* explains the content, organization, and conventions in this guide, as well as the system requirements. This chapter also describes how to get technical support, provides a brief

outline of how to build forms using Gentran:Director, and gives you an introduction to the Forms Integration subsystem.

- *Chapter 2, Forms Basics* explains how to perform basic functions in Gentran:Director, including creating a new form; using EDI files; opening, saving, and naming a form; compiling a translation object, and printing a report. This chapter also explains customizing display options; setting the default date format; selecting which form components to use; cutting, copying, and pasting; setting up the document name and reference; and formatting your data.
- *Chapter 3, Customizing the Properties of Form Components* explains how to customize the properties of segments, looping structures, elements, and composite elements. 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 Gentran:Director standard rules: select, update, constant, use accum, loop count, and use code. This chapter also illustrates examples of when you might use certain standard rules.
- *Chapter 5, Using Extended Rules* defines what extended rules are and when they are used, and explains the proprietary programming language syntax.
- *Chapter 6, Customizing Screen Entry Translation Objects* explains how to customize screen entry translation objects, including customizing field labels, setting an initial value for a field, setting a constant value for a field, creating definitions for a list, and creating Help text for fields and lists. This chapter also defines how to format the various field types, how to hide fields, and how to store EDI code value descriptions.
- *Chapter 7, Customizing Print Translation Objects* explains how to customize print translation objects in Gentran:Director, including customizing field labels, hiding fields, and formatting the various field types.
- *Chapter 8, Formatting Screen Entry and Print Translation Objects* explains how to format the organization and size of the fields in the translation object.
- *Chapter 9, Error Messages* describes Gentran:Director error messages and specifies corrective actions.
- The *Glossary* contains definitions of Forms Integration and Gentran:Director terms and concepts.
- The *Index* provides you with a list of terms, concepts, functions, and processes to enable you to quickly find them in this guide.

Conventions in this Manual

The different typefaces used in this manual are as follows:

<i>Creating a new form</i>	This typeface is used for emphasis or to indicate a chapter or section in this manual or other Gentran:Director manuals.
NEW...	This typeface indicates a button in the program. Click the button with your mouse, or press TAB until the button is highlighted, then press ENTER to execute that command.
F1	This typeface indicates a key that you need to press on your keyboard. A plus sign (+) between key names indicates a combination of keys to be pressed together (e.g., ALT + TAB).
<i>Click this button</i>	This typeface highlights an action that you should take or indicates descriptive text. It is most often used for callouts (text that is “pulled out” to the side or bottom of an illustration) in this manual to highlight specific steps that you need to take to complete a task. In the callouts, any button that you need to click is described in this same typeface.

The different icons used in this manual are as follows:



This icon indicates a note that gives you additional information on the current subject.



This icon indicates a caution that alerts you to the potential loss of information. Please read any text preceded by this icon carefully.



This icon indicates helpful tips, suggestions, shortcuts, and alternative actions.

Getting Support

Sterling Commerce's Gentran:Director software is supported by trained product support personnel who are available to help you with product questions or concerns. 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.
- Make copies of the tables in the following sections so that you can enter the requested information and have it available when you call us.

For assistance, please refer to your *Getting Support* card to determine which support phone number you should use.

When you call product support, you should be prepared to provide us with the information below.

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.
 1. **System Configuration:**
Information about the primary system controller and all machines experiencing problems, including the computer manufacturer and model number, Windows version, amount of memory, available disk space, and the network adapter type.
 2. **Error Messages:**
Record the exact wording of any error messages you receive and the point in the software where the error occurred.
 3. **Attempted Solutions:**
Record any steps that you took attempting to resolve the problem and note all the outcomes.

Introducing Forms Integration

A form is a set of instructions you define in the Forms Integration subsystem that indicates how the system should format data. When you compile the form, Gentran:Director generates a translation object. After you register the translation object with Gentran:Director and associate the translation object with a partner relationship, the system uses the translation object to format EDI documents. The Forms Integration subsystem enables you to create print and screen entry translation objects.

A *print translation object* organizes and formats the printout of EDI documents that are received from or sent to the trading partners for which you have established a trading relationship (inbound or outbound) that utilizes that print translation object. The print translation object enables you to view the EDI document in an easily readable format.

A *screen entry translation object* provides a standardized format for keying an EDI document into the Document Editor, for translation and transmission to your trading partners, for which you have established an outbound trading relationship that utilizes that screen entry translation object. The screen entry translation object ensures that your users key all the data necessary to create the required EDI document.

Regardless of whether the translation object is print or screen entry, Gentran:Director generates the actual format of the translation object based on the EDI file format (in the EDI File Format Window), and displays it in the Layout Window.

EDI File Format Window





The EDI File Format Window (left side of the Forms Integration subsystem main window) contains the EDI file format. The EDI file must contain all the information that you need to include to either create a document using the Document Editor (screen entry translation object) or print documents (print translation object), so that your system can accurately process the data.

Gentran:Director 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 *Creating a New Form* on page 16 for more information on generating an EDI file when you create a new form. You typically need to customize the system-generated EDI file by modifying the properties of the form components, and using specialized Gentran:Director functions to manipulate the EDI file structure.

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

Regardless of whether the system generates the EDI file or you define it, the specific EDI form components that you use depend on the type of form 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 form components you are using before generating or defining an EDI file.

The form components that Gentran:Director uses to define the EDI file format are as follows:

- 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. Groups are defined by the EDI standards. A group that is subordinate to another group is a sub-group (this corresponds to a nested looping structure – a loop within a loop). The EDI file is a group and therefore, it is visually represented the same way as other groups and sub-groups in Gentran:Director: 
- 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 segment is visually represented in Gentran:Director as the following: 
- A **composite** is a data element that 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). A composite is visually represented in Gentran:Director as the following: 
- An **element** is the smallest piece of information defined by the EDI standards. An individual element can have different meaning depending on the context. Therefore, elements are normally not considered to have useful meaning until they are combined into segments. An element is visually represented in Gentran:Director as the following: 

When an element has an operation performed against it, a red checkmark is displayed over the element icon.

Please see *Using an EDI File Format* on page 20 for more information on customizing the properties of the form components, and using specialized `Gentran:Director` functions to manipulate the EDI file structure.

Layout Window

The Layout Window (right side of the Forms Integration subsystem main window) contains the actual format of the translation object.

`Gentran:Director` uses *frames* to format the EDI file for the translation object. Each frame contains the groups, repeating segments, and elements at that level (single segments are not represented on the frame).

For screen entry translation objects, groups (with a maximum usage greater than one) and repeating segments are formatted as lists. Elements are formatted as edit boxes (fields) or lists on the frame. Elements typically are formatted as fields, but are formatted as a list if you apply a standard rule that allows a selection of multiple items. Each group and repeating segment also has a corresponding frame that contains all the groups (with a maximum usage greater than one), repeating segments, and elements at that level.

For print translation objects, groups, repeating segments, and elements are always formatted as fields. Each group and repeating segment also has a corresponding frame that contains all the groups (with a maximum usage greater than one), repeating segments, and elements at that level.

When you click a segment or group in the EDI File Format Window, the corresponding frame is displayed in the Layout Window. When you click an element in the EDI File Format Window, the corresponding layout component is highlighted in the Layout Window.

Introducing the Forms Integration Subsystem

The Gentran:Director Forms Integration subsystem enables you to design forms to facilitate the keying (screen entry) and printing (print) of EDI documents that you receive inbound or send outbound.

Navigation

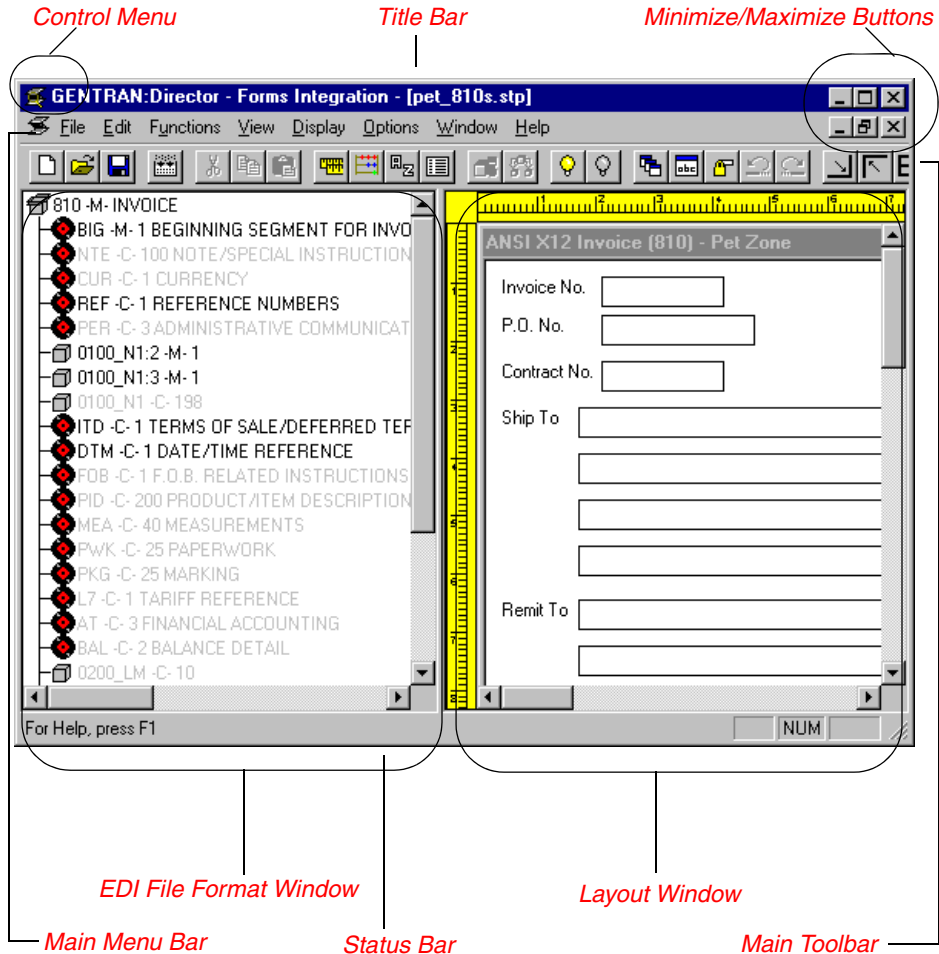
The Gentran:Director Forms Integration Window allows you to access its functionality in four different ways, as follows:

- Click the menu option from the Main Menu Bar.
- Click the appropriate button on the Main Toolbar.
- Click or double-click the appropriate part of the form.
- Click a form component (EDI File Format Window or Layout Window) with the right mouse button to access a shortcut menu that contains all the functions that are currently available for that form component. The shortcut menus enable you to quickly and easily access available functionality. The content of the shortcut menus vary, depending on the type and level of the selected form component.



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

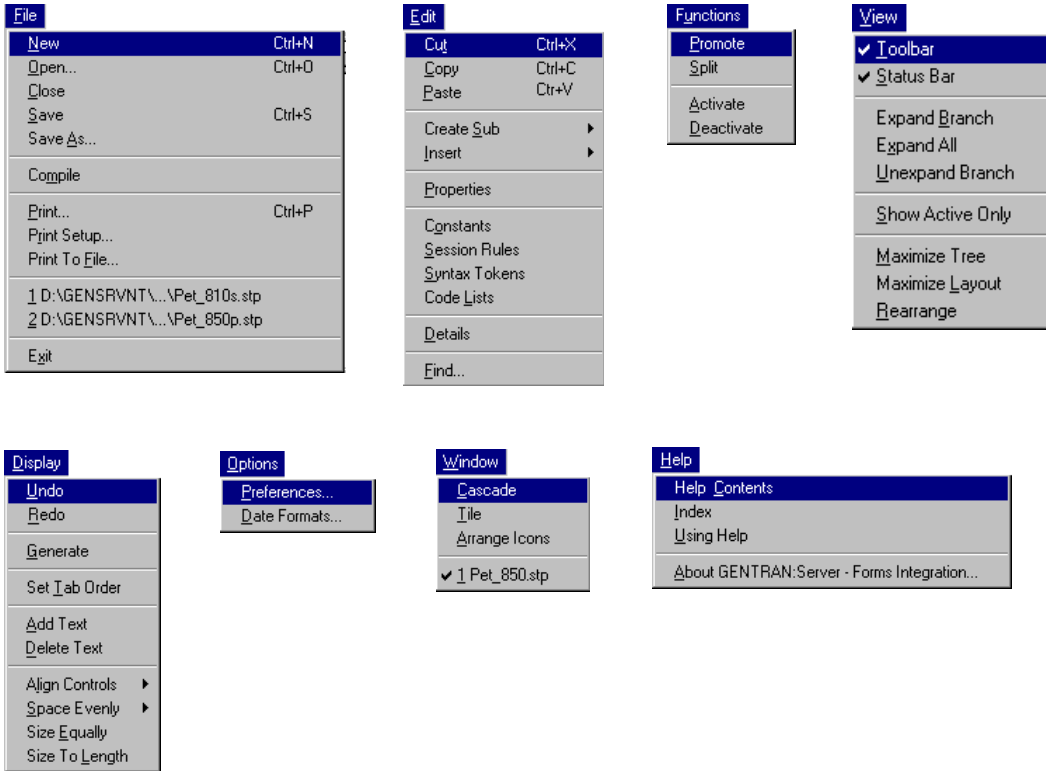
The Gentran:Director Forms Integration Window appears as follows:



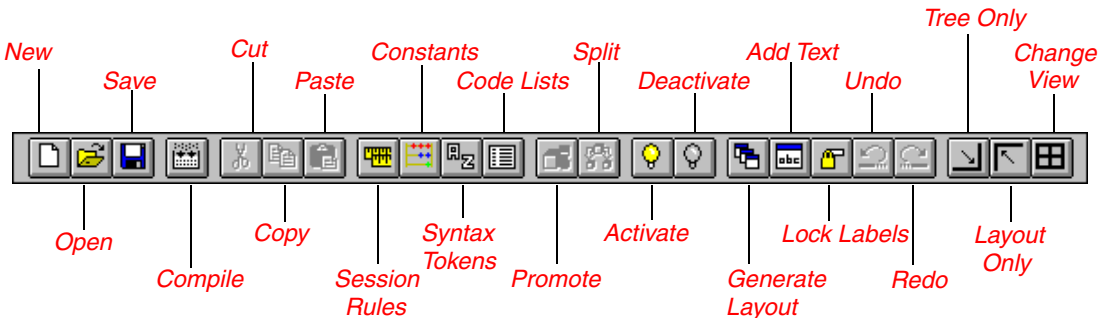
- The **Control Menu** contains the following commands: **Restore**, **Move**, **Size**, **Minimize**, **Maximize**, and **Close**.
- The **Title Bar** contains the name of the application. You can use this to move the position of the Forms Integration window on your desktop.
- The **Minimize** icon reduces the window to an icon.

The **Maximize** icon enlarges the window to its maximum size.

- The **Main Menu Bar** contains the following drop-down menus (detailed descriptions of each of the commands on these menus are found in the online Help). Unavailable items are dimmed.



- The **Main Toolbar** gives you access to some of the most common operations in Gentrans:Director. Unavailable items are dimmed. The Main Toolbar is a dockable toolbar, so you can affix it to any edge of the client window. The Main Toolbar is illustrated below:



- The **EDI File Format Window** contains the EDI file format. When you click a segment or group in the EDI file format, the corresponding frame is displayed in the Layout Window. When you click an element in the EDI file format, the corresponding layout component is highlighted in the Layout Window.
- The **Layout Window** contains the actual format of the translation object. Each group (with a maximum usage greater than one) and repeating segment has a corresponding frame in the Layout Window. When you click a segment or group in the EDI file format, the corresponding frame is displayed in the Layout Window. When you click an element in the EDI file format, the corresponding layout component is highlighted in the Layout Window.
- The **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 (e.g., CAP for the “Caps Lock” key, NUM for the “Num Lock” key, etc.).

Keyboard Techniques



The procedures in this manual are designed to be executed with a mouse. If you prefer to use the keyboard instead of the mouse to click a desired menu option, you may also press the **ALT** key and the underlined letter of the desired menu option to open the menu. Once the menu is open, press the underlined letter of the desired menu option to access that option.



If a key combination (e.g., CTRL + N for the NEW command) is specified on the menu for that option, you may press that key combination as well.

Instead of clicking a button to execute a command on a dialog or browser, you can press TAB to highlight the desired button and press ENTER to execute that command.

Building a Form with Forms Integration

This section defines how to proceed with building a form using the Gentran:Director Forms Integration subsystem.

1. **Prepare and Analyze**

Obtain a layout of how you want the document displayed and determine how it corresponds with the EDI standard you are using. Determine which operations need to be performed to generate the required layout.

2. **Set Global Defaults (first time only)**

The first time you use the Gentran:Director Forms Integration subsystem, you should establish the default date format that the system will use. Please see *Setting Default Date Format* on page 65 for more information on setting the global date default.

3. **Create, Save, and Name a New Form**

Please see *Creating a New Form* on page 16 for more information on creating, saving, and naming a form.

4. **Activate the Appropriate EDI Groups, Segments, and Elements**

Please see *Selecting Which Form Components to Use* on page 67 for more information on activation.

5. **Define the Layout of the Form**

Please see *Using an EDI File Format* on page 20 for more information on customizing your EDI file, including setting the EDI delimiters. Please see Chapter 4, Using Standard Rules for more information on using standard rules. Please see Chapter 6, Using Extended Rules for more information on using extended rules.

6. **Compile the Translation Object**

Please see *Compiling a Translation Object* on page 42 for more information on compiling the translation object and translation object naming conventions.

7. **Print the Report**

Validate and review the form, and make modifications as needed. Please see *Printing a Report* on page 44 for more information on printing the report.

8. **Register the Translation Object with Gentran:Director**

Please see the *Gentran:Director User's Guide* for more information on registering translation objects.

9. **Create the Appropriate Trading Relationship**

Establish the appropriate trading relationship in Gentran:Director for your trading partners. Please see the *Gentran:Director User's Guide* for more information on creating trading relationships.

10. **Testing the Translation Object**

For a screen entry translation object, create a new document in the Document Editor, using that translation object. For a print translation object, print a document. Verify that the output is correct.

Form Integration Basics

This chapter explains some of the basic (not lengthy) form integration functions. Some of the functions are performed initially (creating a form), some are performed as you define the form, and some are performed after the form is completed (compiling and printing a report).

This chapter includes the following topics:

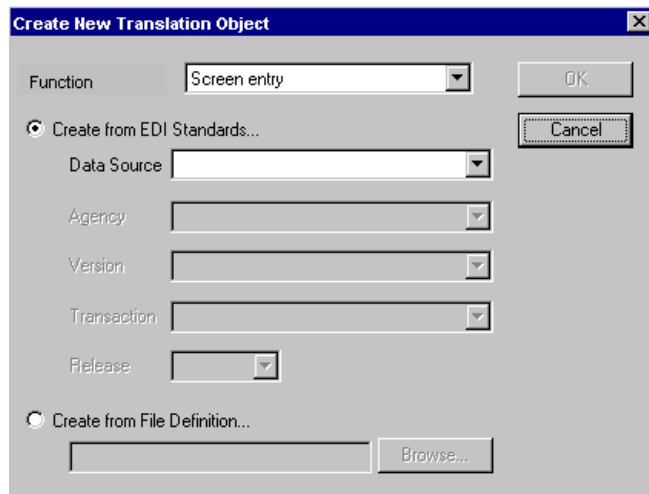
- Creating a New Form
- Using an EDI File Format
- Opening a New Form
- Saving a Form
- Naming a Form
- Compiling a Form
- Printing a Report
- Customizing Your Display
- Setting the Default Date Format
- Selecting Which Form Components to Use
- Cutting, Copying, and Pasting
- Setting Up the Document Name and Reference
- Formatting Your Data

Creating a New Form

You can create two different types of forms: Screen Entry (for keying documents in Document Editor) and Print (for printing documents). The initial procedures for creating a new form of both types is the same.

Complete the following steps to create a new form:

1. Select New from the File menu to display the Create New Translation Object dialog.



2. From the Translation Object Function list, select the type of form. Valid values are “Screen entry” or “Print.”
3. If you want to create the form from an EDI Standards database, click the “Create from EDI Standards...” option and continue with step 4.

If you want to load a form from a previously-saved file definition, click the “Create from File Definition” option and continue with step 9.

4. From the Data Source list, select the ODBC data source you wish to use.
5. Select the EDI standard you are using for the input side of the form from the Agency list. Valid values are:

- [A] TRADACOMS
- [E] EDIFACT
- [O] ODETTE
- [T] TDCC
- [U] UCS/WINS
- [X] ANSI X12

6. Select the version of the standard from the Version list. The versions that are available depend on which standard you selected.
7. Select the transaction set (document or message) from the Transaction list. The transactions that are available depend on which standard and version you selected.
8. For TRADACOMS only, select the release number of the version that you are using from the Release list. The releases that are available depend on which standard, version, and transaction you selected.
9. In the Create from File Definition box, type the path and file name that you want to load or click Browse... and the Open File Definition dialog is displayed.

The “Look in” list should reflect the drive\folder where GENTRAN:Integrator is installed.

The default is GENTRANNT\Bin. If the current folder is not the correct one, select the appropriate folder.

Select the type of file that you want to display from the “Files of type” list. The default file extension is .IFD.

Select the file definition that you want to load into the system from the list. Double-click the filename or click OPEN to load the selected file format definition.

10. Click OK to generate a new form (this may take a few seconds). The new form is displayed in the Gentran:Director - Forms Integration Window.

- After you create a new form, it is important to define the details of the form and save it. Select Save from the File menu to display the Translation Object Details dialog.

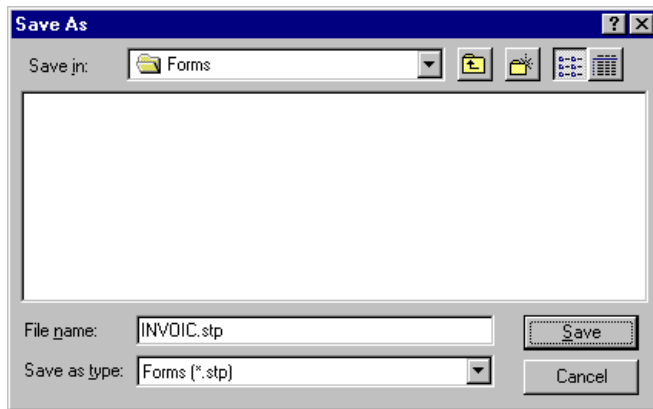
- Complete the Author box. This box typically contains the name of the person who created the form.
- Complete the Description box. This box must be unique because it is used by the system to identify the form. We recommend that you use the following identifying characteristics: the partner that this form is used for, the standard, the version, the type of transaction this form uses, and the type of form. For example, “MWT X 3030 850 Print” is the description of a print translation object used to print documents for partner MWT, for an ANSI X.12 version 003030 Purchase Order (850).
- Verify the type of translation object in the Translation Object Function list.
- Click the System check box if you want to designate this form as a system translation object (one that is used internally by the system).



You should only select this check box if you are certain to want to define this translation object as a system translation object. Once a translation object is registered with Gentran:Director, it cannot be deleted.

- Complete the Major and Minor boxes if you want to designate a version number for this translation object. Type the version number in the Major box and the release number in the Minor box.

17. The Input and Output boxes (Agency, Version, Transaction, Release, and F Group) contain the EDI format. You can change the information in these boxes, but it will not alter the content of the form. You may wish to alter these boxes if, for example, you want the form to reflect a standards version that is not loaded on your system. You can change the version on this dialog, and then alter the form to be compliant with that version.
18. Click OK to save the translation object details and access the Save As dialog.



19. We recommend that you store forms in the translation object folder where Gentran:Director is installed.
20. Type the name of the form (one to eight characters) in the “File name” box. The default file extension is .STP.
21. Click SAVE to save the form.

After you finish creating and saving a new form, you need to customize the EDI file format and generate the form. The steps you take are different, depending on whether the form is a screen entry or a print form.

Using an EDI File Format

Gentran:Director generates an EDI file format for you, based on the standard (agency), version, transaction set, and release 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 *Creating a New Form* on page 16 for more information on generating an EDI file format when you create a new form. You can modify the system-generated EDI file format by modifying the properties of the form 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:Director standards database, it may be appropriate for you to define the EDI file format yourself.

Regardless of whether the system generates the EDI file format or you define it, the specific EDI form components that you use depend on the type of form 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 form components you are using before generating or defining an EDI file format.

Verifying EDI Delimiters

If you are using an EDI standard that contains composite elements or sub-elements, you must verify that Gentran:Director 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.



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

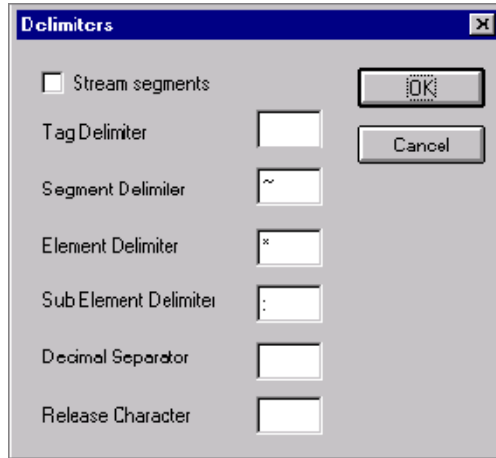
Complete the following steps to verify EDI delimiters:

1. Click the EDI file icon with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu to display the File Format Properties dialog.

The screenshot shows the 'File Format Properties' dialog box with the following fields and controls:

- Identification:**
 - Name: BEO
 - Desc: PURCHASE ORDER
- Loop Control:**
 - Min Usage: 1
 - Max Usage: 1
- Display:**
 - Promote group to parent frame
 - Frame Caption: Purchase Order
 - List Caption: (empty)
 - Frame Help Text: (empty text area)
 - Listbox Help Text: (empty text area)
- Buttons (right side):** OK, Cancel, Promote, Split..., Rules..., Delimiters...

2. Click DELIMITERS... to access the Delimiters dialog.



3. Verify the required delimiters for the EDI standard you are using. If the delimiters differ from the defaults in the table below, type either the character or the hexadecimal value in the correct box.
4. Click OK to exit the File Format Properties dialog.

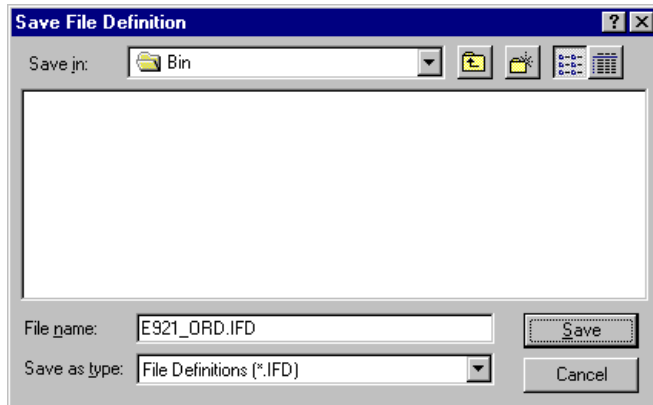
Saving a File Definition

Gentran:Director enables you to save an individual file format definition so that you can use it as a guide in future forms. This provides you with a quick way to build your form. This file format definition can also be used to create a new map using Application Integration.

Complete the following steps to save an individual file format definition:

1. Use the right mouse button to click the fixed-format icon (either the INPUT or OUTPUT side of the map) to access the shortcut menu.

2. Select Save File Definition As... from the shortcut menu. The Save File Definition dialog is displayed.



3. The "Save in" list should reflect the drive\folder where Gentran:Director is installed. The default is GENSRVNT. If the current folder is not the correct one, select the appropriate folder.
4. Select the file definition from the list or type the name of the file definition (one to eight characters) in the "File name" box. The default file extension is .IFD.
5. Click SAVE to save the file definition.

Creating a Group

A group contains related segments and/or groups 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).

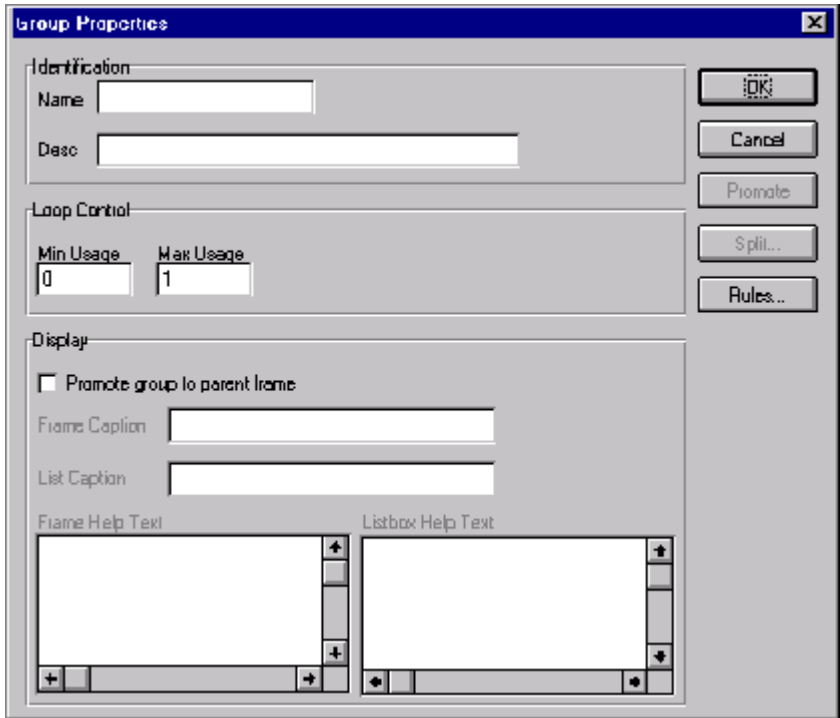
Complete the following steps to create a group:

1. To create a group that is at the same level as a selected form component, highlight the form component and select Insert from the Edit menu. Select Group from the cascading menu.

OR

To create a group that is subordinate to a selected form component, highlight the form component and select Create Sub from the Edit menu. Select Group from the cascading menu.

The Group Properties dialog is displayed.



2. In the Name box, type the segment loop identification code.



Do not use spaces or dashes (-) in the Name box. You can use the underscore (_) to separate words.

3. In the Desc box, type the loop description. This box is used to provide a brief explanation of the loop.
4. In the Min Usage box, type the minimum number of times the loop must be repeated. For a conditional loop, the minimum usage should always be "0" (zero). For a mandatory loop, the minimum usage should be "1" or greater.
5. In the Max Usage box, type the maximum number of times the loop can be repeated.
6. Click OK to create the group.

Modifying Group Properties

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) Complete the following steps to modify the properties of a group:

1. Click the group you want to modify with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu.

The Group Properties dialog is displayed.

2. If you want to modify the group name, type the new group name in the Name box.



If a segment or element occurs more than once in a form 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 form.

3. If you want to modify the group description, type the new description in the Desc box. This box is used to provide a brief explanation of the loop.
4. In the Min Usage box, type the minimum number of times the loop must be repeated. For a conditional loop, the minimum usage should always be “0” (zero). For a mandatory loop, the minimum usage should be “1” or greater.
5. In the Max Usage box, type the maximum number of times the loop can be repeated.
6. Click OK to save the changes to the group.

Modifying Segment Properties

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. Please see *Using Loop Start and Loop End Segments* on page 80 for more information on using loop start and loop end segments.

Complete the following steps to modify the properties of a segment:

1. Click the segment you want to modify with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu.

The Segment Properties dialog is displayed.

2. If you need to modify the name of the segment, type the new segment name in the Name box.



If a segment or element occurs more than once in a form 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 form.

3. If you want to modify the description of the segment, type a new description in the Desc box. This box is used to provide a brief explanation of the segment that allows you to differentiate it from similar segments.
4. In the Min Usage box, type the minimum amount of times the segment must repeat.



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.”

5. If you want the segment to repeat (loop), type the maximum amount of times it can repeat in the Max Usage box.

- Unless you want to indicate that the segment is the beginning of a loop (Loop Start) or the end of a loop (Loop End), accept the default of Normal for the Normal/Loop Start/Loop End options. These options are defined as follows:

Normal = This segment is in the loop but is not the beginning or ending segment

Loop Start = This segment marks the beginning of the loop

Loop End = This segment marks the end of the loop



If you need to modify the display properties of the segment, you can do so in the Display section of the Segment Properties dialog. Please see *Chapter 6, Customizing Screen Entry Translation Objects* on page 205 for more information on modifying display properties.

- Click OK to save the changes to the segment.

Modifying Composite Properties

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

Complete the following steps to modify the properties of a composite:

- Click the composite you want to modify with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu.

The Composite Properties dialog is displayed.

- If you want to change the composite name, type the data element identification number in the Name box.



If a segment, composite, or element occurs more than once in a form 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 form.

3. Click the Mandatory check box to indicate whether or not this composite must be present. A checkmark means that the composite is mandatory.
4. If you want to change the composite description, type the new description in the Description box.
5. Click OK to save the changes to the composite.

Modifying Element Properties

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.

Complete the following steps to modify the properties of an element:

1. Double-click the element you want to modify to access the Field Properties dialog.

Data Validation						
	Min	Max	Type	Screen Format	EDI Format	
<input checked="" type="checkbox"/> Mandatory	1	35	String	X	X	

2. To change the name of the element, type the new name in the Name box. The name is typically the element sequence number.



If a segment or element occurs more than once in a form 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 form.

3. To change the description of the element, type the new description in the Description box. The description is used to provide a brief explanation of the element that allows you to differentiate it from similar elements.
4. If you want to designate the element as “mandatory” (must be present), click the Mandatory check box.

5. To change the minimum length of the element, type the minimum length in the Min box. If the data is less than the minimum length, a compliance error is generated when translation occurs.
6. To change the maximum length of the element, type the maximum length in the Max box.
7. From the Type list, select the type of the element. Valid values are:
 - String = alphanumeric element
 - Number = numeric or real element
 - Dt/Tm = date or time element
8. From the Screen Format list, select how the data element is formatted on the form. The choices for this box depend on the type of the element you selected from the Type list. If you choose Number or Dt/Tm in the Type box, you can select the data format from the Format list. If you selected String from the Type box, you should type a syntax token to denote that this element must be formatted as the specified syntax token dictates (the default syntax token is “X”). Please see *Using Syntax Tokens* on page 53 for more information on defining and using syntax tokens.
9. From the EDI Format list, select how the data element is formatted in the EDI file. The choices for this box depend on the type of the element you selected from the Type list. If you choose Number or Dt/Tm in the Type box, you can select the data format from the Format list. If you selected String from the Type box, you should type a syntax token to denote that this element must be formatted as the specified syntax token dictates (the default syntax token is “X”). Please see *Using Syntax Tokens* on page 53 for more information on defining and using syntax tokens.
10. Click OK to save the changes to the element.

Activating EDI Form Components

When Gentran:Director generates the EDI sides of the form, 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 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 the form.

Complete the following steps to activate groups, segments, composites, and elements:



1. Click the **ACTIVATE** button on the Main Toolbar.
2. Open (double-click) each group that contains groups or segments that need to be activated.



As an alternative to opening each form component (steps 2, 5, and 7), you can select **Expand All from the **View** menu to open every form component. You can also select a form component and select **Expand Branch** from the **View** menu to open that form component.**

3. Click each inactive group that you need to use to activate it.s.



If you accidentally click a group, segment, composite, or element that you did not mean to activate, click the item with the right mouse button to access the shortcut menu. Select **Deactivate from the shortcut menu.**

4. Click each inactive segment that you need to use to activate it.
5. Open (double-click) each segment that contains composites or elements that need to be activated.
6. Click each inactive composite that you need to use to activate it.
7. Open (double-click) each composite that contains elements that need to be activated.
8. Click each inactive element that you need to use to activate it.
9. Once you have activated all the necessary groups, segments, and elements, click the **ACTIVATE** button on the Main Toolbar to turn activation mode off.

Promoting Groups and Repeating Segments

The Promote function extracts one iteration (instance) of a group or repeating segment. It also sets the promote flag and places the active elements in the parent frame.

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. Please see *Splitting Groups and Repeating Segments* on page 32 for more information on splitting groups and repeating segment. Please see *Using Copy, Cut, and Paste* on page 33 for more information on copying, cutting, and pasting.



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

Complete the following steps to promote a group or repeating segment:

1. Highlight the group or repeating segment from which you want to extract one iteration.
2. Click the PROMOTE button on the Main Toolbar. This extracts one iteration (instance) of the looping structure.



Splitting Groups and Repeating Segments

The Split function enables you to break a group or repeating segment into two loops. You typically use this function when you need more than one instance of the same form component that still occurs multiple times. You can also use the Split function to split one instance of a group without setting the promote flag (the Split function never sets the promote flag).

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.



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

Complete the following steps to split a group or repeating segment:

1. Highlight the group or repeating segment from which you want to extract one iteration.



2. Click the `SPLIT` button on the Main Toolbar. The Split dialog is displayed.

Split

First Loop Entries: 0

OK

Cancel

3. 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. For 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.



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.

4. Click OK to exit the Split dialog.

Using Copy, Cut, and Paste

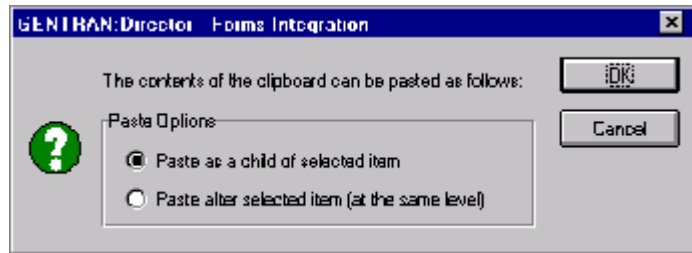
The Copy, Cut, and Paste functions are typically used to move EDI information in the form. You may need to use these functions if you are using an EDI standard differently, or if you need to create nested looping structures.

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

Complete the following steps to cut, copy, and paste a form component:

1. Select the form component that you want to cut or copy.
2. Click the Copy button on the Main Toolbar.
3. If you are pasting the form component in another form, open that form, if it is not already open. Please see *Opening a Form* on page 38 for more information on opening a form.

4. Select the form component that you want the cut or copied selection pasted *after*.
5. Click the PASTE button on the Main Toolbar to paste the contents of the Clipboard. If the form component that you selected is a group, Gentran:Director 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.

Complete the following steps to create a nested looping structure:

1. Highlight the HL group. Click the Copy button on the Main Toolbar.
2. Select the form component that you want the cut or copied selection to be pasted *after*.
3. Click the PASTE button on the Main Toolbar to paste the contents of the Clipboard.

If the form component that you selected is a group, Gentran:Director 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.

4. Click the Paste after selected item (at the same level) option and click OK to paste the Clipboard contents at the same level as the group.

5. Select Properties from the Edit menu to access the Group Properties dialog.

The screenshot shows the 'Group Properties' dialog box with the following details:

- Identification:** Name: 0100_HL, Desc: (empty)
- Loop Control:** Min Usage: 1, Max Usage: 200000
- Display:** Promote group to parent frame, Frame Caption: 0100_HL, List Caption: 0100_HL
- Help Text:** Frame Help Text and Listbox Help Text (both empty)
- Buttons:** OK, Cancel, Promote, Split..., Rules...

6. In the Desc box, type a specialized definition to identify this group as containing shipment information.
7. Click OK to save the change and exit the Group Properties dialog.
8. Select the form component in the HL that you want the nested HL group (order information) pasted *after*.
9. Click the PASTE button on the Main Toolbar to paste the contents of the Clipboard. Click the Paste after selected item (at the same level) option and click OK on the prompt dialog to paste the Clipboard contents at the same level as the group. The second HL group is added as a nested loop to the original HL loop.

10. Select Properties from the Edit menu to access the Group Properties dialog for the second HL group.

11. In the Desc box, type a specialized definition to identify this group as containing order information.
12. Click OK to save the change and exit the Group Properties dialog.
13. Double-click the second HL group (order information) to open it.
14. Select the form component in the HL that you want the nested HL group (item information) pasted *after*.
15. Click the PASTE button on the Main Toolbar to paste the contents of the Clipboard. Click the Paste after selected item (at the same level) option and click OK on the prompt dialog to paste the Clipboard contents at the same level as the group. The third HL group (item information) is added as a nested loop of the second HL group.

16. Select Properties from the Edit menu to access the Group Properties dialog for the third HL group.

The screenshot shows the 'Group Properties' dialog box with the following details:

- Identification:** Name: 0100_HL3, Desc: (empty)
- Loop Control:** Min Usage: 1, Max Usage: 200000
- Display:** Promote group to parent items: ; Frame Caption: (empty); List Caption: 0100_HI; Frame Help Text: (empty); Listbox Help Text: (empty)
- Buttons:** OK, Cancel, Promote, Split..., Rules...

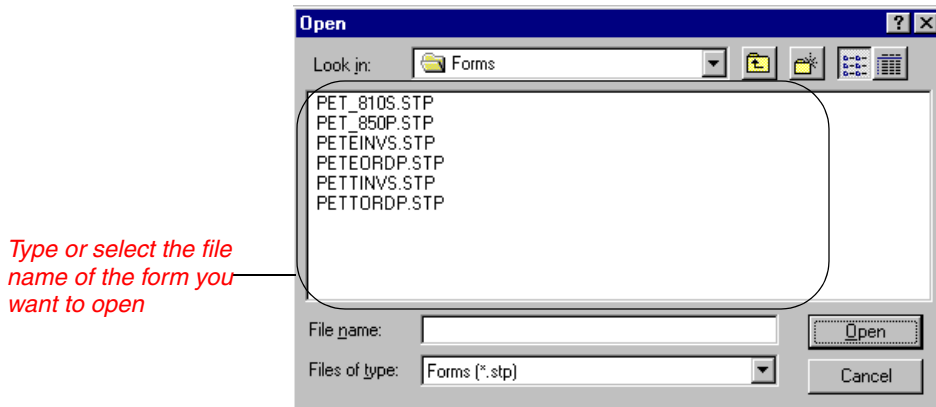
17. In the Desc box, type a specialized definition to identify this group as containing line item information.
18. Click OK to save the change and exit the Group Properties dialog.
19. You are now ready to activate the appropriate groups, segments, and elements in the new HL nested looping structure. Please see *Selecting Which Form Components to Use* on page 67 for more information on activating.

Opening a Form

Opening a form loads it into Gentran:Director and enables you to view or edit it. The Forms Integration subsystem allows you to have multiple forms open, but only one form has focus.

Complete the following steps to open a form:

1. Select Open from the File menu to display the Open dialog.



2. The “Look in” list should reflect the drive\folder where Gentran:Director is installed. The default is GENTRAN Director\BIN\Forms. If the current folder is not the correct one, select the appropriate folder.
3. Select the type of file that you want to display from the Files of Type list. The default file extension is .STP.
4. Select the map that you want to load into the system from the list or type it in the File Name box.
5. Double-click the filename or click OPEN to open the selected form.

Saving a Form

The Save function writes (saves) the currently selected form to disk. If the form was saved before and you completed the Translation Object Details dialog, complete the following step to save the current form:

Select Save from the File menu or the Main Toolbar. The system saves the current form under the same name that it was previously saved.



If you have performed any functions that invalidate the current format of the translation object, the Save function is not available until you regenerate the layout.

If it is a new form that you have not saved before, complete the following steps:

1. Select Save from the File menu to display the Translation Object Details dialog.

Summary			
Author	Description	Translation Object Function	System
		Screen entry	<input type="checkbox"/>

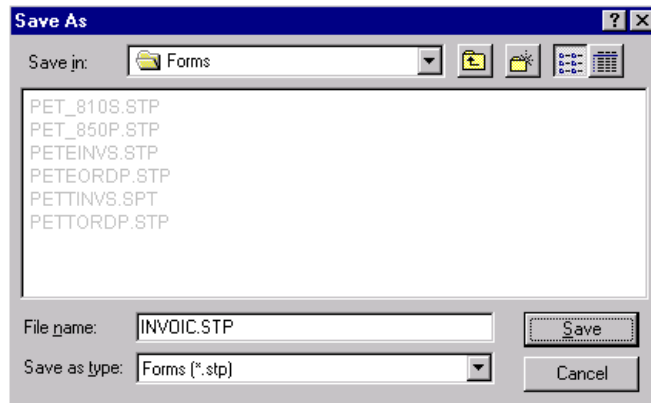
Version Control		
Major version	Minor version	Compiled on
0	0	

EDI Associations					
	Input		Output		
Agency	E	Edifact	E	Edifact	
Version	092001	EDIFACT 1992 REL	092001	EDIFACT 1992 REL	
Transaction	INVOIC	INVOICE MESSAG	INVOIC	INVOICE MESSAG	
Release	0		0		
F Group					

2. Complete the Author box. This box typically contains the name of the person who created the form.
3. Complete the Description box. This box must be unique because it is used by the system to identify the form. We recommend that you use the following identifying characteristics: the partner that this form is used for, the standard, the version, the type of transaction this form uses, and the type of form. For example, "MWT X 3030 850 Print" is the

description of a print translation object used to print documents for partner MWT, for an ANSI X.12 version 003030 Purchase Order (850).

4. Verify the type of translation object in the Translation Object Function list.
5. Complete the Major and Minor boxes if you want to designate a version number for this translation object. Type the version number in the Major box and the release number in the Minor box.
6. The Input and Output boxes (Agency, Version, Transaction, Release, and F Group) contain the EDI format. You can change the information in these boxes, but it will not alter the content of the form. You may wish to alter these boxes if, for example, you want the form to reflect a standards version that is not loaded on your system. You can change the version on this dialog, and then alter the form to be compliant with that version.
7. Click OK to save the translation object details and access the Save As dialog.



8. Type the name of the form (one to eight characters) in the “File name” box, or accept the default that the system prompts for you. The default file extension is .STP.
9. Click SAVE to save the form.

Naming a Form

You allocate a filename to a form when you save the form for the first time or use the Save As function to save a copy of the form under a different name. The name of the form must be one to eight unique characters that define the form in a meaningful way. For example, “850_MWT.STP” is the name of a form for an ANSI Purchase Order (850) for partner MWT.

The naming conventions in Gentran:Director are as follows:

- The source form (form created in Gentran:Director) should be named using the file extension **.STP** (i.e., MWT_850.STP)
- The compiled translation object (translation object that is the result of compiling the source form) should be named using the same filename that was allocated to the source form BUT with the file extension **.TPL** (i.e., MWT_850.TPL)

We recommend that you use these naming convention to distinguish source forms and compiled translation objects from other types of files. We also recommend that you use the same filename (different file extension) for a source form and the translation object that was compiled from it.



Be very careful not to overlay the source form with the compiled translation object. Use the .TPL file extension to distinguish the translation object.

Complete the following steps to save a copy of the current form under a different name:

1. Select Save As... from the File menu to display the Save As dialog.
2. Type the name of the form (one to eight characters) in the “File name” box, or accept the default that the system prompts for you (using the default .STP file extension). Click **SAVE** to save the form copy under the new name and exit the Save As dialog.

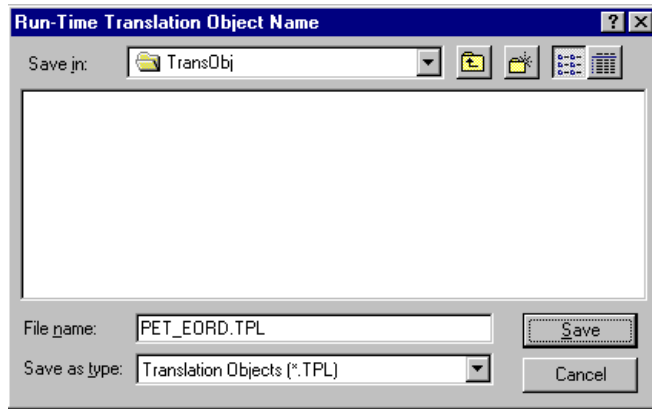
Compiling a Translation Object

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

This translation object must be registered with the Gentran:Director system before you can use it. You use the Compile function after the form is completed and saved. Please see *Saving a Form* on page 39 for more information on saving a form.

Complete the following steps to compile a form and generate a translation object:

1. Select Compile from the File menu or the Main Toolbar to display the Run-Time Translation Object Name dialog.



2. The system defaults the name of the translation object (.TPL file) with the same filename (one to eight characters long) as you named the form (.STP file). Preserving the same filename (with different file extensions) means that the relationship between the source form and the compiled translation object will remain evident. For example, if the source form name is MWT_850.STP, the compiled translation object name is MWT_850.TPL.



Be very careful not to overlay the source form with the compiled translation object. Use the .TPL file extension to distinguish the translation object.

- From the “Save in” list, change the drive\folder where the compiled translation object is stored, if necessary.



Do NOT store the compiled translation object in the GENTRAN Director\RegTransObj folder. This folder is reserved for storing a copy of each translation object you register with Gentran:Director.

- Click SAVE and the system compiles the form and creates a translation object. The Compile Errors dialog is displayed to indicate any errors or warnings. Click OK to exit this dialog. If any errors occurred, correct the problems and recompile the translation object. The date that the translation object was compiled on is automatically loaded into the Compiled on box on the Translation Object Details dialog.

Summary	
Author	Sterling
Description	Print 850 Pet Zone
Translation Object Function	Print
System	<input type="checkbox"/>

Version Control		
Major version	1	Minor version
	0	Compiled on
		05-Dec-1995

EDI Associations					
	Input		Output		
Agency	X	X12	X	X12	
Version	003040	ANSIX 12 VERSIO	003040	ANSIX 12 VERSIO	
Transaction	850	PURCHASE ORDE	850	PURCHASE ORDE	
Release	0		0		
F Group	PD		PD		

- Select Save from the File menu to save the source form with the Compiled on date.
- This translation object must be registered with the Gentran:Director system before you can use it. See the Gentran:Director *User's Guide* for information on registering a translation object.

Printing a Report

The Print function enables you to print a report for the current form.



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

Complete the following steps to print a report.

1. Select Print from the File menu or the Main Toolbar to access the Print Options dialog.
2. Select the reports that you want by clicking the appropriate check boxes in the Report section. Click OK and the Print dialog is displayed.
3. Set the appropriate options. Click OK if you do *not* need to change Setup options, and the report is printed.

Click **SETUP...** if you need to access printer setup information (to select a specific printer, paper orientation, or size and source of paper). The Print Setup dialog is displayed.

4. Set the appropriate options. Click OK if you do *not* need to change Print Options (dithering, intensity control, or print true type as graphics), and you return to the Print dialog. Click OK on the Print dialog and the report is printed.

Click **OPTIONS...** if you need to access printer setup options information. The Options dialog is displayed.

5. Set the appropriate options. Click OK to return to the Print Setup dialog. Click OK on the Print Setup dialog to return to the Print dialog. Click OK on the Print dialog and the report is printed.

Formatting Your Data

When you define or modify an element, you must specify the type and format. The options that are available for the format of the element depend on which type you select (string, number, or Dt/Tm).

Using the String Type

A string type element contains one or more printable characters. If you specify that an 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/or numbers that are allowed to be used for a string-type element. You can then use the syntax tokens in the Format box of the Field Properties dialog. This enables you to define what type of characters should be used while compliance checking each element (i.e., alphanumeric within a certain range, numeric within a certain range, etc.).

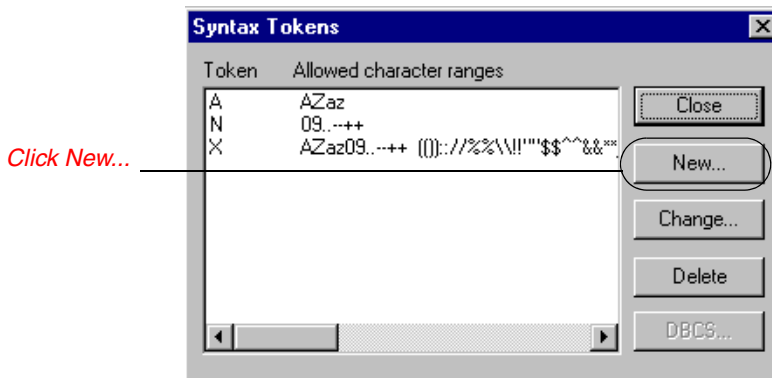
Creating a Syntax Token (Western European languages)

Complete the following steps to create a syntax token:



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

1. Select Syntax Tokens from the Edit menu to display the Syntax Tokens dialog.



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.



The DBCS... button is only available if you are executing a double-byte version of Windows NT.

2. Click NEW... to access the Edit Syntax Token dialog.



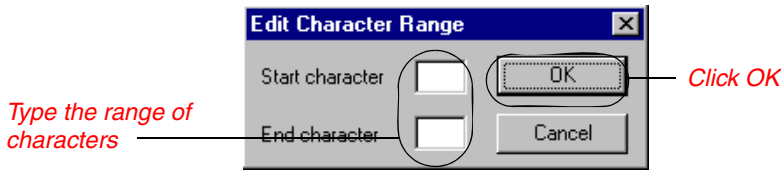
3. In the Token box, type the unique one-character alphanumeric value that the system recognizes as containing the allowed range of characters you designate.



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” and the character range “a” through “z”. This indicates that token “A” only allows upper-case and lower-case alphabetic.

- Click the NEW... button on the Edit Syntax Tokens dialog. The Edit Character Range dialog is displayed, which enables you to create a character range.



Gentran:Director 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 Œ and œ.

Start	End
0x8C	0x8C
0x9C	0x9C

- In the Start character box, type the character that begins the allowed token range. For example, if the character range you want to define is “B” through “D,” type “B” in the Start character box. 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. In the End Character box, type the character that terminates the allowed token range. Use the guidelines from step 5.
7. Click OK to return to the Edit Syntax Tokens dialog.
8. To add additional character ranges to the syntax token, repeat steps 4 - 7 as many times as necessary.
9. Click OK to save the syntax token and return to the Syntax Tokens dialog.
10. Click CLOSE to exit the Syntax Tokens dialog.

Creating a Syntax Token (Far Eastern languages)

The DBCS syntax token function enables you to create a form that accepts double-byte characters. If you are running Gentran:Director on a Chinese, Japanese, or Korean version of Windows NT, the DBCS button on the Syntax Tokens dialog is available.

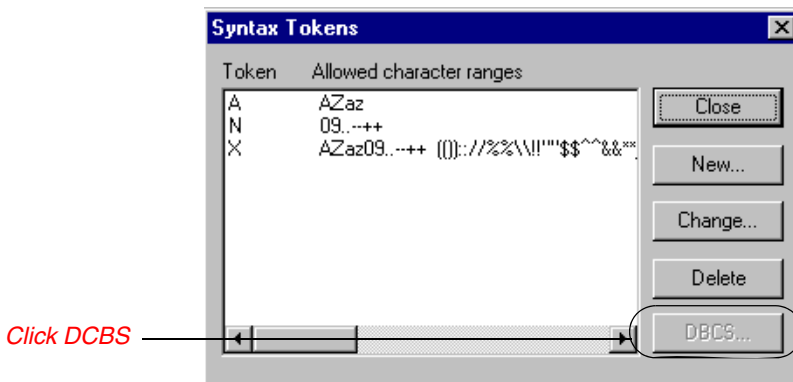
DBCS tokens are displayed only in the DBCS Syntax Tokens dialog, not in the list in the Syntax Tokens dialog.



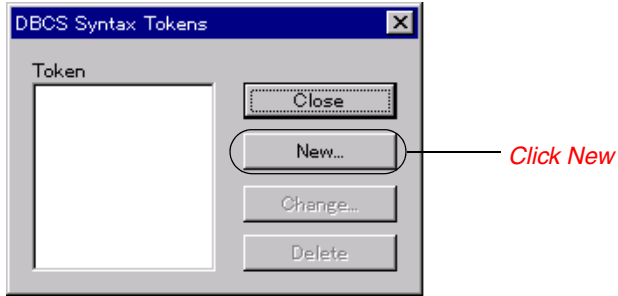
When you set up a DBCS token, it applies only to that form. You may need to set one up for each form that you create.

Complete the following steps to create a DBCS syntax token:

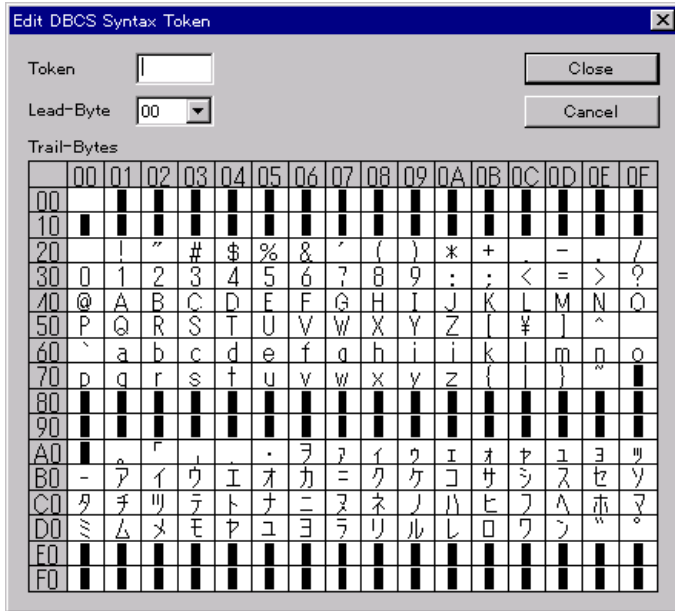
1. Select Syntax Tokens from the Edit menu to display the Syntax Tokens dialog.



- Click DCBS to access the DCBS Syntax Token dialog.



- Click NEW to access the Edit DBCS Syntax Token dialog.



Double-byte characters are composed of a lead byte and a trail byte. In the above example, the character A is code point 0041.

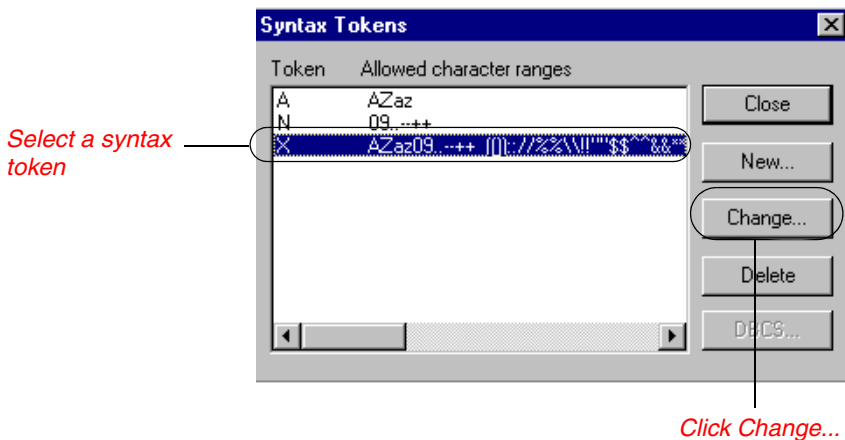
- In the Token box, type the unique one-character alphanumeric value that the system recognizes as containing the allowed range of characters you designate.

5. Do you want to include all characters in the Trail-Bytes section of the dialog?
 If *yes*, continue with step 7..
 If *no*, continue with step 6..
6. 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.
8. Click **CLOSE** to exit the Syntax Tokens dialog.

Editing a Syntax Token

Complete the following steps to edit a syntax token:

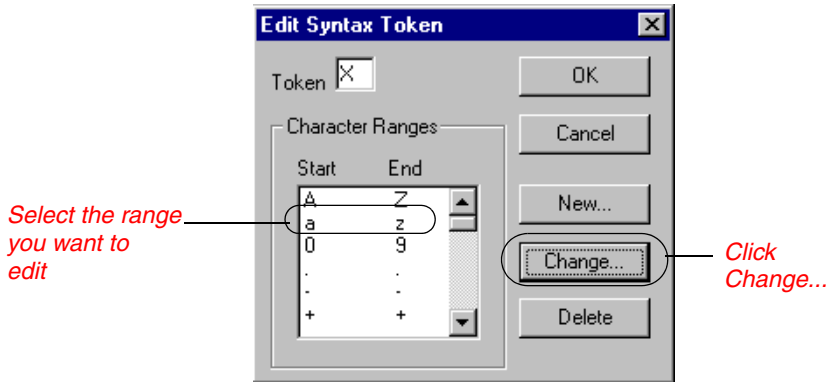
1. Select Syntax Tokens from the Edit menu to display the Syntax Tokens dialog.



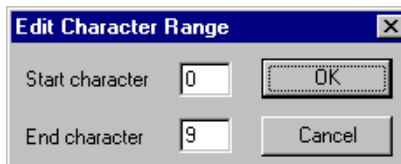
The Token column contains the value designated as the syntax token, for each existing syntax token. The Token contains a range of characters that, when applied to an element, dictate the way that element must be formatted. If the element is not formatted as specified, the system generates a compliance 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.

2. Select a syntax token and click CHANGE... to access the Edit Syntax Token dialog.



3. If you want to change the token value, type the unique one-character alphanumeric value that the system recognizes as containing the allowed range of characters you designate in the Token box.
4. Select the character range you want to modify and click CHANGE... on the Edit Syntax Tokens dialog. The Edit Character Range dialog is displayed.



5. In the Start character box, type the character that begins the allowed token range.



The Start character and End character can only be one (1) character, upper- or lower-case alphabetic or numeric “1” - “9.”

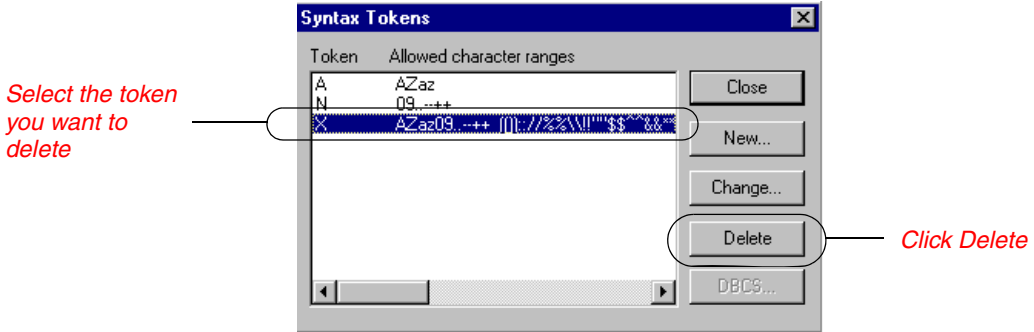
6. In the End Character box, type the character that terminates the allowed token range.
7. Click OK to return to the Edit Syntax Tokens dialog.
8. To edit additional character ranges for this syntax token, repeat steps 4 - 7 as many times as necessary.
9. Click OK to save the syntax token and return to the Syntax Tokens dialog.
10. Click CLOSE to exit the Syntax Tokens dialog.

Deleting a Syntax Token

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

Complete the following steps to delete a syntax token:

1. Select Syntax Tokens from the Edit menu to display the Syntax Tokens dialog.



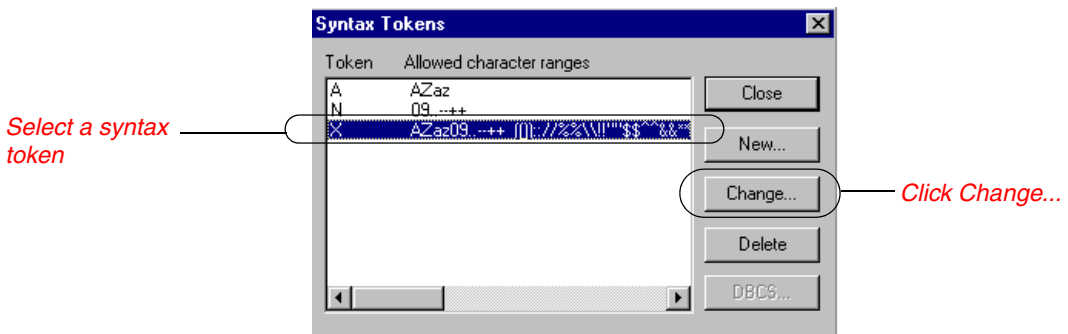
2. Highlight the token that you want to delete and click DELETE.



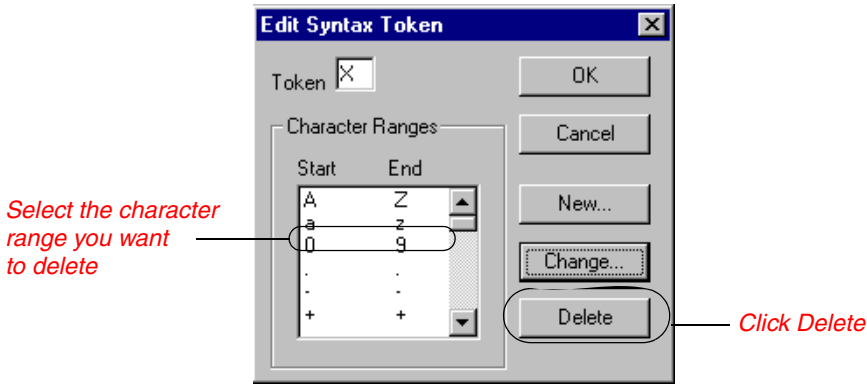
The selected entry is deleted without further warning.

Complete the following steps to delete a character range from a syntax token:

1. Select Syntax Tokens from the Edit menu to display the Syntax Tokens dialog.



2. Select a syntax token and click CHANGE... to access the Edit Syntax Token dialog.



3. Highlight the character range that you want to delete and click DELETE.



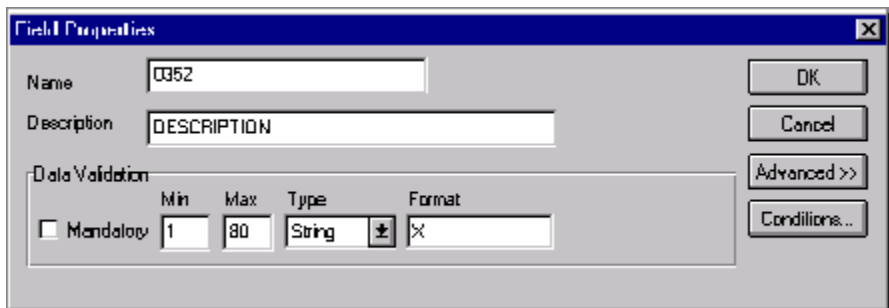
The selected entry is deleted without further warning.

Using Syntax Tokens

You can use syntax tokens in the Format box of the Field Properties dialog. 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 element. The use of a syntax token enables you to define what characters should be used while compliance checking this element (i.e., alphanumeric within a certain range, numeric within a certain range, etc.).

Complete the following steps to use syntax tokens:

4. Double-click an existing element. The Field Properties dialog is displayed.



5. From the Type list, select String. This indicates that the element contains characters.

6. In the Format box, type an existing syntax token to denote that this element must be formatted as the specified syntax token dictates. Please see *Creating a Syntax Token (Western European languages)* on page 45 for more information on creating syntax tokens.
7. Click OK to exit the Field Properties dialog.

Using the Number Type

A number type element contains either an implied decimal or real number that can be mathematically manipulated. If you specify that an element 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.

An “N” formatted number is a number that has an implied decimal point (e.g., “2.01” is formatted as “201”). An “R” formatted number has an explicit decimal point (e.g., “2.01” is formatted as “2.01”), and truncates trailing zeroes. Whether you use the “N” or “R” format depends on the requirements of your application. Regardless of whether you use the “N” or “R” format, you must also indicate the number of decimal places in the element.



You can specify that *Gentran:Director* displays and prints all numeric-format elements using decimal points (formatted as real). This default data format is used when elements are read from the standards database. You specify this option on the Standard Formats tab of the Preferences dialog (choose Preferences... from the Options menu and click the Standard Formats tab). Then select the “Display and print numerics using decimal points (Rx format)” check box.

Complete the following steps to use the number type:

1. Double-click an existing element. The Field Properties dialog is displayed.

Data Validation				
	Min	Max	Type	Format
<input checked="" type="checkbox"/> Mandatory	1	10	Number	N2

2. From the Type list, select Number. This indicates that the element is a numeric or real number that can be mathematically manipulated.
3. From the Screen Format list, select the format for the number.
 - Number formats beginning with “R” represent numbers with an explicit decimal point. For example, if the format you select is R2, then numbers is formatted with two places to the right of the decimal point (e.g., “2.01”).
 - Number formats beginning with “N” represent numbers with an implied decimal point. The decimal point is not displayed, but the system still processes the number correctly. For example, if the format you select is N2, then numbers is formatted with two places to the right of an implied decimal point.

Valid number formats are the following:

- | | |
|----|--|
| R0 | Number formatted with an explicit decimal point and no decimal places |
| R1 | Number formatted with an explicit decimal point up to 1 decimal place |
| R2 | Number formatted with an explicit decimal point up to 2 decimal places |
| R3 | Number formatted with an explicit decimal point up to 3 decimal places |
| R4 | Number formatted with an explicit decimal point up to 4 decimal places |
| R5 | Number formatted with an explicit decimal point up to 5 decimal places |
| R6 | Number formatted with an explicit decimal point up to 6 decimal places |

R7	Number formatted with an explicit decimal point up to 7 decimal places
R8	Number formatted with an explicit decimal point up to 8 decimal places
R9	Number formatted with an explicit decimal point up to 9 decimal places
N0	Number formatted with an implicit decimal point and no decimal places
N1	Number formatted with an implicit decimal point up to 1 decimal place
N2	Number formatted with an implicit decimal point up to 2 decimal places
N3	Number formatted with an implicit decimal point up to 3 decimal places
N4	Number formatted with an implicit decimal point up to 4 decimal places
N5	Number formatted with an implicit decimal point up to 5 decimal places
N6	Number formatted with an implicit decimal point up to 6 decimal places
N7	Number formatted with an implicit decimal point up to 7 decimal places
N8	Number formatted with an implicit decimal point up to 8 decimal places
N9	Number formatted with an implicit decimal point up to 9 decimal places



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".



Usually, the number formats in the Screen Format box and in the EDI Format box are the same. However, if there is an "N" format selected in the EDI Format box, you might want to select an "R" format in the Screen Format box so that the decimal point in the numbers is displayed.

- Verify that the number format selected in the EDI Format list is the correct format that is written to the EDI file.



A number format must be designated for number-type elements in the EDI Format box, even for hidden fields.

- Click OK to exit the Field Properties dialog.

Using the Dt/Tm Type

A Dt/Tm type element contains a date or time. If you specify that an element is a Dt/Tm type, you must specify exactly how the date or time must be formatted.

Complete the following steps to use the Dt/Tm type:

- Double-click an existing element. The Field Properties dialog is displayed.

Data Validation				
	Min	Max	Type	Format
<input checked="" type="checkbox"/> Mandatory	6	6	Dt/Tm	YYMMDD

- If the screen format for the date or time is longer than the length in the Max box, type a new maximum field length in the Max box. This value should be the maximum number of digits, including separators, that you expect for the date or time.



The value in the Max box should be the same as the value in the Display Length box on the Display Properties dialog (click DISPLAY... on the Field Properties dialog to access the Display Properties dialog).

- Verify that Dt/Tm is selected in the Type list, indicating that the field value must be a date or a time.

4. From the Screen Format list, select the format for how you want the date or time to be formatted in the screen entry translation object.

Valid Dt/Tm formats are the following:

MM/DD/YY	Two-digit month, separator, two-digit day, separator, last two digits of year (example: 12/15/97 for December 15, 1997)
YYYYMMDD	Four-digit year, two-digit month, two-digit day (example: 19971215)
YYYY/MM/DD	Four-digit year, separator, two-digit month, separator, two-digit day (example: 1997/12/15)
DD/MM/YYYY	Two-digit day, separator, two-digit month, separator, four-digit year (example: 15/12/1997)
MM/DD/YYYY	Two-digit month, separator, two-digit day, separator, four-digit year (example: 12/15/1997)
HHMM	Two-digit hour, two-digit minutes (example: 0330 for thirty minutes past three o'clock)
HHMMSS	Two-digit hour, two-digit minutes, two-digits seconds (example: 033045 for thirty minutes and forty-five seconds past three 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)
DDMMYY	Two-digit day, two-digit month, last two-digits of year (example: 151297)
MMDDYY	Two-digit month, two-digit day, last two-digits of year (example: 121597)
DDMMYYYY	Two-digit day, two-digit month, four-digit year (example: 15121997)
MMDDYYYY	Two-digit month, two-digit day, four-digit year (example: 12151997)
YYDDD	Last two-digits of year, three-digit Julian day (example: 97349 for the 349th day of 1997)
DDYY	Three-digit Julian day, last two digits of year (example: 34997)
YYYYDDD	Four-digit year, three-digit Julian day (example: 1997349)
DDYYYY	Three-digit Julian day, four-digit year (example: 3491997)
YY/DDD	Last two digits of year, separator, three-digit Julian day (example: 97/349)
DDD/YY	Three-digit Julian day, separator, last two digits of year (example: 349/97)

YYYY/DDD	Four-digit year, separator, three-digit Julian day (example: 1997/349)
DDD/YYYY	Three-digit Julian date, separator, four-digit year (example: 349/1997)
MONTH	The month (example: December)
DAY	The day of the week (example: Friday)

5. Verify that the date or time format selected in the EDI Format list is the correct format that is written to the EDI file.



A Dt/Tm format must be designated for Dt/Tm-type elements in the EDI Format box, even for hidden fields.

6. Click OK to exit the Field Properties dialog.

Customizing Your Display

The Forms Integration subsystem allows you to customize the display of forms in several different ways:

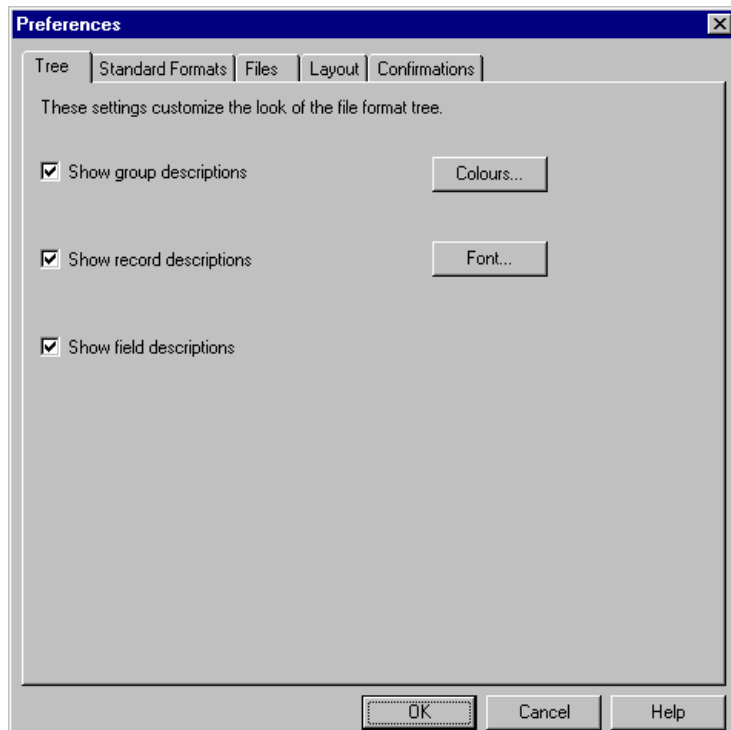
- Customizing global display options
- Customizing colours (global option)
- Customizing fonts (global option)
- Customizing display of EDI File Format and Layout Windows

The form display options can be set or changed at any time.

Customizing Global Display Options

Complete the following steps to customize global (for all forms) display options:

1. Select Preferences... from the Options menu to display the Preferences dialog (the Tree tab is displayed by default). The Preferences dialog is a property sheet that enables you to set global defaults for Gentran:Director.



2. To turn on the default display of group, record (segment), and field (element) descriptions, click the check box in front of each in the List section of the dialog (so a checkmark is displayed in the check box). To turn off the default display of group, record (segment), and field (element) descriptions, click the appropriate check box again to clear the box.

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 see the entire form if the descriptions are not displayed.



You may want to experiment with shrinking the size of the font for the form (see *Customizing Global Fonts* on page 63) before you turn off the display of descriptions.

3. Click OK to save changes and exit the Preferences dialog.

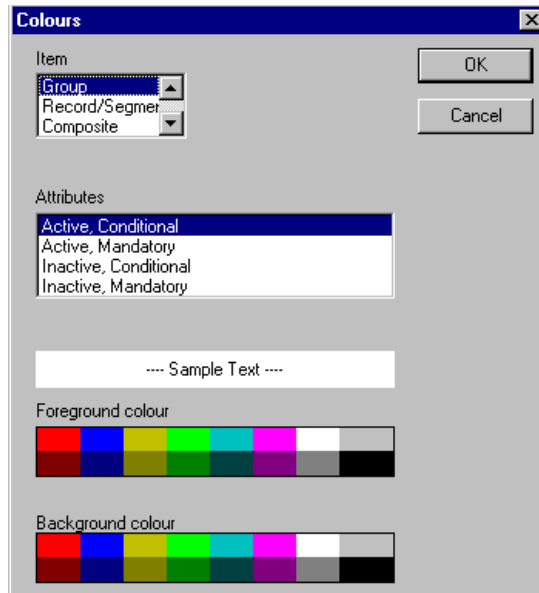
Customizing Global Colours

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

Complete the following steps to customize colours for all forms:

1. Select Preferences... from the Options menu to display the Preferences dialog (the Tree tab is displayed by default). The Preferences dialog is a property sheet that enables you to set global defaults for Gentran:Director.

2. Click COLOURS... to display the Colours dialog.



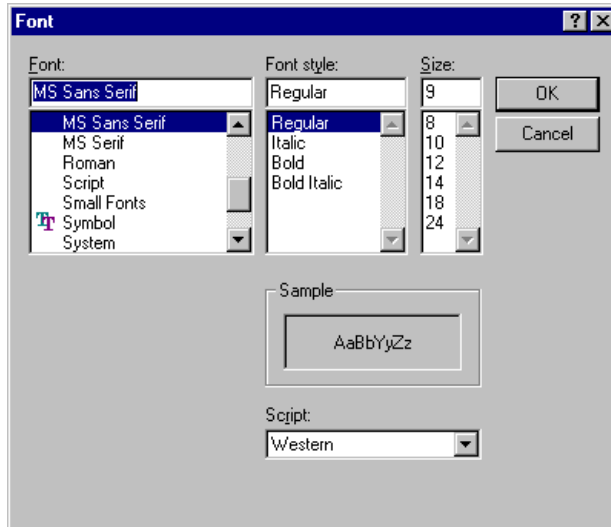
3. Select the type of form component from the Item list (Group, Record/Segment, Composite, Field/Element).
4. Further define the type of form component by choosing whether it is active or inactive and conditional or mandatory from the Attributes list. The valid selections are:
 - Active, Conditional
 - Active, Mandatory
 - Inactive, Conditional
 - Inactive, Mandatory
5. Repeat steps 3 and 4 for each combination of Items and Attributes for which you want to define colours.
6. Click OK to globally define the selected colours for all forms.

Customizing Global Fonts

The Font feature enables you to globally change the font that is used in the display of all forms. The default font that Gentrان:Director uses is a Sans Serif 9 point. This feature gives you the flexibility to shrink the font if you need to view more of the form on your monitor, enlarge the font, or change the type and style to be more easily readable by you.

Complete the following steps to customize the display font for all forms:

1. Select Preferences... from the Options menu to display the Preferences dialog (the Tree tab is displayed by default). The Preferences dialog is a property sheet that enables you to set global defaults for Gentrان:Director.
2. Click FONT... to display the Font dialog.



3. Select the type of font from the Font list. The default is MS Sans Serif.
4. Select the style from the Font Style list. The default is Regular.
5. Select the point size from the Size list. The default is 9.
6. You can preview the changes to the font in the Sample box. Click OK to make the global font change to all forms.

Customizing Display of EDI File Format and Layout Windows

Gentran:Director enables you to choose how you want the EDI File Format and Layout Windows displayed via these three options.

- The Maximize Tree function maximizes the EDI File Format Window.
- The Maximize Layout function maximizes the Layout Window.
- The Rearrange function toggles the display of the EDI File Format Window and the Layout Window from left/right to above/below.

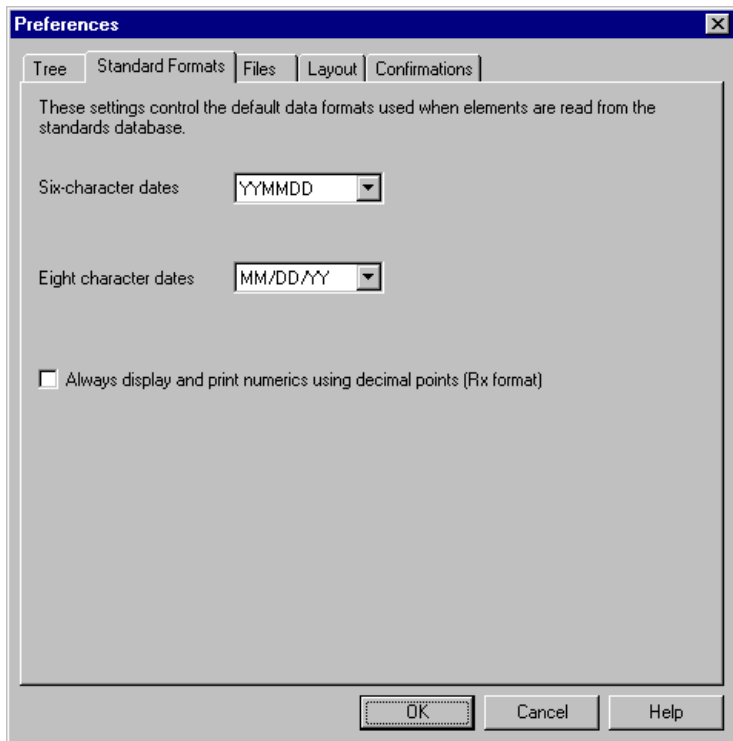
Additionally, Gentran:Director enables you to customize the measurement system that the vertical and horizontal rulers use. The default measurement system is inches, but you can toggle the display to centimeters by double-clicking the yellow square where the vertical and horizontal rulers meet.

Setting Default Date Format

You typically establish the date format to be used for all date elements *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. This global option changes the default date format for all forms. However, the format of the existing date elements does not change; the default is used only for new forms.

Complete the following steps to establish the default date format:

1. Select Preferences... from the Options menu to display the Preferences dialog. The Preferences dialog is a property sheet that enables you to set global defaults for Gentran:Director.
2. Click the Standard Formats tab to display the default date formats that are used when elements are read from the standards database.








3. Select the appropriate six-character and eight-character default date formats from the Six-character dates and Eight character dates lists.
4. Click OK to accept the default date formats for Six- and Eight-character date boxes and exit the Preferences dialog.



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.

Selecting Which Form Components to Use

The form components for an EDI File are the following:

-  EDI File
-  Segment
-  Loop (Group)
-  Composite Element (EDIFACT and TRADACOMS standards *only*)
-  Element

The specific EDI form components that you use depend on the type of form you are creating, the standard, version, transaction set (document), and release (TRADACOMS only) selected, and which segments and elements your company requires. We recommend that you determine which segments and elements you are using before starting this step.

When Gentran:Director generates the EDI document, 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. 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 generating the form, the system does not process groups, segments, composites, and elements 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 the form.

Complete the following steps to activate groups, segments, composites, and elements:



1. Click the **ACTIVATE** button on the Main Toolbar.
2. Open (double-click) each group that contains groups or segments that need to be activated.



As an alternative to opening each form component (steps 2, 5, and 7), you can select **Expand All from the **View** menu to open every form component. You can also select a form component and select **Expand Branch** from the **View** menu to open that form component.**

3. Click each inactive group that you need to use to activate it.



If you accidentally click a group, segment, composite, or element that you did not mean to activate, click the item with the right mouse button to access the shortcut menu. Select Deactivate from the shortcut menu.

4. Click each inactive segment that you need to use to activate it.
5. Open (double-click) each segment that contains composites or elements that need to be activated.
6. Click each inactive composite that you need to use to activate it.
7. Open (double-click) each composite that contains elements that need to be activated.
8. Click each inactive element that you need to use to activate it. When you activate an element for a print translation object, the word “Printed” is displayed before that element’s name. When you activate an element for a screen entry translation object, either the words “Edit Box” or “Drop Down” are displayed before that element’s name. “Edit Box” indicates that the element is formatted as a field. “Drop Down” indicates that the element is formatted as a list because you applied a standard rule that allows a selection of multiple items.
9. Once you have activated all the necessary groups, segments, and elements, click the **ACTIVATE** button on the Main Toolbar to turn activation mode off.

Cutting, Copying, and Pasting

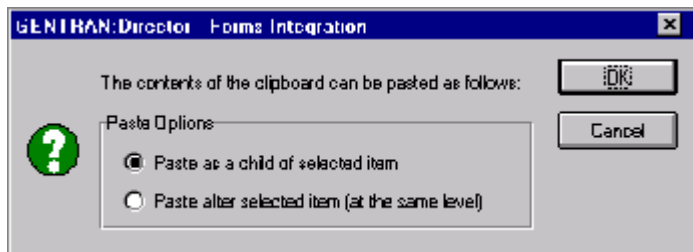
You can cut or copy a single form component (loop, segment, composite, or element) and paste it in another location in the form. Copied form components retain all the information of the original form component. If the copied form component contains subordinate form components (like a segment contains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

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

Complete the following steps to cut, copy, and paste a form component:

1. Select the form component that you want to cut or copy.
2. Click the Copy button on the Main Toolbar.
3. If you are pasting the form component in another form, open that form, if it is not already open. Please see *Opening a Form* on page 38 for more information on opening a form.
4. Select the form component that you want the cut or copied selection pasted *after*.
5. Click the PASTE button on the Main Toolbar to paste the contents of the Clipboard.

If the form component that you selected is a group, Gentran:Director 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.



Complete the following steps to cut, copy, and paste text from one box to another or in the extended rules dialogs:

1. Select the text that you want to cut or copy.
2. Click the CUT or COPY button on the Main Toolbar.
3. If you are pasting the text in another form, open that form, if it is not already open. Please see *Opening a Form* on page 38 for more information on opening a form.
4. Position the cursor where you want the cut or copied selection pasted *after*.
5. Click the PASTE from the Main Toolbar to paste the contents of the Clipboard.

Setting up the Document Name and Reference

The Document Name and Reference makes the identification of a document created by the form easier in Gentran:Director. The Document Name and Reference are two recommended fields that you can specify in the Forms Integration subsystem. If you specify either, the corresponding field in Gentran:Director document browsers (In Documents, ?In Documents, Out Documents, ?Out Documents, Workspace, In Drawer, Out Drawer, Send Queue, Interchanges) is populated.

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

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

- Change the element’s data type to “String” on the Field Properties dialog.
- OR
- Set up an extended rule to update the document record.



If you change a Dt/Tm or Numeric element's data type to “String,” you lose data type compliance checking for the original type.

Complete the following steps to set up the Document Name:

1. We recommend that you select a non-recurring element from the header or trailer segment. The element that you select varies for each type of document. The selected element should be something that will help identify this document meaningfully in Gentran:Director.

For example:

- For an Invoice this element is typically the Invoice Number.
- For a Purchase Order this element is typically the Purchase Order Number.

2. Use the right mouse button to click the element. Select Properties... from the shortcut menu. The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input checked="" type="checkbox"/> Mandatory	1	8	String	X	X

3. If the Advanced section of the dialog is not currently displayed, click ADVANCED>> to display the Advanced section.
4. Click the Use Standard Rule check box.
5. Click the Update option.
6. From the Table list, select Document Record.
7. From the Field list, select Document Name.
8. Click OK to set up the Document Name.

Complete the following steps to set up the Reference:

1. Select an element from the first segment of the EDI side of the document. The element that you select varies for each type of document. The selected element should be something that will help identify this document meaningfully in Gentran:Director.

For example:

- For an Invoice this element is typically the Purchase Order Number.
 - For a Purchase Order this element is typically the Contract Number.
2. Use the right mouse button to click the element you chose. Select Properties... from the shortcut menu. The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	1	10	String	X	X

3. Click the Use Standard Rule check box.
4. Click the Update option.
5. From the Table list, select Document Record.
6. From the Field list, select Reference Data.
7. Click OK to set up the Reference.

Customizing the Properties of Form Components

You can customize the properties of each type of form component (segment, group, element, and composite) on the Properties dialog for that component.

This chapter explains how to customize the properties of form components, including the following topics:

- Customizing the Properties of a Segment or Looping Structure
- Using LS and LE Segments
- Customizing the Properties of an Element
- Customizing the Properties of a Composite Element

Customizing a Segment or Group

You can modify the properties of the segments and groups by changing their properties on the appropriate Properties dialog, and by using the promote, split, copy, cut, and paste functions.

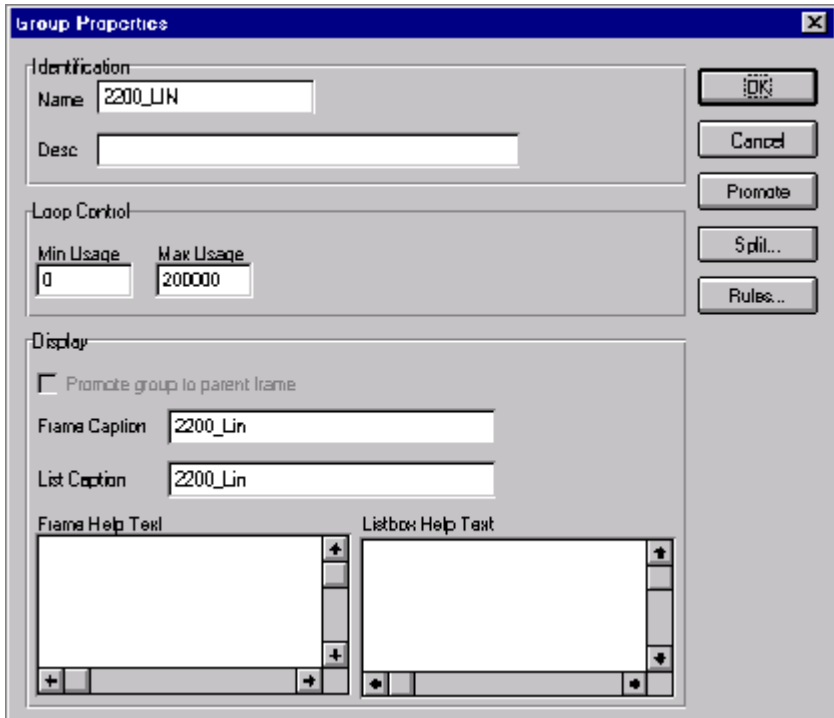
Customizing Group Properties

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).

Complete the following steps to modify the properties of a group:

1. Click the group you want to modify with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu.

The Group Properties dialog is displayed.



2. If you want to modify the group name, type the segment loop identification code in the Name box.



Do not use spaces or dashes (-) in the Name box. You can use the underscore (_) to separate words.

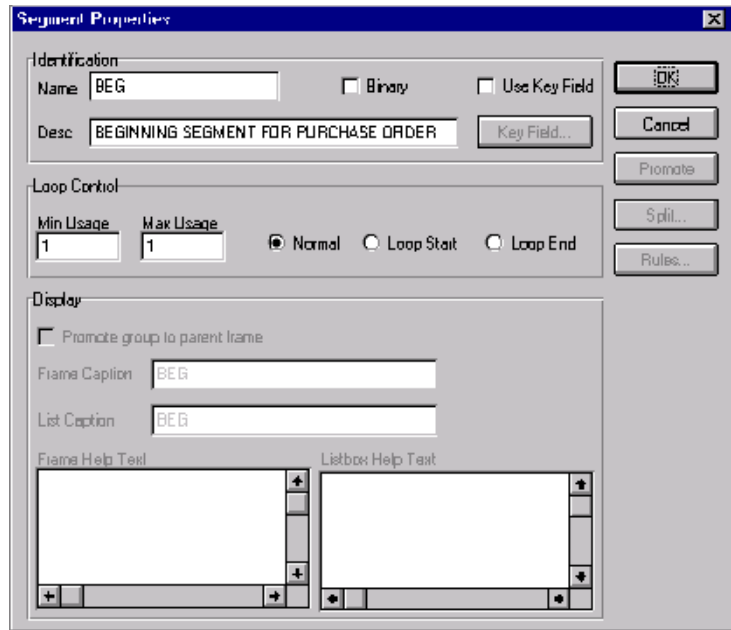
3. If you want to modify the group description, type the new description in the Desc box. This box is used to provide a brief explanation of the loop.
4. In the Min Usage box, type the minimum number of times the loop must be repeated. For a conditional loop, the minimum usage should always be “0” (zero). For a mandatory loop, the minimum usage should be “1” or greater.
5. In the Max Usage box, type the maximum number of times the loop can be repeated.
6. Click OK to save the changes to the group.

Customizing Segment Properties

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. Please see *Using Loop Start and Loop End Segments* on page 80 for more information on using loop start and loop end segments.

Complete the following steps to modify the properties of a segment:

1. Click the segment you want to modify with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu to display the Segment Properties dialog is displayed.



2. If you need to modify the name of the segment, type the segment identification code in the Name box.



If a segment or element occurs more than once in a form 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 form.

3. If you want to modify the description of the segment, type a new description in the Desc box. This box is used to provide a brief explanation of the segment that allows you to differentiate it from similar segments.
4. In the Min Usage box, type the minimum amount of times the segment must repeat.



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.”

5. If you want the segment to repeat (loop), type the maximum amount of times it can repeat in the Max Usage box.
6. Unless you want to indicate that the segment is the beginning of a loop (Loop Start) or the end of a loop (Loop End), accept the default of Normal for the Normal/Loop Start/Loop End options. These options are defined as follows:
 - Normal = This segment is in the loop but is not the beginning or ending segment
 - Loop Start = This segment marks the beginning of the loop
 - Loop End = This segment marks the end of the loop



If you need to modify the display properties of the segment, you can do so in the Display section of the Segment Properties dialog. Please see *Chapter 6, Customizing Screen Entry Translation Objects* on page 205 for more information on modifying display properties.

7. Click OK to save the changes to the segment.

Using Loop Start and Loop End Segments

Certain EDI standards use Loop Start (LS) and Loop End (LE) segments. LS and LE segments specify the explicit start and end of loops.

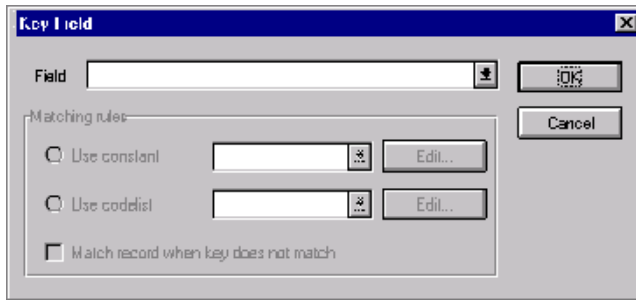
Defining a Loop Start

Complete the following steps to define an LS segment for a form:

1. Click the LS segment with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu to display the Segment Properties dialog.

2. Verify that the Loop Start option in the Looping section of the dialog is selected.
3. Click the Use Key Field check box to specify that the LS segment matches a selected constant.

- Click **KEY FIELD...** to access the Key Field dialog.



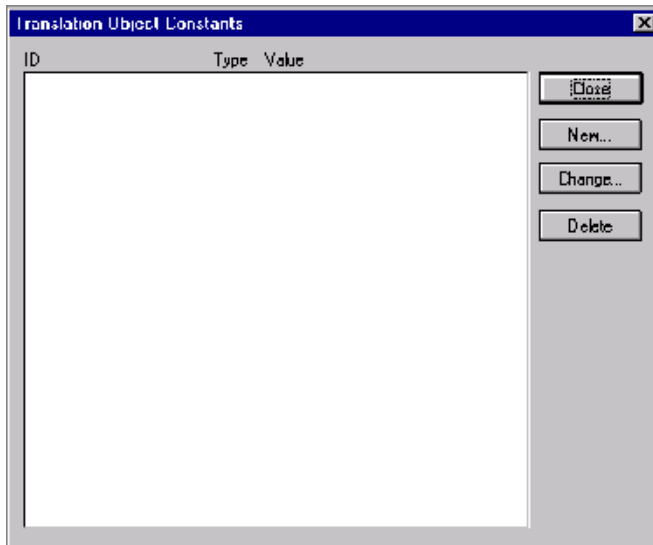
- From the Fields list, select the Loop Identifier Code. This list contains all the elements that are contained in the segment.

This box 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.

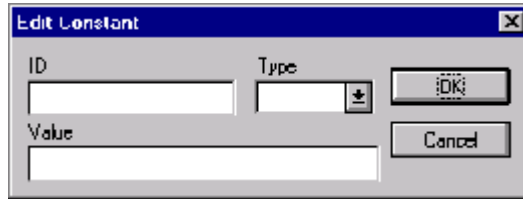
- The Matching Rules section enables you to access all the literal constants and code lists currently defined for this form.

Click the Use constant option.

- Click **EDIT...** (at the right of the Use constant list) to access the Translation Object Constants dialog.



8. Click NEW... to access the Edit Constant dialog.



9. In the ID box, type the literal constant identifier.
10. From the Type list, select "String." This is the category of this literal constant.
11. In the Value box, type the value of the literal constant.

For a screen entry form, this is the loop identifier code that you expect to receive from your trading partner.

12. Click OK to add the constant to the system.
13. Select the constant you created from the list. The system matches the selected constant against the loop identifier code.
14. Click OK to exit the Key Field dialog.
15. Click CLOSE to exit the Segment Properties dialog.

Defining a Loop End

Complete the following steps to define an LE segment for a form:

1. Click the LE segment with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu to display the Segment Properties dialog.

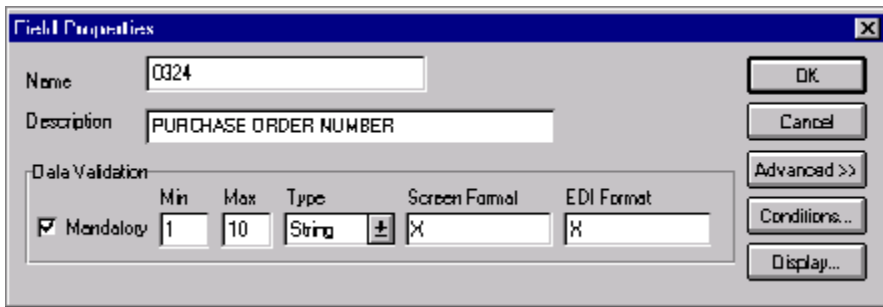
2. Verify that the Loop End option in the Looping section of the dialog is selected.
3. Click the Use Key Field check box to specify that the LE segment matches a selected constant.
4. Click KEY FIELD... to access the Key Field dialog.
5. From the Fields list, select the loop identifier code. This list contains all the elements that are contained in the segment. This box 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.
6. The Matching Rules section enables you to access all the literal constants and code lists currently defined for this form. Select the constant you created for the LS segment from the list. The system matches the selected constant against the loop identifier code.
7. Click OK to exit the Key Field dialog.
8. Click CLOSE to exit the Segment Properties dialog.

Customizing an Element

You can modify the properties of the elements by changing their properties on the Field Properties dialog, and by defining and modifying element relational conditions.

Complete the following steps to modify the properties of an element:

1. Double-click the element you want to modify to access the Field Properties dialog.



2. To change the name of the element, type the new name in the Name box. The name is typically the element sequence number.



Do not use spaces or dashes (-) in the Name box. You can use the underscore (_) to separate words.

3. To change the description of the element, type the new description in the Description box. The description is used to provide a brief explanation of the element that allows you to differentiate it from similar elements.
4. If you want to designate the element as “mandatory” (must be present), click the Mandatory check box.
5. To change the minimum number of characters that is acceptable for the element, type the minimum length in the Min box. If the data is less than the minimum length, a compliance error is generated when translation occurs.
6. To change the maximum length of the element, type the maximum length in the Max box.

7. From the Type list, select the type of the element. Valid values are:
 - String = alphanumeric element
 - Number = numeric or real element
 - Dt/Tm = date or time element
8. From the Screen Format list, select how the data element is formatted on the form. The choices for this box depend on the type of the element you selected from the Type list. If you choose Number or Dt/Tm in the Type box, you can select the data format from the Format list. If you selected String from the Type box, you should type a syntax token to denote that this element must be formatted as the specified syntax token dictates (the default syntax token is “X”). Please see *Using Syntax Tokens* on page 53 for more information on defining and using syntax tokens.
9. From the EDI Format list, select how the data element is formatted in the EDI file. The choices for this box depend on the type of the element you selected from the Type list. If you choose Number or Dt/Tm in the Type box, you can select the data format from the Format list. If you selected String from the Type box, you should type a syntax token to denote that this element must be formatted as the specified syntax token dictates (the default syntax token is “X”). Please see *Using Syntax Tokens* on page 53 for more information on defining and using syntax tokens.
10. Click OK to save the changes to the element.

Defining and Modifying Relational Conditions

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

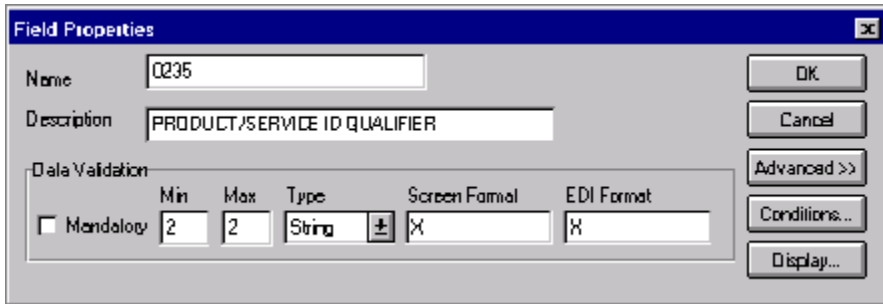
You can also view the conditional relationships between elements, as provided by the standard.



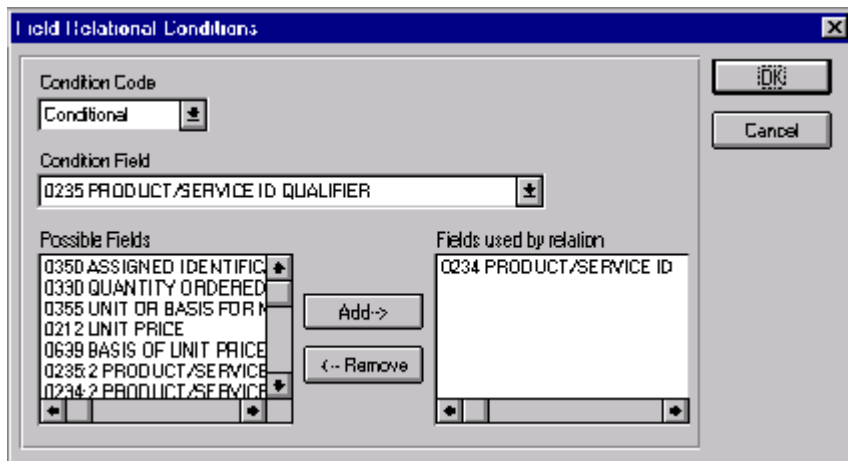
The system enables you to edit EDI conditional relationships, but if you do, you may alter the compliance checking criteria. We recommend that you view EDI conditions ONLY.

Complete the following steps to define, modify, and view element relational conditions:

1. Double-click the element to access the Field Properties dialog.



2. Click CONDITIONS... to access the Field Relational Conditions dialog.



This dialog enables you to view, modify, or define element relational conditions. If you want to simply view an element relational condition, do so and click OK to exit the dialog. Continue with the following steps to define or modify conditions.

3. Select the element connection condition from the Condition Code list. Valid values are:
 - **Paired/Multiple** – if any of the specified elements are present then all elements must be present
 - **Required** – at least one of the specified elements must be present
 - **Exclusion** – no more than one of the specified elements may be present

- **Conditional** – if the first Condition element is present, the rest of the elements must also be present
 - **List Conditional** – if the first Condition element is present, at least one of the specified elements must also be present
4. Select the first element from the Condition Field list. This is the element on which the conditional relationship hinges if you chose Conditional or List Conditional from the Condition Code list.



The Condition Field list is only active if you chose either Conditional or List Conditional from the Condition Code list.

5. The Possible Fields list contains all the elements in the translation object that are valid to be used in a condition at this point. Highlight an element or elements in the Possible Fields list and click the **ADD-->** button to move the elements to the Fields used by relation list, to include the elements as a part of the conditional relationship.

To highlight a block of adjacent items, click the first item and then press the **SHIFT** key and click the last item. To highlight several items that are not adjacent to each other, press the **CTRL** key and click the items.

6. The Fields used by relation list contains the elements that you selected to be a part of the conditional relationship. To remove the elements from the conditional relationship, highlight an element or elements in the Fields used by relation list and click the **<--REMOVE** button to move the elements back to the Possible Fields list.
7. Click **OK** to add the conditional relationship to the element.

Customizing a Composite Element

You can modify the properties of a composite element changing the properties on the Composite Properties dialog. A composite is a data element that contains two or more component data elements or sub-elements. You can modify the composite name and description, and whether or not it is mandatory.

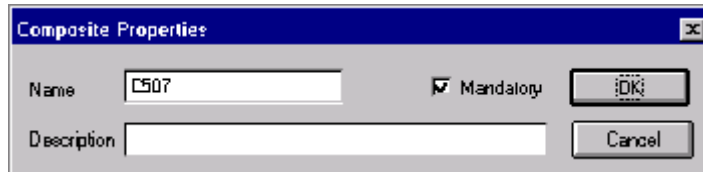


Composite elements are only used for EDI forms, for the EDIFACT, TRADACOMS, and ANSI X12 003030 and later standards *only*.

Complete the following steps to modify the properties of a composite:

1. Click the composite you want to modify with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu.

The Composite Properties dialog is displayed.



2. If you want to change the composite name, type the data element identification number in the Name box.



If a segment, composite, or element occurs more than once in a form 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 form.

3. Click the Mandatory check box to indicate whether or not this composite must be present. A checkmark means that the composite is mandatory.
4. If you want to change the composite description, type the new description in the Description box.
5. Click OK to save the changes to the composite.

Using Standard Rules

Gentran:Director provides you with standard rules that you can apply to elements. Standard rules give you access to functions that are necessary for operations that are less involved than extended rules. Extended rules enable you to use a Gentran:Director proprietary programming language to perform virtually any operation you require. Please see *Introducing Extended Rules* on page 166 for more information on extended rules.

Each of the standard rules are mutually exclusive (you can use only one on a particular element). This chapter defines the standard rules and provides examples (where appropriate) of when you may use them. This chapter includes the following topics:

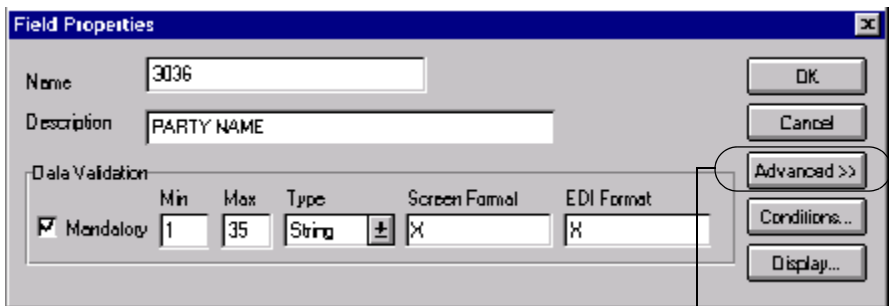
- Using the Select standard rule
- Using the Update standard rule
- Using the Constant standard rule
- Using the Use Accum standard rule
- Using the Loop Count standard rule
- Using the Use Code standard rule
- Using the System Date and Time
- Using Literal Constants and Qualifying Relationships
- Using Partner-Related Information in Forms
- Using Accumulators
- Using Code List Tables

Using the Select Standard Rule

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:Director Partner Editor). You can then map the elements in those tables to one or more elements in the data. The Select function uses the value of the current element to perform the selection. The results of the select can be displayed on the screen or printed on a report.

Complete the following steps to use the Select function:

1. Double-click an existing element, or create a new element (please see *Using an EDI File* for more information on creating elements). The Field Properties dialog is displayed.



Click Advanced >>

2. If the Advanced section of the dialog is not currently displayed, click ADVANCED >> to access this section, which enables you to define rules.
3. In the Advanced section, click the Use Standard Rule check box to display the rule options.

4. Verify that the Select option is selected.

The dialog box element definitions are as follows:

- Table** Select the table or table and key that you want to select. Valid values are:
- **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. You entered these fields on the Partner Definition dialog of the Partner Editor.
 - **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. You entered these fields on the Partner Definition dialog of the Partner Editor.
 - **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. You entered these fields on the Partner Definition dialog of the Partner Editor.
 - **Partner location by name** indicates that the name on the Locations dialog (for the current partner) of the Partner Editor is the key and enables you to map from any field in that table. You entered these fields on the Location Entry dialog of the Partner Editor.
 - **Partner lookup (activates Sub Table field)** indicates that the key is the partner lookup table name for this partner and enables you to map from any field in that table. You entered these fields on the Lookup Entry dialog of the Partner Editor.

- **Partner xref by my item (activates Sub Table field)** 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. You entered these fields on the CrossRef Entry dialog of the Partner Editor.
- **Partner xref by partner item (activates Sub Table field)** 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. You entered these fields on the CrossRef Entry dialog of the Partner Editor.
- **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. You entered these fields on the Partner Locations dialog of the Partner Editor.
- **Division location by name** indicates that the key is the name on the Locations dialog of the Partner Editor for the <Internal System User> and enables you to map from any field in that table. You entered these fields on the Locations dialog of the Partner Editor.
- **Division lookup (activates Sub Table field)** 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. You entered these fields on the Lookup Entry dialog of the Partner Editor.
- **Division xref by my item (activates Sub Table field)** 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. You entered these fields on the CrossRef Entry dialog of the Partner Editor.
- **Division xref by partner item (activates Sub Table field)** 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. You entered these fields on the CrossRef Entry dialog of the Partner Editor.
- **Document record** enables you to map from any of the fields within the current document record.
- **Generic envelope segment** enables you to map from any of the fields within the current envelope information, regardless of the EDI standard used. This function is normally not used unless you are familiar with the generic envelope table structure for a given standard.
- **Interchange** enables you to map from any of the fields within the current interchange record.
- **Group** enables you to map from any of the fields within the current group record.

Sub Table Type the sub table name (if appropriate) in this box.



The Sub Table box is active only if you select Partner Lookup, Partner xref by my item, Partner xref by partner item, Division lookup, Division xref by my item, or Division xref by partner item from the Table/Key box.

Exclusive Click the Exclusive check box if you want the system to generate an error if the Select does not find a valid table entry. The default is blank (do not generate an error if the Select is not successful).

Map From From this list, choose the element from the specified table entry from which you want to map the contents.



Entries appear in the Map From list *only* after you select a Table/Key.

Map To From this list, choose the element to which you want to map the contents of the Map From box. Each element from the side of the map with which you are currently working (input or output) is displayed in this list. A total of eight fields can be mapped using one Select rule.



Entries appear in the Map To list *only* after you select a Table/Key, and these entries are activated *only* after you highlight an entry in the Map From box.

Using the Update Standard Rule

The Update function enables you to update a specific field in a document record, envelope segment, interchange, group, current partner, or document, with the contents of the element.



Please note that this function updates the internal Gentran:Director 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!

Complete the following steps to use the Update function:

1. Double-click an existing element, or create a new element (please see *Using an EDI File* for more information on creating elements). The Field Properties dialog is displayed.

Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box to display the rule options.

- Click the Update option.

The dialog box element definitions are as follows:

Table/Key Select the table or key from the Table/Key list that the system updates with the contents of the current element. Valid values are:

- Document record (use to update the document name and reference)
- Generic envelope segment (advanced use only)
- Interchange (advanced use only)
- Group (advanced use only)
- Current partner (advanced use only)
- Document (advanced use only)

Field Select the element from the Field list that the system updates with the contents of the current element. The elements that are available in this list depend on the Table/Key selected.

Example

We recommend that you set up a Document Name for each form, to make the identification of a document created by this form easier in Gentran:Director. This allows you to differentiate between documents in the document browsers in Gentran:Director.

To set up a Document Name, you must select an element in the form that contains data that distinguishes the documents translated by this form. Then,

the Name column of the document browsers that contain this document contains the data from the selected element.



You can choose any element as the Document Name, but the element must only occur once in the document and must have a data type of “String.” For a Purchase Order, the document name is typically the Purchase Order Number.

Complete the following steps to set up the Document Name for an ANSI X12 Purchase Order:

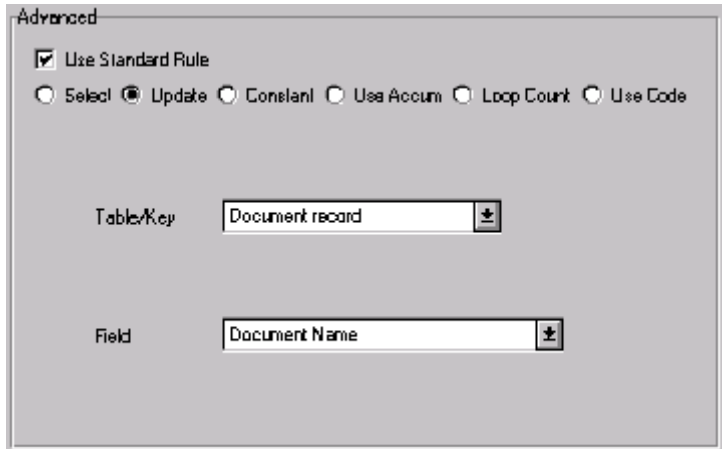
1. Double-click the BEG03 (0324) element to display the Field Properties dialog.

Data Validation						
	Min	Max	Type	Screen Format	EDI Format	
<input checked="" type="checkbox"/> Mandatory	1	10	String	X	X	

Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box to display the rule options.
3. Click the Update option.
4. From the Table/Key list, choose “Document record.” This indicates that you are updating the document record in Gentran:Director.
5. From the Field list, choose “Document Name.” This indicates that you are updating the Name column in the document browsers with the contents of this element.

The Advanced section of the Field Properties dialog should now look like the following:



Advanced

Use Standard Rule

Select Update Constant Use Accum Loop Count Use Code

Table/Key

Field

6. Click OK to set up the document name.

Using the Constant Standard Rule

The Constant option enables you to access the System Variable function and the Constant function. The System Variable function enables you to map the current date or time to the specified element.

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

Using System Variables

The System Variable function enables you to set a variable that maps either the current date or the current time to the selected element. The selected element must have a data type of “Date/Time.”

Complete the following steps to use a system variable:

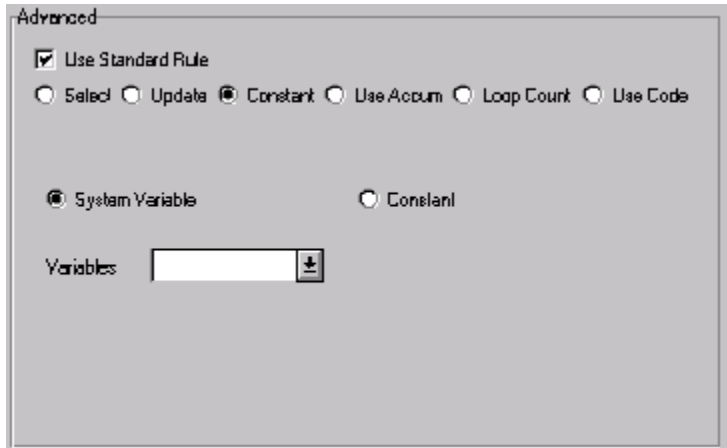
1. Double-click an existing element, or create a new element (please see *Using an EDI File Format* on page 20 for more information on creating elements). The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	1	8	String	X	X

Click Advanced>>

2. In the Advanced section, click the Use Standard Rule check box to display the rule options.

- Click the Constant option. The System Variable option is selected by default.



The screenshot shows a dialog box titled "Advanced" with the following options:

- Use Standard Rule
- Select
- Update
- Constant
- Use Accum
- Loop Count
- Use Code

Below these options, there are two radio buttons:

- System Variable
- Constant

At the bottom, there is a text field labeled "Variables" with a dropdown arrow button to its right.

The dialog box element definition is as follows:

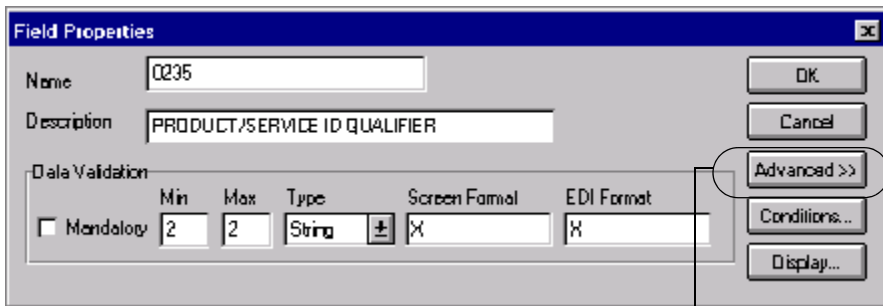
Variables Select current date or current time from the Variables list to map that variable to the element.

Using a Constant

Constants are used in forms to hold information that is needed later in the form in a conditional statement. Typically, EDI qualifiers are typically generated from constants (to indicate a qualifying relationship with another element).

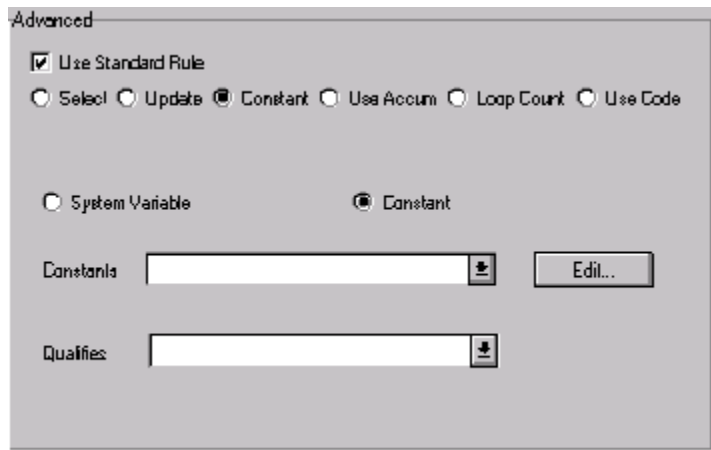
Complete the following steps to use a constant:

1. Double-click an existing element, or create a new element (please see *Using an EDI File* for more information on creating elements). The Field Properties dialog is displayed.



Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box to display the rule options.
3. Click the Constant option.
4. Click the second Constant option.



The dialog box element definitions are as follows:

- Constants** Select a constant from the Constants list. This constant is mapped to the current element. If the necessary constant is not present in the Constants list, you need to create it by using the Translation Object Constants dialog, which you can access by clicking the **EDIT...** button.
- Qualifies** Select an element from the Qualifies list to indicate that the current element qualifies the selected element (establish a qualifying relationship between the two elements). If the Qualifies element is not generated because of a lack of data, the constant is not moved to the current element.
- Edit...** This button enables you to access the Translation Object Constants dialog, which allows you to create, edit, and delete literal constants.

Example

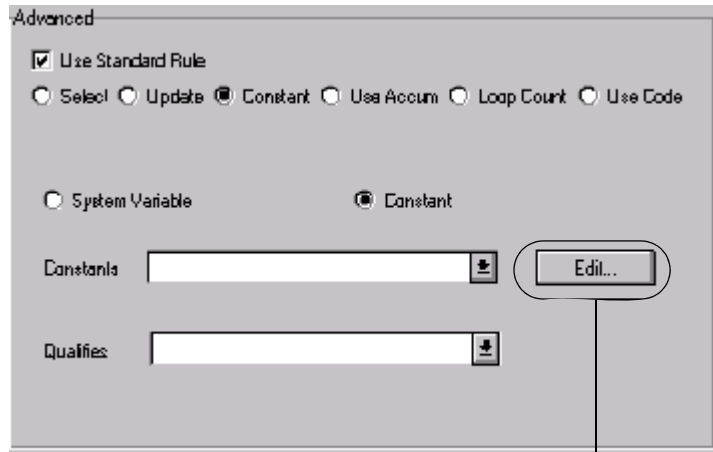
Literal constants are used by the system as a repository to store information that is used at a later point in the form. Typically, constants are used in a form to generate a qualifier. A qualifier is an element that has a value expressed as a code that gives a specific meaning to the function of another 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.

Complete the following steps to define a qualifying relationship:

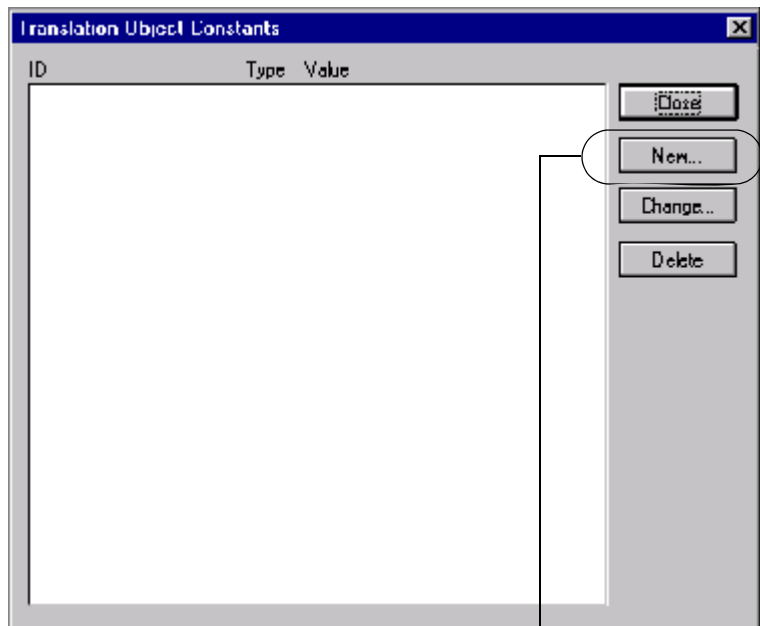
1. Double-click the Product/Service ID Qualifier that you want to use to further define (qualify) the Product/Service ID. The Field Properties dialog is displayed.
2. Click the Use Standard Rule check box.
3. Click the Constant option.

- Click the second Constant option.



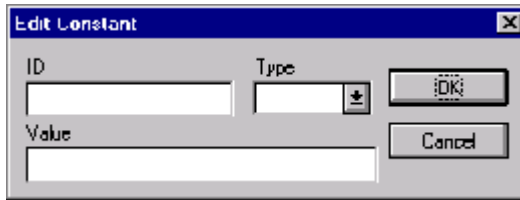
Click Edit...

- Click EDIT... to access the Translation Object Constants dialog.



Click New...

- Click NEW... to access the Edit Constant dialog.

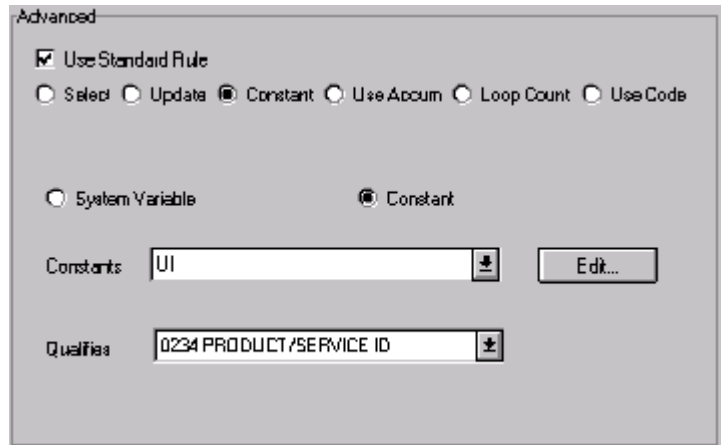


- In the ID box, type the literal constant identifier. This is typically a description of the element in which the constant is used. If you need to refer to the constant in an extended rule, you should use the data from this element.
- From the Type list, select String because the Product/Service ID Qualifier is formatted as a string data type. This indicates the category of the literal constant.
- In the Value box, type "UI" to indicate that the Product/Service ID element must contain the UPC Code. This is the value of the literal constant.
- Click OK to add the constant to the system.
- Click CLOSE to exit the Translation Object Constants dialog.
- From the Qualifies list, select Product/Service ID. This is the element that the Product/Service ID Qualifier qualifies.



The Qualifies list contains only the other *active* elements in the same segment as the qualifying element.

The Advanced section of the Field Properties dialog should now look like the following.



Advanced

Use Standard Rule

Select Update Constant Use Accum Loop Count Use Code

System Variable Constant

Constants:

Qualifies:



When you set a constant value for a qualifier field, you know that you never need to override the value in the field. In this instance, you would probably hide the field, as well. Please see *Hiding Fields* on page 221 for more information on hiding fields.

13. Click CLOSE on the Field Properties dialog and the qualifying relationship between the two elements is established.

Using the Use Accum Standard Rule

The Use Accum function gives you access to a set of numeric variables that you can manipulate via numeric operations, and then transfer to and from elements. This function enables you to add, change, or delete calculations for the element, including hash totals (used to accumulate numeric element values, i.e., quantity, price, etc.). This function also enables you to load the accumulated total into a control total element, and use accumulators.

Accumulators are used generally for counting the occurrences of a specific element or generating increasing or sequential segment or line item numbers.

Please see *Using Accumulators* for more information about defining accumulators.

Complete the following steps to use the Use Accum function:

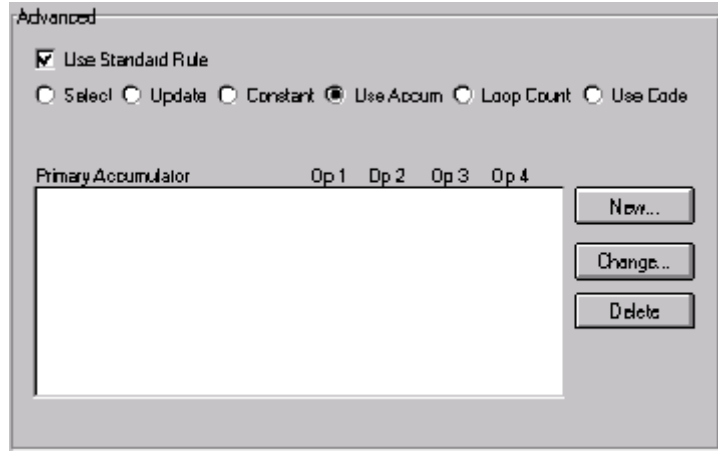
1. Double-click an existing element, or create a new element (please see *Using an EDI File* for more information on creating elements). The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	1	6	Number	N0	N0

Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box to display the rule options.

- Click the Use Accum option.



The dialog box element definitions are as follows:

Primary Accumulator The Primary Accumulator list contains all existing calculations that were created for this element.

New... Click this button to access the Edit Accumulator Entry dialog to create a new calculation for this element. Please see *Using Accumulators* for more information on defining accumulators in your form.

Change... Highlight a calculation in the Primary Accumulator list and click this button to access the Edit Accumulator Entry dialog to edit the selected calculation. Please see *Using Accumulators* for more information on defining accumulators in your form.

Delete Highlight a calculation in the Primary Accumulator list and click this button to delete the selected calculation.



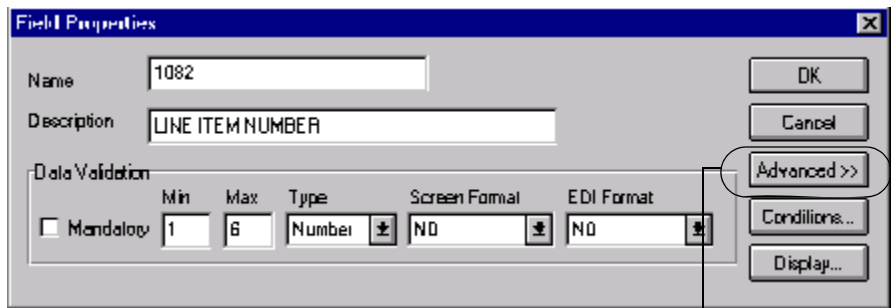
The selected calculation is deleted without warning.

Using the Loop Count Standard Rule

The Loop Count function enables you to obtain the value of the current loop count, if the element 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).

Complete the following steps to use the Loop Count function:

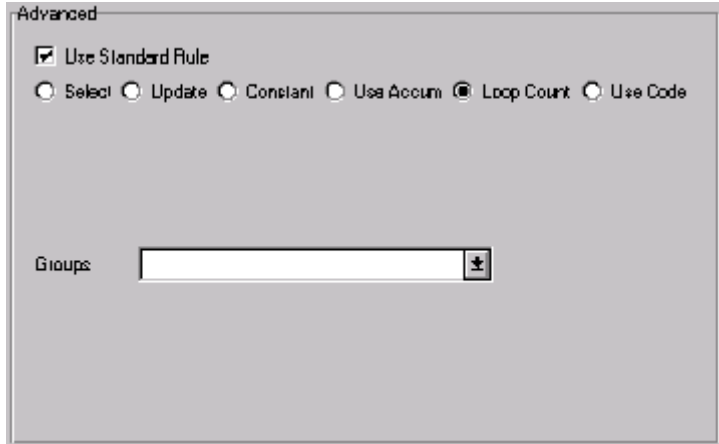
1. Double-click an existing element, or create a new element (please see *Using an EDI File* for more information on creating elements). The Field Properties dialog is displayed.



Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box to display the rule options.

3. Click the Loop Count option.



The screenshot shows a dialog box titled "Advanced" with a light gray background. At the top, there is a checked checkbox labeled "Use Standard Rule". Below it, there are six radio button options: "Select", "Update", "Constant", "Use Accum", "Loop Count", and "Use Code". The "Loop Count" radio button is selected. At the bottom of the dialog, there is a label "Groups:" followed by a text input field and a small square button with a downward-pointing arrow.

The dialog box element definition is as follows:

Groups Select the loop that you want to obtain the current value of the loop count. 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).

Using the Use Code Standard Rule

The Use Code function enables you to select an element from a predefined code table and specify whether or not a compliance error is generated if the element does not contain one of the values in the code table for screen entry translation objects. The Use Code function enables you to extract and print the description of a code for print translation objects.

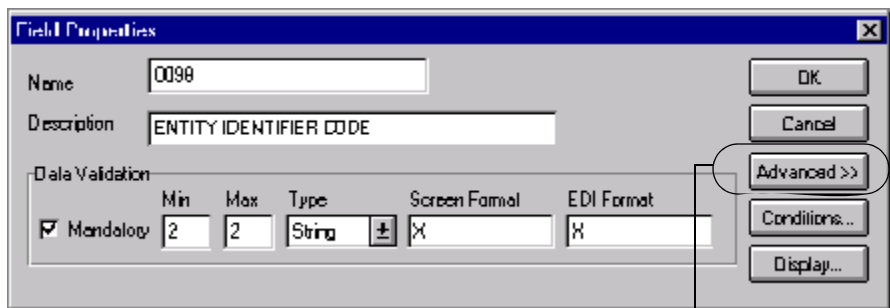
The Forms Integration subsystem enables you to create code tables to be used with the current form. You can set up code tables in the Forms Integration subsystem to function like the partner cross-reference and lookup tables in Gentran:Director. However, code tables that are set up in the Forms Integration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

An element 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*.

Please see *Using Code List Tables* on page 145 for more information on using code list tables.

Complete the following steps to use the Use Code function:

1. Double-click an existing element, or create a new element (please see *Using an EDI File* for more information on creating elements). The Field Properties dialog is displayed.



Click Advanced>>

2. In the Advanced section, click the Use Standard Rule check box to display the rule options.
3. Click the Use Code option.

The screenshot shows a dialog box titled "Advanced". At the top, there is a checked checkbox labeled "Use Standard Rule". Below it is a row of radio buttons with labels: "Select", "Update", "Constant", "Use Accum", "Loop Count", and "Use Code". The "Use Code" radio button is selected. Below the radio buttons, there is a text box labeled "Table" followed by a dropdown arrow and a button labeled "Edit Table...". Below that is an unchecked checkbox labeled "Exclusive". At the bottom, there is a text box labeled "Store Field" followed by a dropdown arrow.

The dialog box element definitions are as follows:

Table Select a table from the list. The Tables list contains all the code tables. If the necessary code table is not listed in this box, click **EDIT TABLE...** to load or create a code table.

Exclusive Click the Exclusive check box to specify that, for compliance reasons, the element *must* contain one of the codes from the specified table (nothing else is valid for that element). For example, if the element 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 element, the system flags it as an error.

Store Field Select an element from this list where you want the description of the code (that is used) to appear when the selection is made. For 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 Field list, the description for the code used is mapped to element "XX."

Edit Table... Click this button to access the Edit Code List dialog, which enables you to create and edit code list tables.

Example

For this example you need to have a list displayed during data entry, so the user can select your partner's customer product code. Therefore, you need to establish and use a code table with the Product Service ID *Qualifier*.

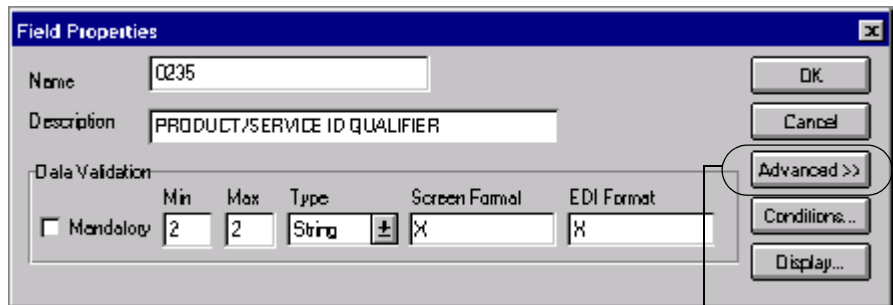
Complete the following steps to select from a code list table:

1. Load the code list table from the standard for the Product/Service ID Qualifier element. Please see *Loading a Code List Table from the Standard* on page 157 for more information on loading a code table.



Please note that 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.

2. Define a code list table for the Product/Service ID Qualifier element that only contains the code value "BP" (Buyer's Part Number). Please see *Defining a Code List* on page 146 for more information on defining a code table.
3. Double-click the element for which you need to validate data against a code table (Product/Service ID Qualifier). The Field Properties dialog is displayed.



Click Advanced >>

4. Click the Use Standard Rule check box.
5. Click the Use Code option.
6. From the Table list, select the code list table that the data in this element is validated against (0235).

7. If you need to specify that, for compliance reasons, the element *must* contain one of the codes from the specified table (nothing else is valid for that element), click the Exclusive check box.

For example, if the element 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 element, the system flags it as an error.

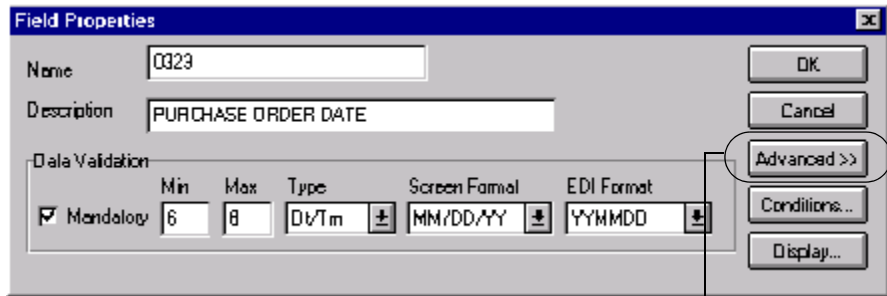
8. Click OK to add this standard rule to the element.

Using the System Date and Time

Gentran:Director enables you to update any element (that has a data type of “Dt/Tm”) with the current system date or time.

Complete the following steps to update a element with the current system date or time:

1. Double-click the desired element. The Field Properties dialog is displayed.



Click Advanced >>

2. Click the Use Standard Rule check box.
3. Click the Constant option. The Standard Rules section is displayed.
4. Click the System Variable option if it is not already selected (System Variable is the default).
5. From the Variables list:
 - Select “Current Date” if you want the element to be updated with the current system date.OR
 - Select “Current Time” if you want the element to be updated with the current system time.

Using Literal Constants and Qualifying Relationships

Literal constants are used by the system as a repository to store information that is used later in the form. You create the literal constant and name it. You can “hard code” (enter) a value that is loaded into the element that uses the constant. Or, you can store information from an element in a constant by using an extended rule. However, to *use* a constant (the information that is stored in it), you need to use a standard rule, as explained below.

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 defined using a standard rule, as explained below.

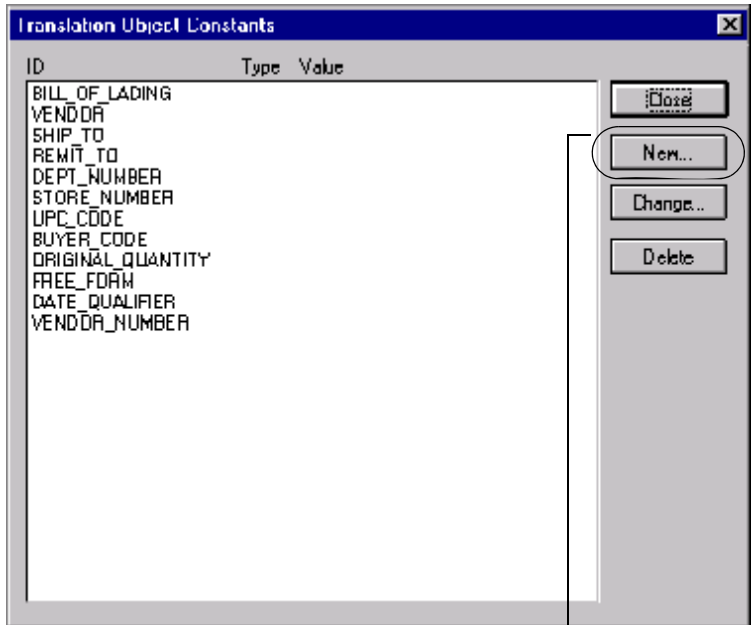
You can perform the following functions with literal constants and qualifying relationships:

- Defining Literal Constants
- Editing Literal Constants
- Deleting Literal Constants
- Mapping Literal Constants
- Generating Qualifiers

Defining Literal Constants

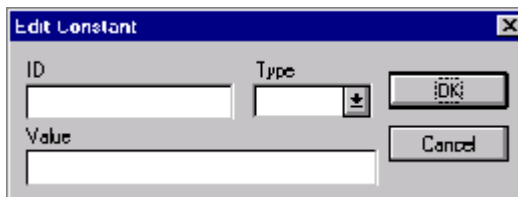
Complete the following steps to create a literal constant so you can use it to store information:

1. Select Constants from the Edit menu or the Main Toolbar. The Translation Object Constants dialog is displayed.



Click New...

2. Click NEW... to access the Edit Constant dialog.



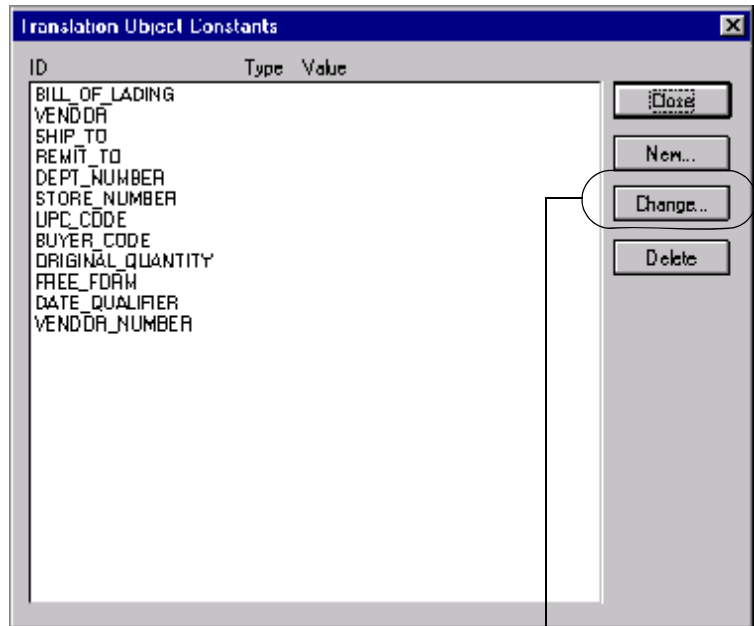
3. In the ID box, type the literal constant identifier. This is typically a description of the element in which the constant is used. If you need to refer to the constant in an extended rule, you should use the data from this box.

4. From the Type list, select the category of this literal constant.
 - **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)
5. In the Value box, type the actual constant expression. This is the value of the literal constant.
6. Click OK to add the constant to the system.
7. Click CLOSE to exit the Translation Object Constants dialog.

Editing Literal Constants

Complete the following steps to edit a literal constant:

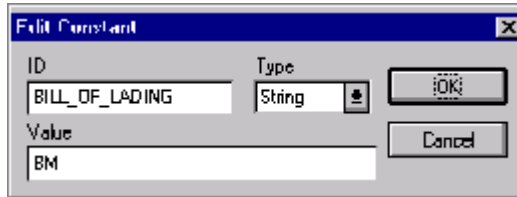
1. Select Constants from the Edit menu or the Main Toolbar. The Translation Object Constants dialog is displayed.



Select a constant and click Change...

2. Highlight the constant you want to modify.

- Click CHANGE... to access the Edit Constant dialog.

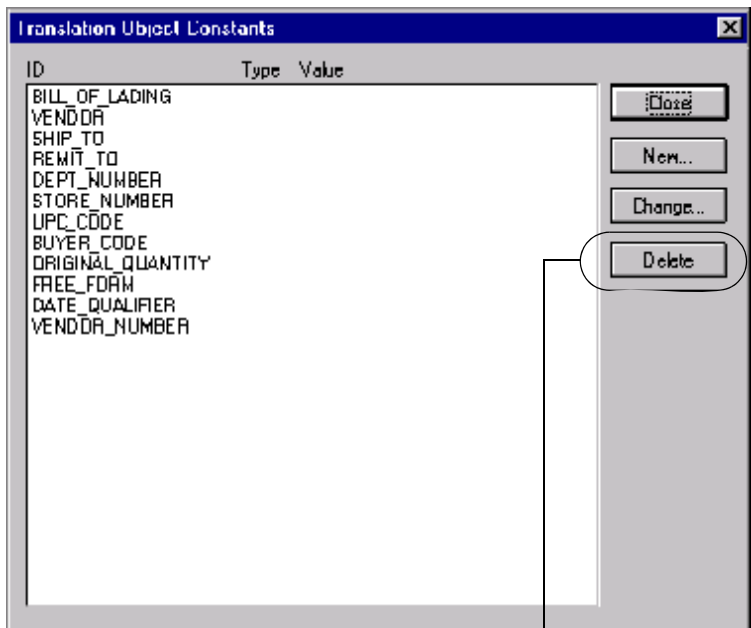


- Make the changes to the constant.
- Click OK to accept the changes.
- Click CLOSE to exit the Translation Object Constants dialog.

Deleting Literal Constants

Complete the following steps to delete a literal constant:

- Select Constants from the Edit menu or the Main Toolbar. The Translation Object Constants dialog is displayed.



Select a constant and click Delete

2. Highlight the constant you want to delete.
3. Click DELETE to remove the constant.



The constant is deleted without warning.

4. Click CLOSE to exit the Translation Object Constants dialog.

Using Literal Constants

Complete the following steps to use a constant in which you previously stored information using an extended rule:

1. Double-click the element in which you want to use the constant. The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	2	2	String	X	X

Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box.
3. Click the Constant option.

- Click the second Constant option.

- From the Constants list, select the constant that you want to use.
- Click OK and the data that was stored in the selected constant is loaded in the element.

Generating Qualifiers

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.

Complete the following steps to define qualifying relationships:

- Double-click the element that you want to use to further define (qualify) another element. The Field Properties dialog is displayed.
- In the Advanced section, click the Use Standard Rule check box.
- Click the Constant option.
- Click the second Constant option.
- From the Qualifies list, select the element that this element qualifies.



The Qualifies list contains only the other *active* elements in the same segment as the qualifying element.

6. Click OK and the qualifying relationship between the two elements is established.



When you set a constant value for a qualifier field, you know that you never need to override the value in the field. In this instance, you would probably hide the field, as well. Please see *Hiding Fields* on page 221 for more information on hiding fields.

Using Partner-Related Information in Forms

The Forms Integration subsystem allows you to use information from the Gentran:Director Partner Editor in your forms via standard rules. You can map information from your partner's profile in the Partner Editor to a selected element in the form. You can also map information from the Internal System Partner. 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

Using Information from the Partner Definition

This function enables you to load information from the Partner or Internal System Partner Definition into a screen entry or print translation object. For a screen entry translation object, the system displays a list which allows you to select the entry from the table. For a print translation object, the system prints the information on the report.

Complete the following steps to load information from the Partner or Internal System Partner Definition into a screen entry or print translation object:

1. Double-click the element to which the standard rule is applied. The Field Properties dialog is displayed.

Data Validation						
	Min	Max	Type	Screen Format	EDI Format	
<input checked="" type="checkbox"/> Mandatority	1	35	String	X	X	

Click Advanced>>

2. In the Advanced section, click the Use Standard Rule check box.
3. Click the Select option. The Standard Rules section is displayed.
4. From the Table box, select the key from which the system looks up the trading partner. Valid selections are:
 - **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) on the Partner Definition dialog of the Partner Editor.
 - **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) on the Partner Definition dialog of the Partner Editor.
 - **Partner by partner key** indicates that the key is the partner's Profile ID and enables you to map from one of three fields for this partner (Name, EDI Code, Alternate Code) on the Partner Definition dialog of the Partner Editor.
 - **Division** enables you to map from one of three fields (Name, EDI Code, Alternate Code) for the <Internal System User> (system partner) on the Partner Definition dialog of the Partner Editor.
5. If you want the system to generate an error if the lookup fails, click the Exclusive check box.
6. From the Map From list, select the field (Name, EDI Code, Application Code) from which you want the contents mapped.
7. From the Map To list, select the element to which the information from the Partner Editor is mapped.
8. Click OK to add the standard rule.

Using Information from Location Tables

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. For a screen entry translation object, the system displays a list which allows you to select the entry from the table. For a print translation object, the system prints the information on the report.

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

Complete the following steps to load information from a Location Table into a screen entry or print translation object:

1. Double-click the element to which the standard rule is applied. The Field Properties dialog is displayed.
2. In the Advanced section, click the Use Standard Rule check box.
3. Click the Select option. The Standard Rules section is displayed.
4. From the Table box, select the key that is used to look up the Partner Location entry. You are allowed to map from any box on the Locations dialog. Valid selections are:
 - **Partner location by name** indicates that the name on the Locations dialog (for the current partner) of the Partner Editor is the key and enables you to map from any field on that dialog.
 - **Division location by name** indicates that the key is the name on the Locations dialog of the Partner Editor for the <Internal System User> and enables you to map from any field on the Locations dialog.
5. If you want the system to generate an error if the lookup fails, click the Exclusive check box.
6. From the Map From list, select the field from which the contents are mapped. The fields you can select from are: 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 to which the information from the Partner Editor is mapped.
8. Click OK to add the standard rule.

Using Information from Lookup Tables

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. For a screen entry translation object, the system displays a list which allows you to select the entry from the table. For a print translation object, the system prints the information on the report.

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

Complete the following steps to load information from a Lookup Table into a screen entry or print translation object:

1. Double-click the element to which the standard rule is applied. The Field Properties dialog is displayed.
2. In the Advanced section, click the Use Standard Rule check box.
3. Click the Select option. The Standard Rules section is displayed.
4. From the Table box, select the key from which the system performs the look up of the trading partner. Valid selections are:
 - **Partner lookup (activates Sub Table box)** indicates that the key is the partner lookup table name for this partner and enables you to map from any field on the Lookup Entry dialog of the Partner Editor.
 - **Division lookup (activates Sub Table box)** indicates that the key is the partner lookup table name for the <Internal System User> partner and enables you to map from any field on the Lookup Entry dialog of the Partner Editor.
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, click the Exclusive 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 to which the information from the Partner Editor is mapped.
9. Click OK to add the standard rule.

Using Information from Cross-Reference Tables

Cross-reference tables are 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. Each Partner Profile and the Internal System Partner Profile can have many lookup tables associated with it. For a screen entry translation object, the system displays a list which allows you to select the entry from the table. For a print translation object, the system prints the information on the report.

To use information from a cross-reference (XREF) table, you must have created that table already in the Partner Editor. See the *Gentran:Director User's Guide* for more information on creating Partner cross-reference tables.

Complete the following steps to load information from a Cross-Reference (XREF) Table into a screen entry or print translation object:

1. Double-click the element to which the standard rule is applied. The Field Properties dialog is displayed.
2. In the Advanced section, click the Use Standard Rule check box.
3. Click the Select option. The Standard Rules section is displayed.
4. From the Table box, select the key from which the system performs the look up. For Cross-Reference Tables, valid selections are:
 - **Partner xref by my item** indicates that the key is your item (value) on the partner cross-reference table for this partner and enables you to map from any field on the CrossRef Entry dialog of the Partner Editor.
 - **Partner xref by partner** indicates that the key is your partner's item (value) on the partner cross-reference table for this partner and enables you to map from any field on the CrossRef Entry dialog of the Partner Editor.
 - **Division xref by my item (activates Sub Table box)** indicates that the key is your item (value) on the partner cross-reference table for the <Internal System User> partner and enables you to map from any field on the CrossRef Entry dialog of the Partner Editor.
 - **Division xref by partner item (activates Sub Table box)** indicates that the key is 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 on the CrossRef Entry dialog of the Partner Editor.
5. In the Sub Table box, type the name of the cross-reference table.
6. If you want the system to generate an error if the lookup fails, click the Exclusive check box.
7. From the Map From list, select the field (My Item, Partner 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 to which the information from the Partner Editor is mapped.
9. Click OK to add the standard rule.

Using Accumulators

The Use Accum function gives you access to a set of numeric variables that you can manipulate via numeric operations, and then transfer to and from elements. This function enables you to calculate sums, hash totals, value totals, counts, and generate HL level and parent codes, and other mathematical operations.

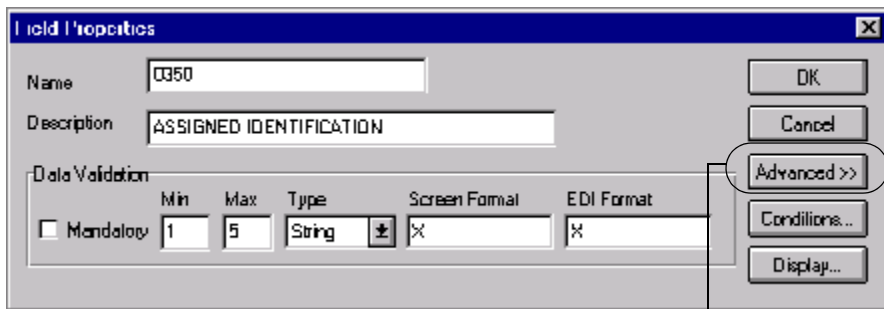
This section addresses three specific mathematical operations that you can perform with accumulators by illustrating an example of when you would use each.

Counting Line Items

In this example, you want an incremental count of the number of line items for a screen entry translation object, and you want to use that total value in the Number of Line Items Total element.

Complete the following steps to count line items and generate a control total for an ANSI X12 purchase order:

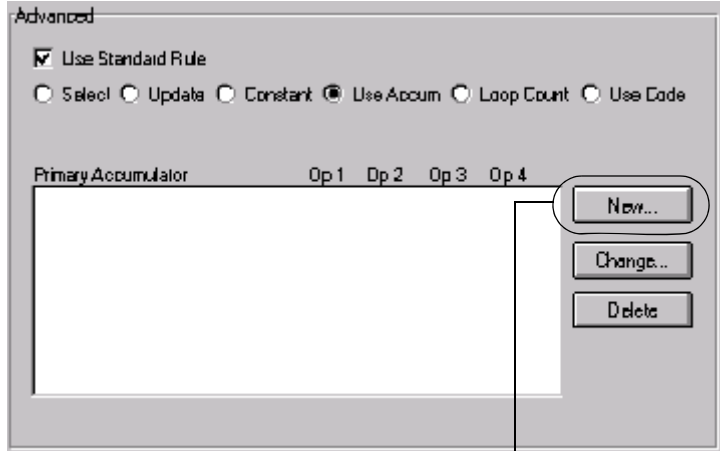
1. Double-click the P0101 element (in the P01 segment in the P01 group). This is the element that you typically use to count the line items. The Field Properties dialog is displayed.



Click Advanced>>

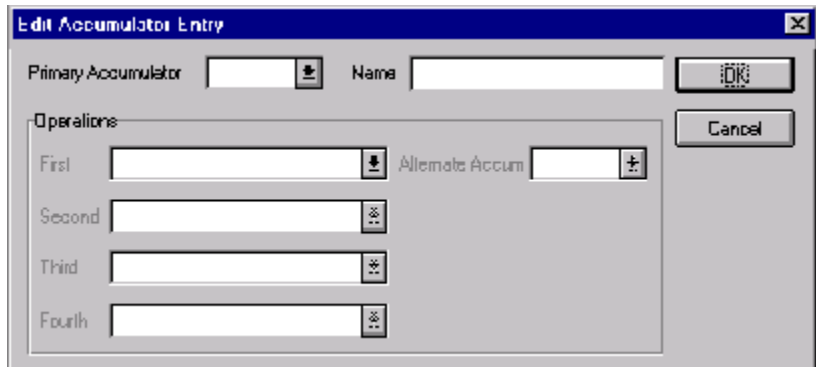
2. In the Advanced section, click the Use Standard Rule check box to display the rule options.

- Click the Use Accum option.



Click New...

- Click NEW... to access the Edit Accumulator Entry dialog to create a new calculation for this element.



- Select a Primary Accumulator from the list. 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.

For this example, select primary accumulator “0.”



There is *only one* set of accumulators for each form. 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 form and then use accumulator “0” again later in the form, 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.

6. In the Name box, type “Line Item Number.” This is a descriptive alias that enables you to differentiate what the accumulators you create are used for.
7. From the First list, select “Increment primary.” This is the first operation that the system performs. This specifies that the system increments the value in this accumulator by one for each iteration of the P01 group.



The First box is active only after you select a Primary Accumulator.

The valid selections for the First box are defined as follows:

- **Increment primary** adds “1” (one) to the contents of the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} + 1$).
- **Decrement primary** subtracts “1” (one) from the contents of the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} - 1$).
- **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., $\text{Primary} = (+/-)\text{Primary} + (+/-)\text{Element}$).
- **Hash sum in primary** adds the absolute value (does not take the positive or negative sign of the numbers into account) of the element to the contents of the Primary Accumulator (i.e., $\text{Primary} + \text{Element}$).
- **Load primary** loads the contents of the element into the Primary Accumulator (i.e., $\text{Primary} = \text{Element}$).
- **Use primary** loads the contents of the Primary Accumulator into the element (i.e., $\text{Element} = \text{Primary}$).
- **Zero primary** sets the value of the Primary Accumulator to zero (i.e., $\text{Primary} = 0$).
- **Multiply with primary** multiplies the element with the contents of the Primary Accumulator, and stores the result in the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} * \text{Element}$).

- **Divide by primary** divides the element with the contents of the Primary Accumulator, and stores the result in the Primary Accumulator (i.e., $\text{Primary} = \text{Element} / \text{Primary}$).
- **Divide primary by field** divides the contents of the Primary Accumulator with the element, and stores the result in the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} / \text{Element}$).
- **Modulo with primary** divides the contents of the element with the contents of the Primary Accumulator, and stores the *remainder* of that operation in the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} \% \text{Element}$).
- **Modulo with field** divides the contents of the Primary Accumulator with the contents of the element, and stores the *remainder* of that operation in the Primary Accumulator (i.e., $\text{Primary} = \text{Element} \% \text{Primary}$).
- **Negate primary** makes the contents of the Primary Accumulator negative (i.e., $\text{Primary} = \text{Primary} * -1$).



The only way to subtract the Primary Accumulator from the element is to “Negate” the Primary Accumulator and then use the “Sum in primary” operation to add the negative Primary Accumulator to the element.

- **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., $\text{Alternate} = \text{Primary}$).
- **Add primary to alternate** adds the contents of the Primary Accumulator to the contents of the Alternate Accum, and stores the result in the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} + \text{Alternate}$).
- **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 *remainder* of that operation in the Primary Accumulator (i.e., $\text{Primary} = \text{Primary} \% \text{Alternate}$).

- From the Second list, select “Use primary.” 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.

The values for the Second, Third, and Fourth boxes are listed above.



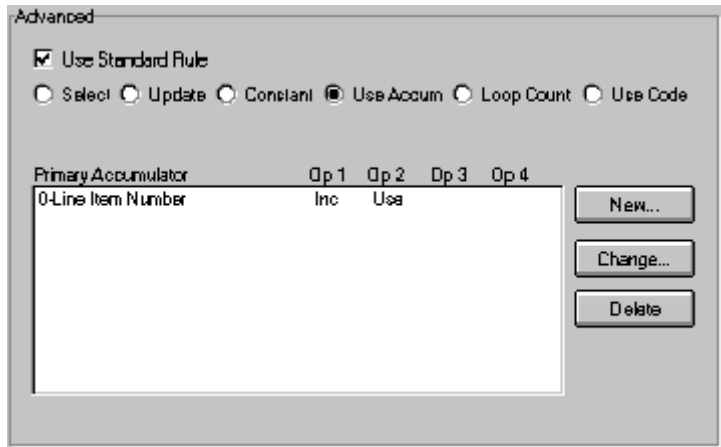
The Second box is active only after you select a First operation that does not involve the Alternate Accum. The Third box is active only after you select a Second operation. The Fourth box is active only after you select a Third operation.

The Edit Accumulator Entry dialog should now look like this:

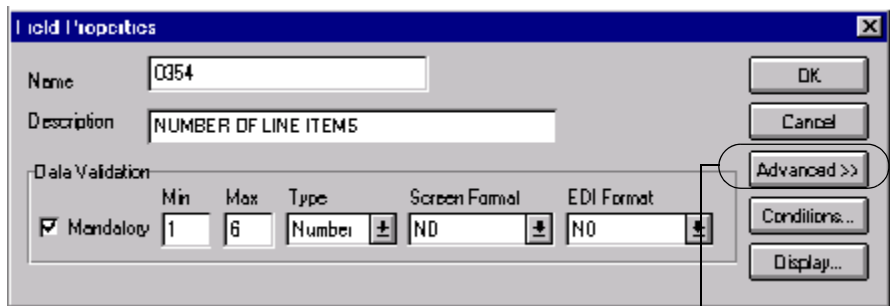
The screenshot shows the 'Edit Accumulator Entry' dialog box. The 'Primary Accumulator' field contains '0' and the 'Name' field contains 'Line Item Number'. The 'Operations' section has four rows: 'First' is 'Increment primary', 'Second' is 'Use primary', 'Third' is empty, and 'Fourth' is empty. There is an 'Alternate Accum' field next to the 'First' operation. 'OK' and 'Cancel' buttons are on the right.

- Click OK to add the accumulator.

The Advanced section of the Field Properties dialog should now look like this:



10. Click OK on the Field Properties dialog to add the standard rule to the P0101 element.
11. Double-click the CTT01 element (in the CTT segment). This is the element that typically contains the total number of line items. The Field Properties dialog is displayed.

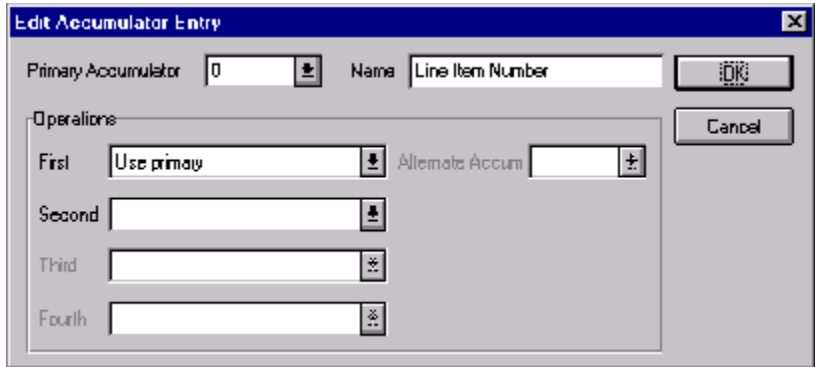


Click Advanced>>

12. In the Advanced section, click the Use Standard Rule check box to display the rule options.
13. Click the Use Accum option.
14. Click NEW... to access the Edit Accumulator Entry dialog to create a new calculation for this element.

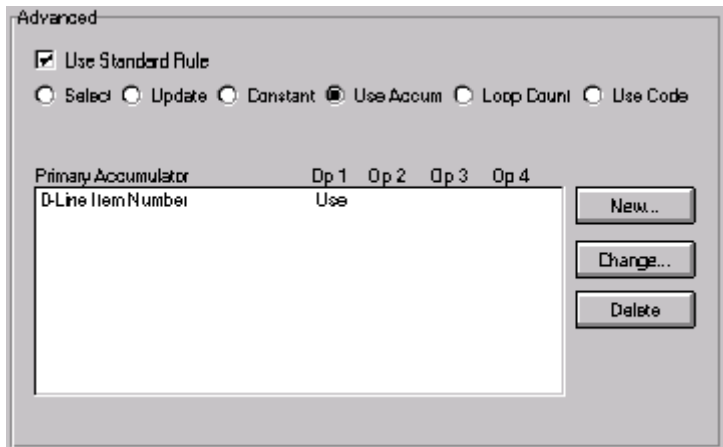
15. From the Primary Accumulator list, select primary accumulator “0.” This accumulator currently contains the total number of line items.
16. From the First list, select “Use primary” to specify that the system loads the current value of the accumulator into the CTT01 (Number of Line Items Total) element.

The Edit Accumulator Entry dialog should now look like this:



17. Click OK to add the accumulator.

The Advanced section of the Field Properties dialog should now look like this:



- Click OK on the Field Properties dialog to add the standard rule to the CTT01 element.

The CTT01 element now contains the total number of line items in the purchase order.



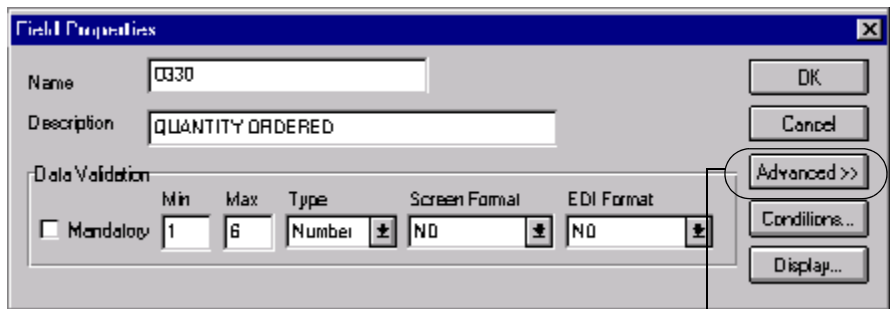
If you are adding this accumulator to a screen entry translation object, you would typically set the CTT01 field to display-only. Please see *Formatting Display-Only (Non-Editable) Fields* on page 220 for more information on formatting display-only fields.

Calculating Hash Totals

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 for a screen entry translation object.

Complete the following steps to count the quantity ordered and generate a hash total for an ANSI X12 purchase order:

- Double-click the P0102 element (in the P01 segment in the P01 group). This is the element that you typically use to count the line items. The Field Properties dialog is displayed.



Click Advanced >>

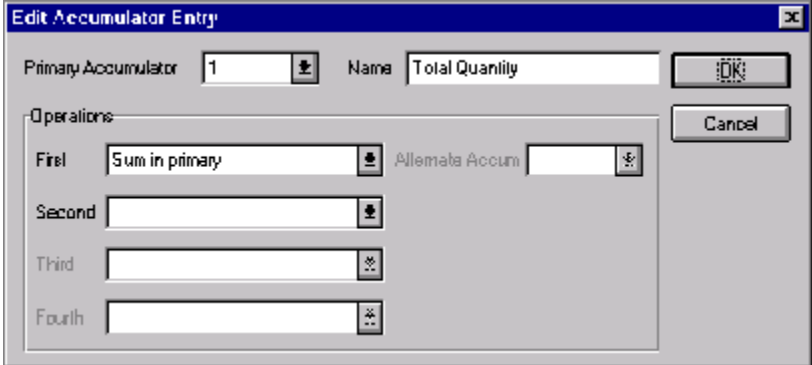
- In the Advanced section, click the Use Standard Rule check box to display the rule options.
- Click the Use Accum option.
- Click NEW... to access the Edit Accumulator Entry dialog to create a new calculation for this element.

5. Select a Primary Accumulator from the list. 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.

For this example, select primary accumulator “1.”

6. In the Name box, type “Total Quantity.” This is a descriptive alias that enables you to differentiate what the accumulators you create are used for.
7. From the First list, select “Sum in primary.” The system adds the numeric value (takes the positive or negative sign of the numbers into account) of the element to the contents of the Primary Accumulator for each iteration of the PO1 group.

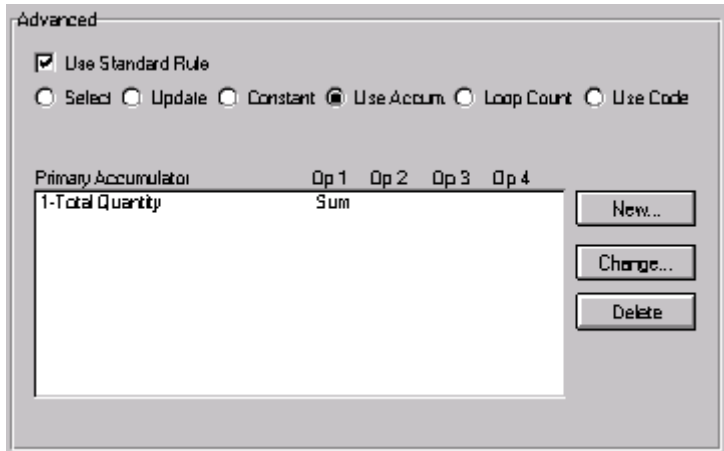
The Edit Accumulator Entry dialog should now look like this:



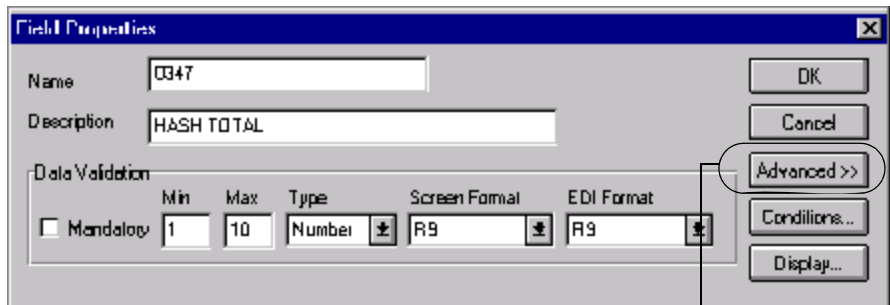
The screenshot shows the 'Edit Accumulator Entry' dialog box. At the top, there is a title bar with the text 'Edit Accumulator Entry' and a close button. Below the title bar, there are two main input fields: 'Primary Accumulator' with a dropdown menu showing '1' and a small arrow icon, and 'Name' with a text box containing 'Total Quantity'. To the right of these fields are 'OK' and 'Cancel' buttons. Below these fields is a section titled 'Operations'. This section contains four rows, each with a dropdown menu and a small arrow icon. The first row is labeled 'First' and has 'Sum in primary' selected in its dropdown. To the right of this row is another dropdown menu labeled 'Alternate Accum'. The second, third, and fourth rows are labeled 'Second', 'Third', and 'Fourth' respectively, and their dropdown menus are currently empty. The 'OK' and 'Cancel' buttons are also visible to the right of the 'Operations' section.

8. Click OK to add the accumulator.

The Advanced section of the Field Properties dialog should now look like this:



9. Click OK on the Field Properties dialog to add the standard rule to the P0102 element.
10. Double-click the CTT02 element (in the CTT segment). This is the element that typically contains the total quantity of the purchase order. The Field Properties dialog is displayed.

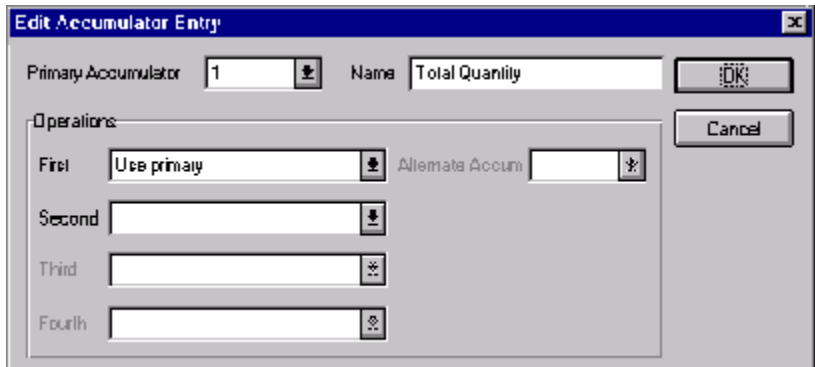


Click Advanced>>

11. In the Advanced section, click the Use Standard Rule check box to display the rule options.
12. Click the Use Accum option.
13. Click NEW... to access the Edit Accumulator Entry dialog to create a new calculation for this element.

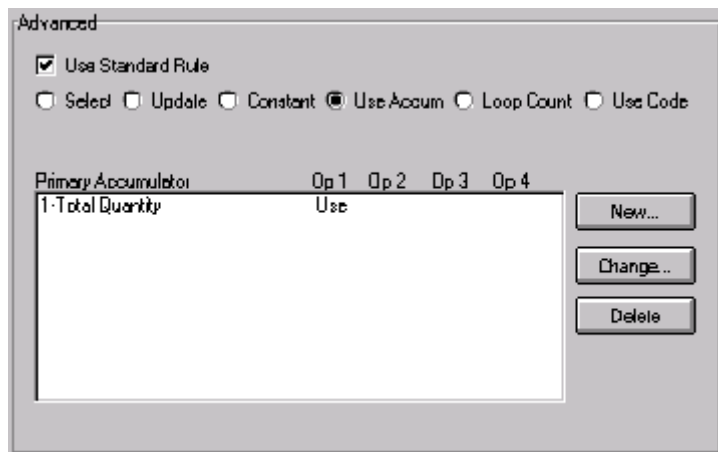
14. From the Primary Accumulator list, select primary accumulator “1.” This accumulator currently contains the total quantity.
15. From the First list, select “Use primary.” The system loads the current value of the accumulator into the CTT02 (Hash Total) element.

The Edit Accumulator Entry dialog should now look like this:



16. Click OK to add the accumulator.

The Advanced section of the Field Properties dialog should now look like this:



17. Click OK on the Field Properties dialog to add the standard rule to the CTT02 element.

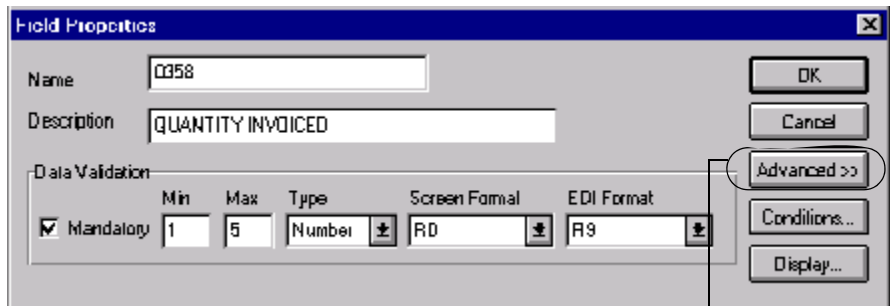
The CTT02 element now contains the total quantity of the purchase order.

Resetting and Calculating a Value Total

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

Complete the following steps to multiply the quantity invoiced for each line item by the unit price to obtain the extended price for an ANSI X12 invoice:

1. Double-click the IT102 element (in the IT1 segment in the IT1 group). This is the element that you typically use to count the quantity invoiced. The Field Properties dialog is displayed.



Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box to display the rule options.
3. Click the Use Accum option.
4. Click NEW... to access the Edit Accumulator Entry dialog to create a new calculation for this element.
5. Select primary accumulator "2."
6. In the Name box, type "Extended Price." This is a descriptive alias that enables you to differentiate what the accumulators you create are used for.
7. From the First list, select "Load primary." The system loads the contents of the element into the Primary Accumulator for each iteration of the IT1 group.

The Edit Accumulator Entry dialog should now look like this:

The dialog box is titled "Edit Accumulator Entry". It features a "Primary Accumulator" field with the value "2" and a "Name" field with the value "Extended Price". Below these is an "Operations" section with four rows labeled "First", "Second", "Third", and "Fourth". The "First" row has a dropdown menu showing "Load primary" and a secondary field labeled "Allegata Accum" with a dropdown menu. The other three rows have empty dropdown menus. On the right side of the dialog are "OK" and "Cancel" buttons.

8. Click OK to add the accumulator.

The Advanced section of the Field Properties dialog should now look like this:

The "Advanced" section of the dialog contains a "Use Standard Rule" checkbox which is checked. Below it are radio buttons for "Select", "Update", "Constant", "Use Accum", "Loop Count", and "Use Code", with "Use Accum" selected. A table below shows the configuration for the standard rule:

Primary Accumulator	Op 1	Op 2	Op 3	Op 4
2-Extended Price	Load			

On the right side of the table are three buttons: "New...", "Change...", and "Delete".

9. Click OK on the Field Properties dialog to add the standard rule to the IT102 element.

- Double-click the IT104 element (in the IT1 segment in the IT1 group). This is the field that contains the unit price for each line item. The Field Properties dialog is displayed.

The image shows the 'Field Properties' dialog box. The 'Name' field contains 'Q212' and the 'Description' field contains 'UNIT PRICE'. Under the 'Data Validation' section, there is a table with columns: 'Mandatory' (checked), 'Min' (1), 'Max' (8), 'Type' (Number), 'Screen Format' (R2), and 'EDI Format' (R9). On the right side of the dialog, there are buttons for 'OK', 'Cancel', 'Advanced >>', 'Conditions...', and 'Display...'. The 'Advanced >>' button is circled in red.

Click Advanced>>

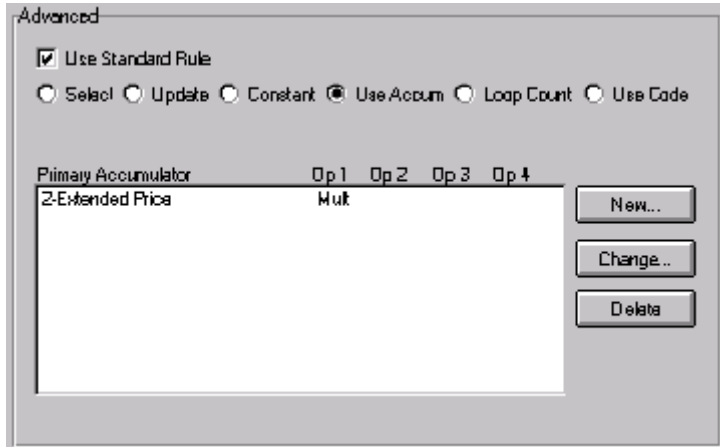
- In the Advanced section, click the Use Standard Rule check box to display the rule options.
- Click the Use Accum option.
- Click NEW... to access the Edit Accumulator Entry dialog to create a new calculation for this element.
- Select primary accumulator "2."
- From the First list, select "Multiply with primary." The system multiplies the value of the IT104 (Unit Price) element with the contents of the primary accumulator, and stores the result in the primary accumulator for each iteration of the IT1 group.

The Edit Accumulator Entry dialog should now look like this:

The image shows the 'Edit Accumulator Entry' dialog box. The 'Primary Accumulator' field contains '2' and the 'Name' field contains 'Extended Price'. Under the 'Operations' section, there are four rows: 'First' (Multiply with primary), 'Second', 'Third', and 'Fourth'. Each row has a dropdown menu and a small icon to its right. The 'First' row's dropdown is set to 'Multiply with primary'. On the right side of the dialog, there are buttons for 'OK' and 'Cancel'.

- Click OK to add the accumulator.

The Advanced section of the Field Properties dialog should now look like this:



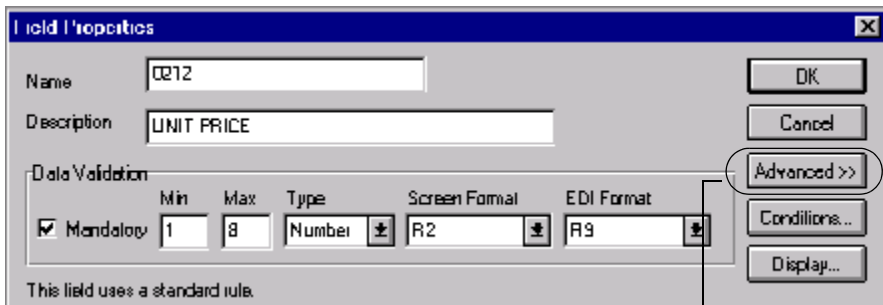
17. Click OK on the Field Properties dialog to add the standard rule to the IT104 element.



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.

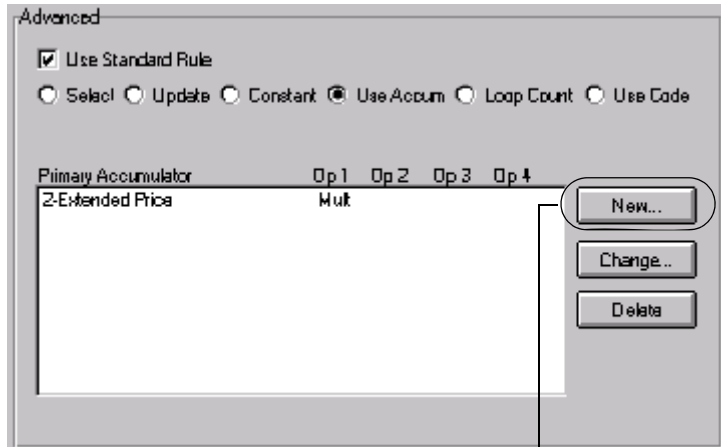
Complete the following steps to generate a running total of the extended price:

1. Double-click the IT104 element (in the IT1 segment in the IT1 group). This is the element that contains the unit price for each line item. The Field Properties dialog is displayed.



Click Advanced>>

- In the Advanced section, you already established one accumulator that is displayed in the list.



Click New...

- Click **NEW...** to access the Edit Accumulator Entry dialog to create another calculation for this element.
- Select primary accumulator "3."
- In the Name box, type "Running Total."
- From the First list, select "Add primary to alternate." 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.



The First box is active only after you select a Primary Accumulator.

- From the Alternate Accum list, select "2." The system adds 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.

The Edit Accumulator Entry dialog should now look like this:

The 'Edit Accumulator Entry' dialog box contains the following fields and controls:

- Primary Accumulator:** A dropdown menu showing the value '3'.
- Name:** A text field containing 'Running Total'.
- Operations:** A section with four rows:
 - First:** A dropdown menu showing 'Add primary to alternate' and an 'Alternate Accum' dropdown showing '2'.
 - Second:** An empty dropdown menu.
 - Third:** An empty dropdown menu.
 - Fourth:** An empty dropdown menu.
- Buttons:** 'OK' and 'Cancel' buttons are located on the right side.

- Click OK to add the accumulator.

The Advanced section of the Field Properties dialog should now look like this:

The 'Advanced' section of the Field Properties dialog includes the following elements:

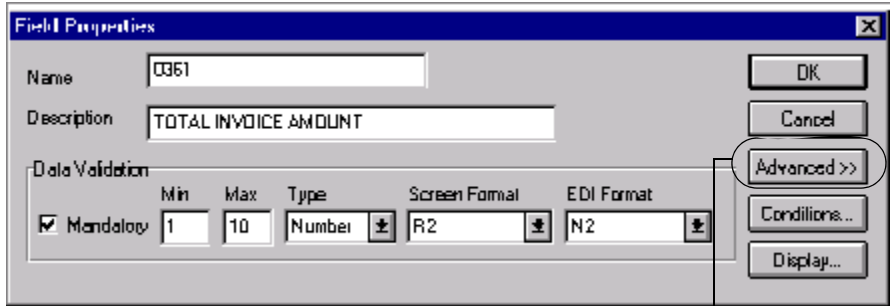
- Use Standard Rule:** A checked checkbox.
- Radio Buttons:** 'Select', 'Update', 'Constant', 'Use Accum' (selected), 'Loop Count', and 'Use Code'.
- Table:**

Primary Accumulator	Op 1	Op 2	Op 3	Op 4
2-Extended Price	Mult			
3-Running Total	AddPA			
- Buttons:** 'New...', 'Change...', and 'Delete' buttons are located on the right side of the table.

- Click OK on the Field Properties dialog to add the standard rule to the IT104 element.

Complete the following steps to load the running total of the extended price into the TDS01 (Total Invoice Amount) element:

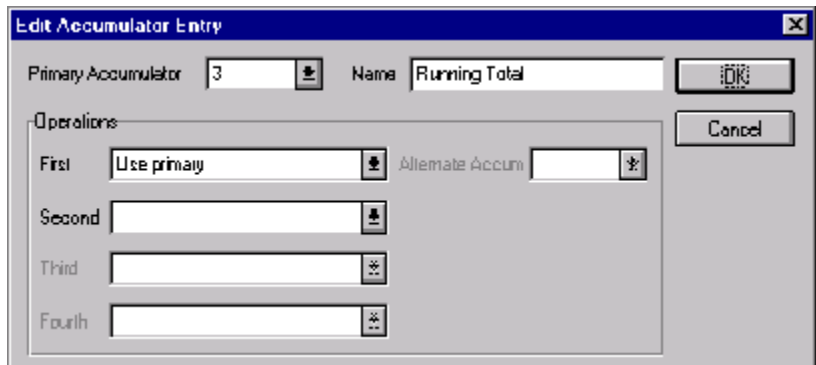
1. Double-click the TDS01 element (in the TDS segment in the IT1 group). This is the element that contains the unit price for each line item. The Field Properties dialog is displayed.



Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box to display the rule options.
3. Click the Use Accum option.
4. Click NEW... to access the Edit Accumulator Entry dialog to create a new calculation for this element.
5. Select primary accumulator "3."
6. From the First list, select "Use primary." The system loads the contents of the primary accumulator into the TDS01 (Total Invoice Amount) element.

The Edit Accumulator Entry dialog should now look like this:



7. Click OK to add the accumulator.

The Advanced section of the Field Properties dialog should now look like this:

Primary Accumulator	Op 1	Op 2	Op 3	Op 4
3-Running Total	Use			

8. Click OK on the Field Properties dialog to add the standard rule to the TDS01 element.



If you are adding this accumulator to a screen entry translation object, you would typically set the TDS01 field to display-only. Please see *Formatting Display-Only (Non-Editable) Fields* on page 220 for more information on formatting display-only fields.

Using Code List Tables

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:Director* 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 or create a code table, you can use a “Use Code” standard rule. For screen entry templates, you can use it to select an element from a list that contains a predefined code table. You can also specify whether or not a compliance error is generated if the element does not contain one of the values in the code table. For print templates, the Use Code function enables you to extract and print the description of a code and validate the contents of a element against the values in the code table for print templates.

The Code Lists function also enables you to store a code's description in an element/field, create a unique code table, use code values from a code table, and flag whether or not the system will generate an error if a validation against the code table fails. You can import and export code lists and copy and paste code lists between forms.

An element with a Use Code standard rule applied to it allows values to either be checked against or selected from the codes in a specified code table. Codes are typically used to further qualify the element. For example, the Product/Service ID (0234) element contains a product code that you expect to be the *customer's* product code. If you receive a value in that element that is something other than the customer's product code, you want the system to generate an error. So, you first use the code table for the Product/Service ID Qualifier (0235), but specify when you load it that the code table should only include the code “BP.” In this code table, the code “BP” is described as a “Buyer’s Part Number.” Then, you make the 0235 code table exclusive for the Product/Service ID Qualifier (0235) element. Therefore, by using this code with the qualifier for the Product/Service ID element, you are indicating that the Product/Service ID element must contain not just a product code, but product code from the *customer*.

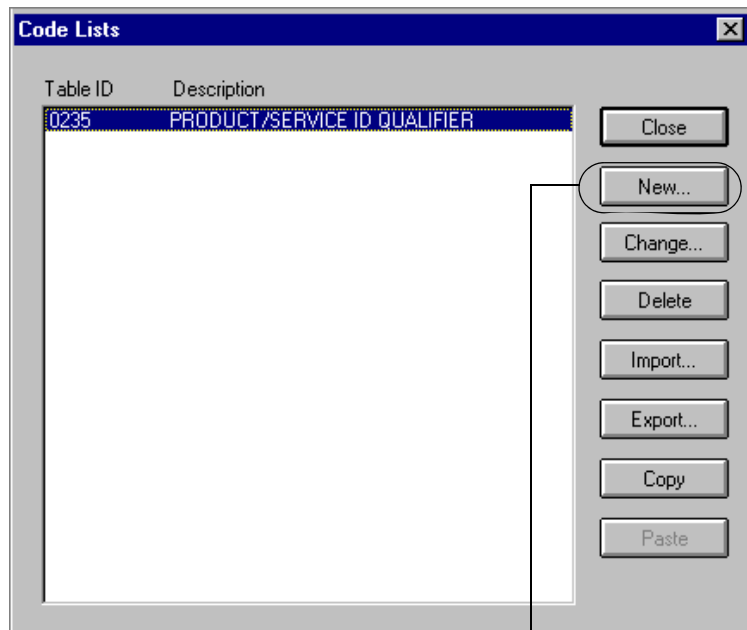
You can perform the following functions on Code List Tables:

- Defining a Code List
- Modifying a Code List or Code List Entry
- Deleting a Code List or Code List Entry
- Importing a Code List
- Exporting a Code List
- Loading Code Tables from the Standard
- Copying and Pasting Code Lists
- Validating Data Against Code List Tables
- Loading Code Item Descriptions

Defining a Code List

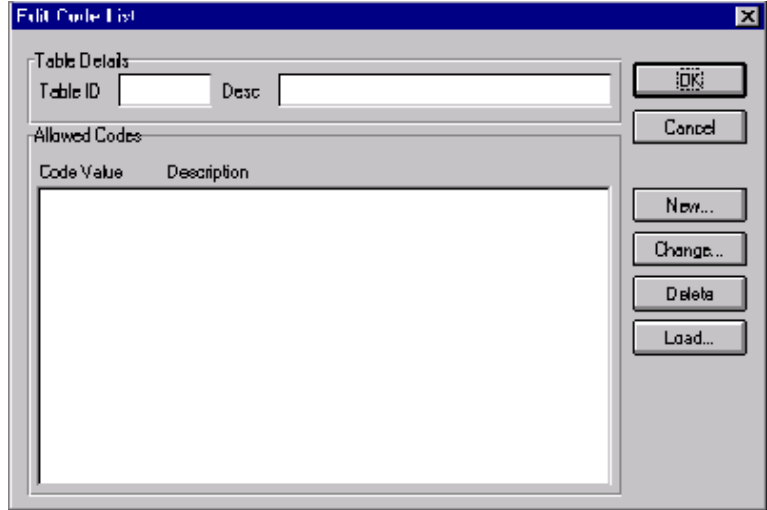
Complete the following steps to define a code list table:

1. Select Code Lists from the Edit menu or the Main Toolbar. The Code Lists dialog is displayed.



Click New...

- Click NEW... to access the Edit Code List dialog.



- In the Table ID box, type the name of the element for which this code list table is used.
- In the Description box, type the description of the element for which this code list table is used.
- Click NEW... to access the Edit Code List Entry dialog.

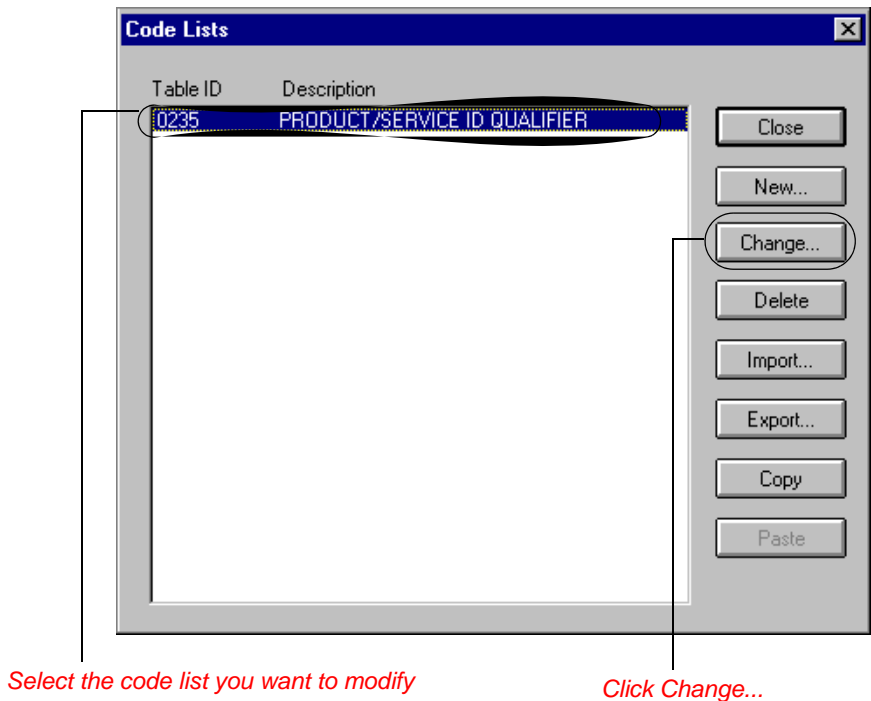


- In the Value box, type the actual value of the code.
- In the Description box, type a description of the code value. The code value description is used if you specify an element (in the Store Fields list on the Field Properties dialog) to which you want the code description mapped.
- Click OK to save the code list entry.
- Repeat steps 5 through 8 to add more code list entries to the code list table.
- Click CLOSE to save and exit the Edit Code List dialog.
- Click CLOSE to exit the Code Lists dialog.

Modifying a Code List

Complete the following steps to modify a code list table or code list entry:

1. Select Code Lists from the Edit menu or the Main Toolbar. The Code Lists dialog is displayed.



2. Highlight the code list that you want to modify.

- Click CHANGE... to access the Edit Code List dialog.

Edit Code List

Table Details

Table ID: 0235 Desc: PRODUCT/SERVICE ID QUALIFIER

Allowed Codes

Code Value	Description
BP	BUYER'S PART NUMBER

Buttons: OK, Cancel, New.., Change.., Delete, Load..

- At this point, you can add new code list entries to the table, delete entries, or modify code list entries. To modify an entry, highlight it and click CHANGE... to access the Edit Code List Entry dialog.

Edit Code List Entry

Value: BP

Description: BUYER'S PART NUMBER

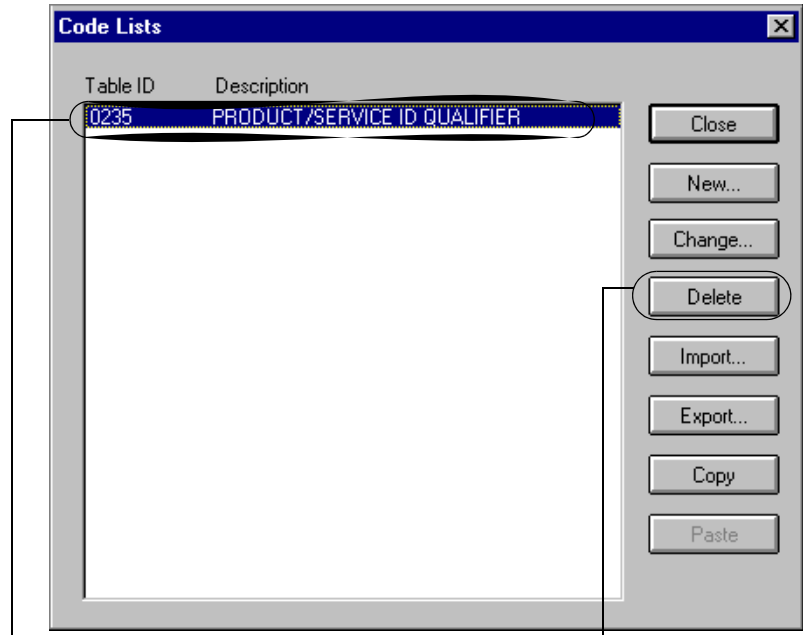
Buttons: OK, Cancel

- Make the necessary changes.
- Click OK to save the code list entry.
- Repeat steps 2 through 6 to modify other code list entries in the code list table.
- Click CLOSE to save and exit the Edit Code List dialog.
- Click CLOSE to exit the Code Lists dialog.

Deleting a Code List or Code List Entry

Complete the following steps to delete a code list table:

1. Select Code Lists from the Edit menu or the Main Toolbar. The Code Lists dialog is displayed.



Select the code list you want to delete

Click Delete

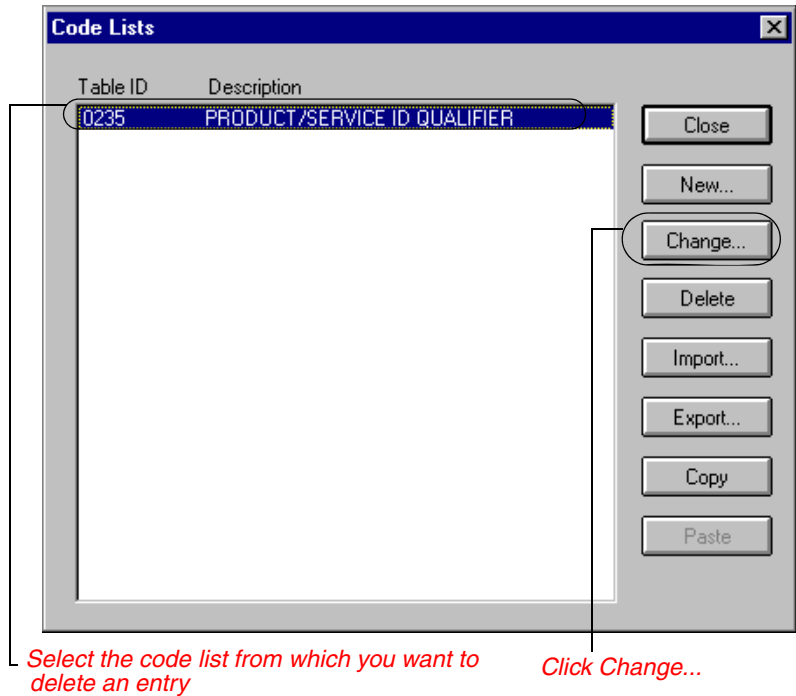
2. If you want to delete a code list, highlight that list.
3. Click DELETE to delete the code list table.



The selected table deleted without warning.

Complete the following steps to delete a code list entry:

1. Select Code Lists from the Edit menu or the Main Toolbar. The Code Lists dialog is displayed.



Select the code list from which you want to delete an entry

Click Change...

2. Highlight the code list from which you want to delete an entry.

3. Click CHANGE... to access the Edit Code List dialog.

The screenshot shows a dialog box titled "Edit Code List". It has a "Table Details" section with "Table ID" set to "0235" and "Desc" set to "PRODUCT/SERVICE ID QUALIFIER". Below this is an "Allowed Codes" section containing a table with two columns: "Code Value" and "Description". The table has one row with "BP" in the "Code Value" column and "BUYER'S PART NUMBER" in the "Description" column. To the right of the table are several buttons: "OK", "Cancel", "New..", "Change..", "Delete", and "Load..".

4. To delete an entry, highlight it and click DELETE to remove the entry. Delete any other entries you wish.



The selected entry is deleted without warning.

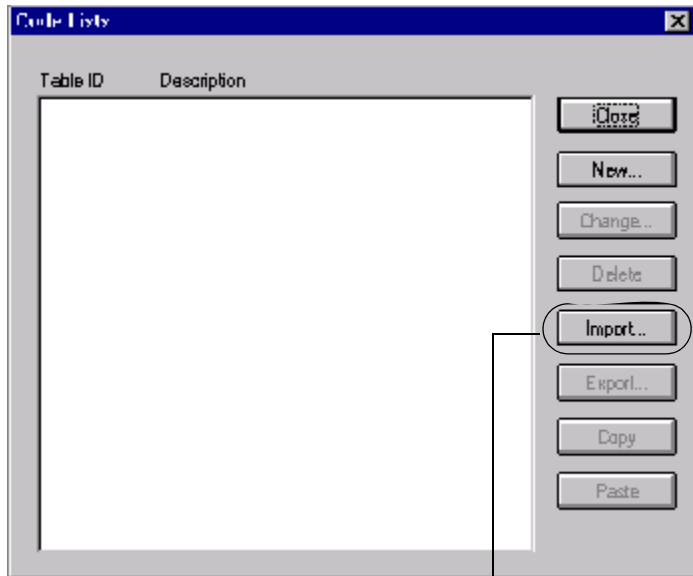
5. Click OK to save the code list table.

Importing a Code List

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

Complete the following steps to import a code list table:

1. Select Code Lists from the Edit menu or the Main Toolbar. The Code Lists dialog is displayed.



Click Import...

2. Click **IMPORT...** to access the Open dialog.



3. Select the location of the code list file. The default location is Application Integration install folder (the default is GENSRVNT\Bin).



The default file extension for code lists is .CDE.

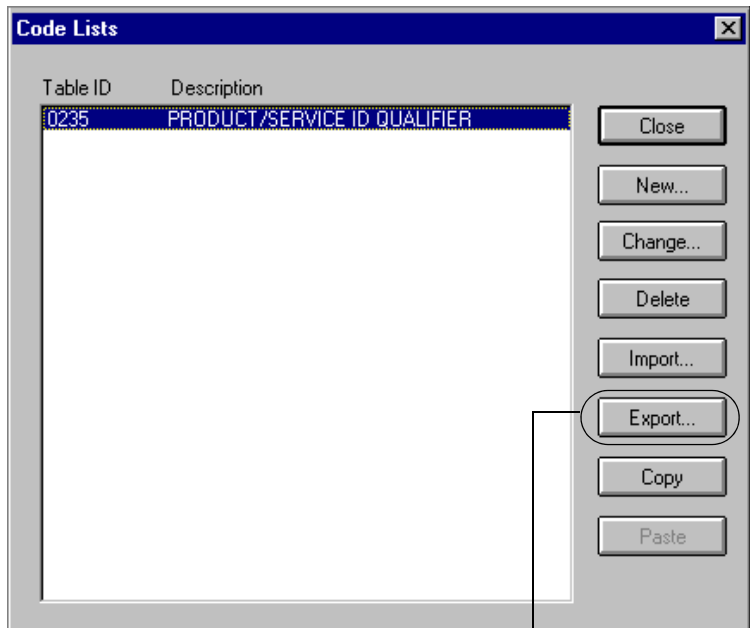
4. The imported code list will be available for your use. Click **CLOSE** to exit the Code Lists dialog.

Exporting a Code List

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:Director.

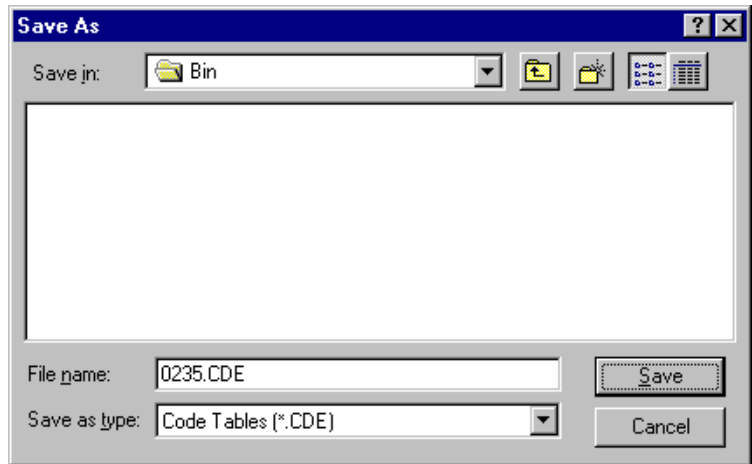
Complete the following steps to export a code list table:

1. Select Code Lists from the Edit menu or the Main Toolbar. The Code Lists dialog is displayed.



Select a code list and click Export..

2. Select a code list and click **EXPORT...** to access the Save As dialog.



3. The file name defaults to the table ID with a “.CDE” file extension. If you want to change the name, type the export file name in the File Name box. The default location is the GENSRVNT\Bin folder.



The default file extension for code lists is .CDE.

4. Click **SAVE** to export the code list and return to the Code Lists dialog.
5. Click **CLOSE** to exit the Code Lists dialog.

Loading a Code List Table from the Standard

The EDI 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.

Complete the following steps to load a code list table from the standard:

1. Double-click the element for which you need to use a code table. The Field Properties dialog is displayed.

Data Validation		Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory		2	2	String	%	%

Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box.
3. Click the Use Code option.

Advanced

Use Standard Rule

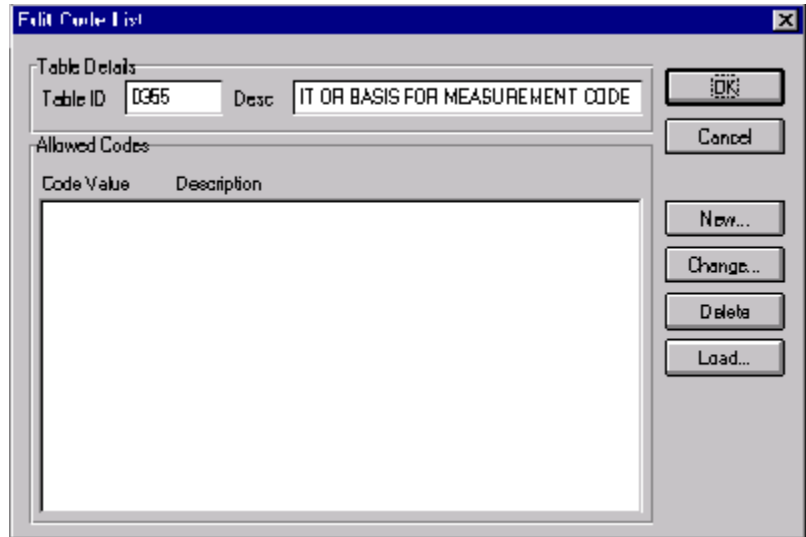
Selected Update Constant Use Accum Loop Count Use Code

Table:

Exclusive

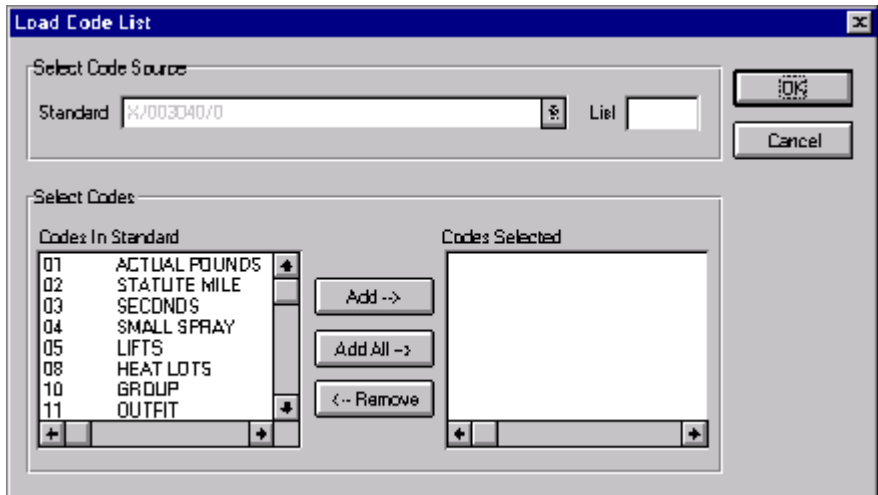
Store Field:

- Click **EDIT TABLE...** to access the Edit Code List dialog.



The Table ID box contains the name of the element for which this code list table will be used. The Description box contains the description of the element for which this code list table will be used.

- Click **LOAD...** to access the Load Code List dialog.



6. You can either select specific codes that you want to load or select the entire list of codes:
 - 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.



We recommend that you only add the codes that you and your trading partners are able to create or accept. Adding all the codes in the code table (using the **ADD ALL--> button) creates a much larger translation object than if you only use selected codes.**

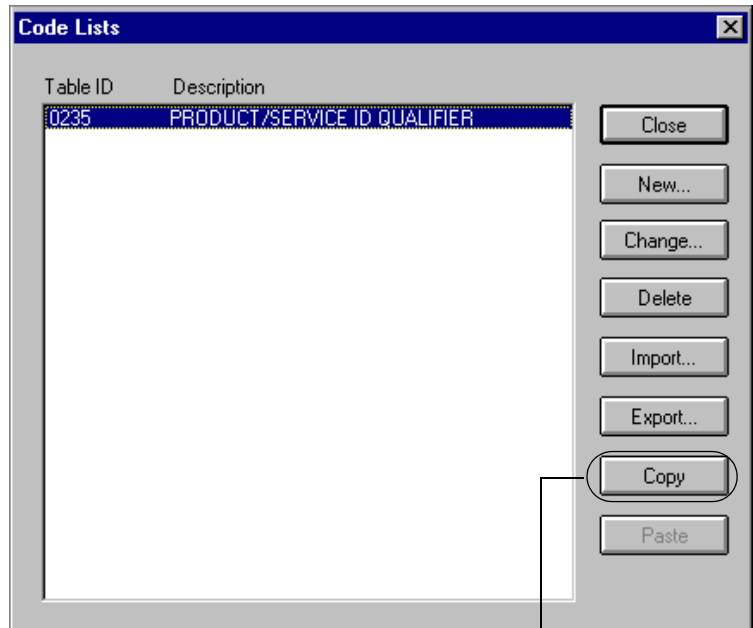
7. Click **OK** to load the code list.
8. Click **CLOSE** to exit the Code Lists dialog.

Copying and Pasting Code Lists

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

Complete the following steps to copy and paste a code list table:

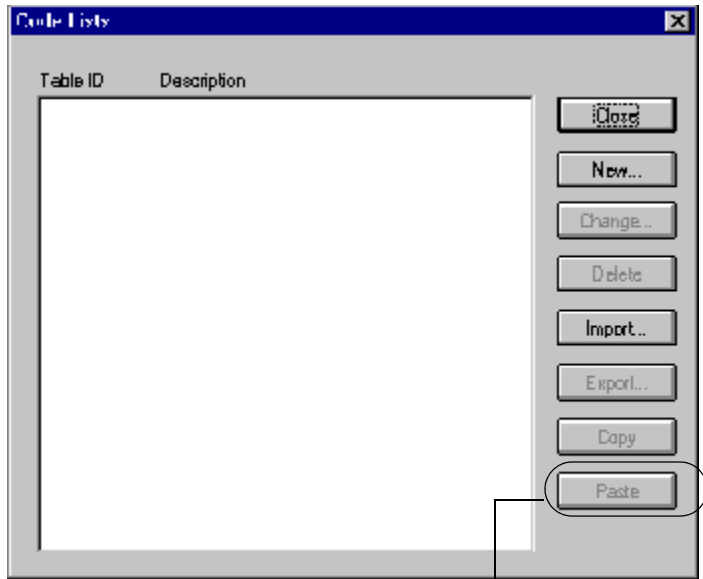
1. Select Code Lists from the Edit menu or the Main Toolbar. The Code Lists dialog is displayed.



Select a code list and click Copy..

2. Select a code list and click COPY... to copy the code list to the clipboard.
3. Click CLOSE to exit the Code Lists dialog.
4. Open the map in which you want to use the code list, if the map is not already open.

5. Select Code Lists from the Edit menu or the Main Toolbar. The Code Lists dialog is displayed.



Click Paste..

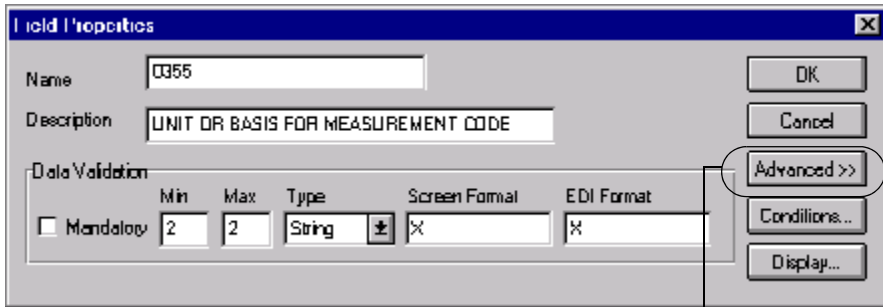
6. Click PASTE... to add the copied code list to this map.
7. Click CLOSE to exit the Code Lists dialog.

Validating Data Against Code List Tables

For print translation objects, you can use code list tables to validate an element for compliance checking.

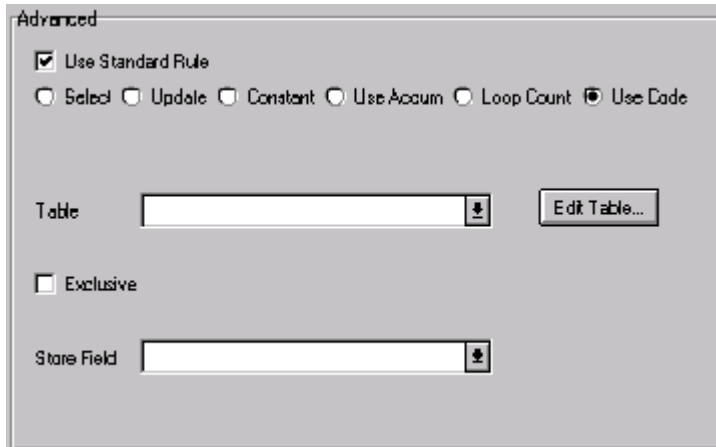
Complete the following steps to validate data against a code list table:

1. Double-click the element for which you need to validate data against a code table. The Field Properties dialog is displayed.



Click Advanced >>

2. In the Advanced section, click the Use Standard Rule check box.
3. Click the Use Code option.



4. From the Table list, select the code list table that the data in this element is validated against. If the Tables list is empty, you need to load a code table. Please see *Loading a Code List Table from the Standard* on page 157 for more information.

5. If you need to specify that the element *must* contain one of the codes from the specified table (nothing else is valid for that element), click the Exclusive check box.

For example, if the element 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 element, the system will flag it as an error.

6. Click OK to add this standard rule to the element.

Loading Code Item Descriptions

Loading code item descriptions is typically only done for print translation objects.

Complete the following steps to load a code item description:

1. Double-click the element for which you need to load a code item description. The Field Properties dialog is displayed.
2. In the Advanced section, click the Use Standard Rule check box.
3. Click the Use Code option.
4. From the Table list, select the code list table that the data in this element will be validated against. If the Tables list is empty, you need to load a code table. Please see *Loading a Code List Table from the Standard* on page 157 for more information.
5. If you need to specify that the element *must* contain one of the codes from the specified table (nothing else is valid for that element), click the Exclusive check box.

For example, if the element 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 element, the system will flag it as an error.

6. From the Store Field 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.

For example, if the code you used is “BP,” it would be useful to view the description of the code (“Buyer's Part Number”). If you selected element “0350” from the Store Field list, the description for the BP code is mapped to element “0350.”

7. Click OK to add this standard rule to the element.

Using Extended Rules

Extended rules enable you to use a Gentran:Director proprietary programming language to perform virtually any operation you require. You can use these rules to define more complex translations than are available through the standard rules.

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

- Introducing extended rules
- When rules are processed
- Defining extended rules
- Extended rule syntax
- Language reference

Introducing Extended Rules

You can use extended rules to define operations that are not possible using standard rules. You define extended rules with the Gentran:Director proprietary programming language. This is a full programming language that gives you access to the entire Gentran:Director internal storage area.

Any variables that are not already defined as part of the specification of the form, 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 Section

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 alphanumerics and the “:” and “_” characters. The first character in a variable name may not be a numeric. Please note that all variable names are case-sensitive.



A declaration must be terminated with a semicolon (;). To improve readability, you typically include a blank line in between the declaration and statement sections.

The data types that are supported by extended rules are the following:

- Integer – is a whole number with no decimal component.
- Real – is a whole number that may have a decimal component.
- String – contains one or more printable characters.
- Datetime – contains a date or time.
- Array – defines a table of multiple occurrences of a single data type.

Examples of declarations of variables of each data type are as follows:

- Declare **i** as an integer
integer i;
- Declare **r** as a real number
real r;
- Declare **d** as a date or time
datetime d;

- Declare **s** as a 20-character string
string[20] s;
- Declare **a** as an array of 10 integers
integer a[10];
- Declare **p** as an array of 50 10-character strings
string[10] p[50];

Statements Section

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.

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. Each of these components is explained in detail in *Extended Rule Syntax* on page 174.

When Rules are Processed

You can specify pre- and post-session rules on the Session Rules dialog. 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.

Additionally, you can attach extended rules to two different levels of form components:

- Groups, sub-groups, repeating segments, and repeating segments, using the Loop Level Extended Rule dialog.
- Fields and elements, using the Field Level Extended Rule dialog.

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 form. The scope of an extended rule is defined as follows:

- Pre-session extended rules (defined on the Session Rules dialog) are in scope for every rule in the translation object.
- On New, On Store, On Open, and On Delete extended rules (defined on the Loop Level Extended Rules dialog) are in scope until the conclusion of the corresponding action (creating a new group, storing a group, opening an existing group, and deleting a group).
- Field level extended rules are only in scope for the duration of the element.

An extended rule attached to the current form component depends on the type and state of the form component. For 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 page 176. An extended rule that is attached to an element is only in scope for the duration of the 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 or in the Pre-Session rule.

The translator builds the data storage area for a form based on the EDI structure. Therefore, extended rules address the form based on the hierarchy of the EDI file.

The translator processes screen entry forms in the following manner:

1. Reads the data into the Document Editor and processes all cumulative standard rules (e.g., accumulators, etc.). Does not process any extended or non-cumulative standard rules).
2. Executes group level extended rules in the following situations:
 - On_New when a new group is created
 - On_Open when an existing group is edited
 - On_Store when a group is saved
 - On_Delete when a group is deleted
3. Executes field level rules in the following sequence when focus is lost from the field:
 - Non-cumulative standard rules
 - Extended rules
4. When the form is saved or recalculated, all standard rules are executed.

The translator processes print forms in the following manner:

1. Reads the data and processes all cumulative standard rules (e.g., accumulators). Does not process any extended or non-cumulative standard rules).
2. Executes group level extended rules in the following situations:
 - On_New just prior to printing a group
 - On_Store just after printing a group
3. Executes field level rules in the following sequence after the field is printed:
 - Non-cumulative standard rules
 - Extended rules

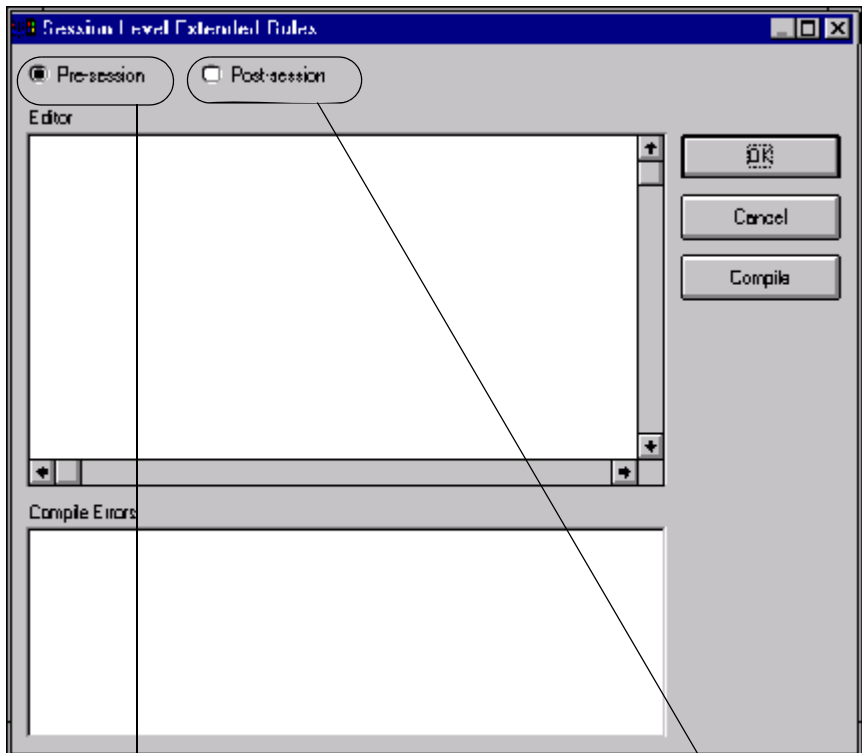
Defining Extended Rules

The form 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 form 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 form component. You can define extended rules to access three different levels of form components.

Complete the following steps to define a session rule:

1. To specify a pre- or post-session rule, select Session Rules from the Edit menu. The Session Level Extended Rules dialog is displayed.



Click the Pre-session option to define a pre-session rule

Click the Post-session option to define a post-session rule

2. If you want to define a pre-session rule, click the Pre-session option. If you want to define a post-session rule, click the Post-session option. 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.

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

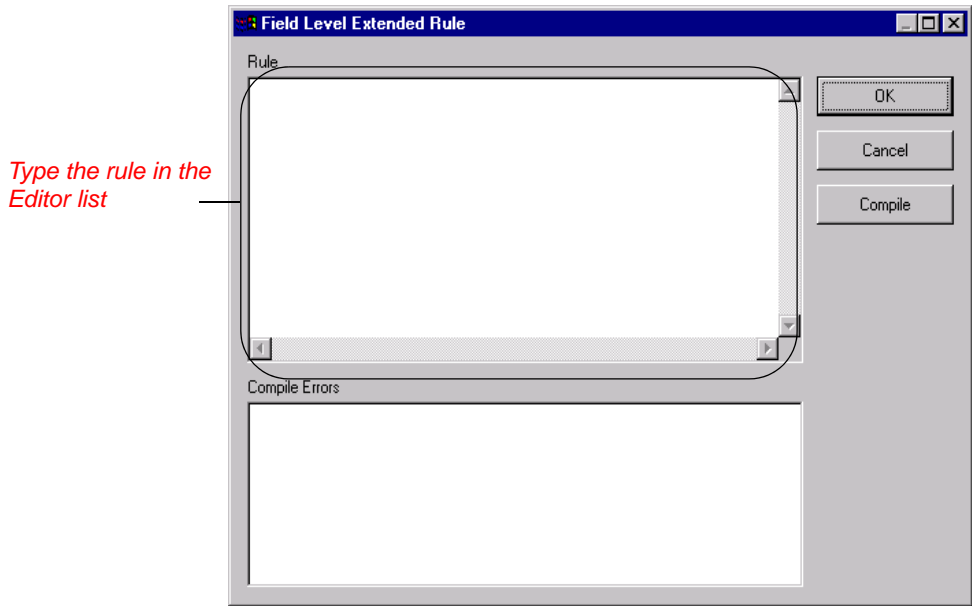
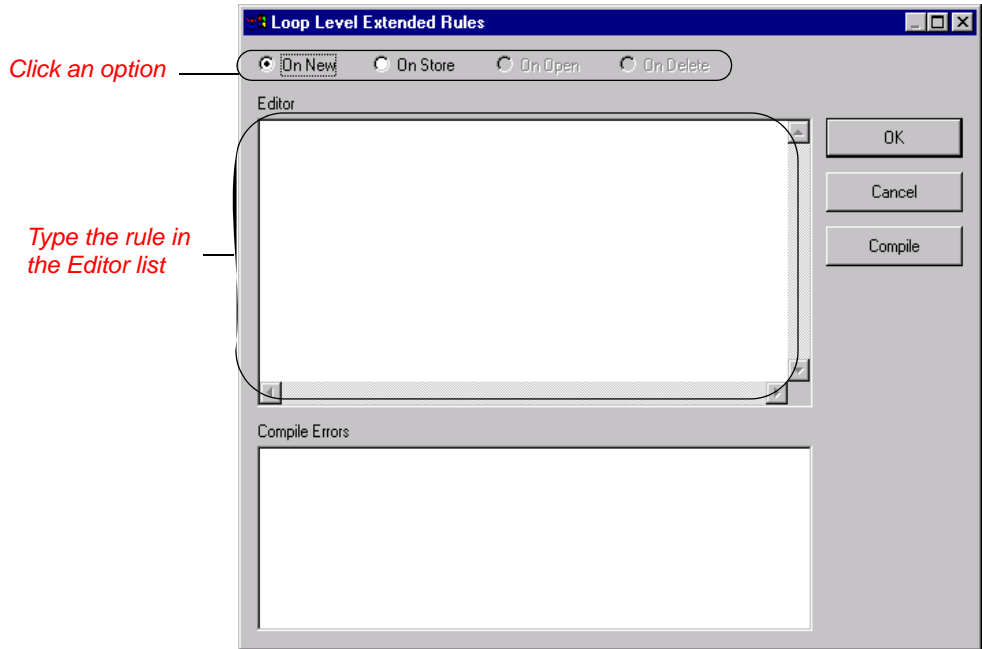
3. In the Editor list, type the extended rule.
4. If you want to check the rule for errors, click `COMPILE` to compile the extended rule. Any warnings or errors are displayed in the Errors list.

This function gives you immediate feedback about the accuracy of your rule. Regardless, the rule is compiled when you compile the entire translation object.

5. Correct any errors that the system flagged and click `COMPILE` again. Repeat this process until there are no errors generated.
6. Click `OK` to add the extended rule.

Complete the following steps to define an extended rule for a form component:

1. Click the form component with the right mouse button to access the shortcut menu. Select `Rules...` (or `Extended Rule...` if the form component is a field or element) from the shortcut menu.
 - If the form component is an EDI file, group, sub-group, repeating segment, the Loop Level Extended Rules dialog is displayed.
 - If the form component is an element, the Field Level Extended Rule dialog is displayed.



2. If you are defining the rule for a group, complete this step, otherwise proceed to step 3. If you want the extended rule to be executed when you create a new group in the translation object, click the On New option. If you want the rule to be executed when you store a group, click the On Store option. If you want the extended rule to be executed when you open an existing group, click the On Open option. If you want the extended rule to be executed when you delete a group, click the On Delete option. You can define an On New, On Store, On Open, and On Delete rule for a single group.
3. In the Editor list, type the extended rule.
4. If you want to check the rule for errors, click `COMPILE` to compile the extended rule. Any warnings or errors are displayed in the Errors list.

This function gives you immediate feedback about the accuracy of your rule. Regardless, the rule is compiled when you compile the entire translation object.

5. Correct any errors that the system flagged and click `COMPILE` again. Repeat this process until there are no errors generated.
6. Click OK to add the extended rule.

Extended Rule Syntax

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.

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

Keywords and Commands

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. A list of Gentran:Director execution control keywords is as follows:

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

A list of Gentran:Director commands is as follows:

- GET
- SET
- STRDATE
- CONCAT
- LEN
- ATOI
- ATON
- EMPTY
- EXIST
- INDEX
- NTOA

- COUNT
- DELETE
- LEFT
- RIGHT
- MID
- SELECT
- UPDATE



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

Operators

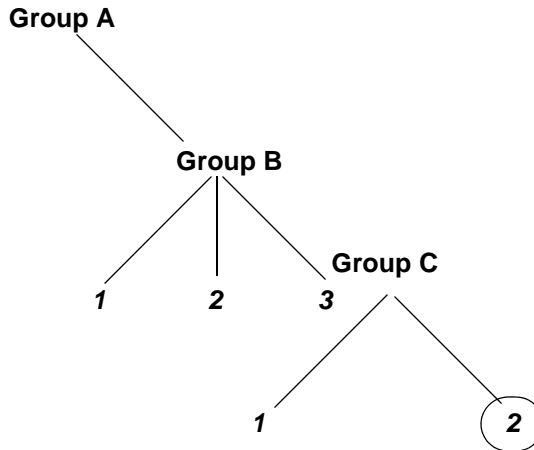
Operators define the simplest operation in an expression. You can use the following operators in extended rules:

- + 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

Operations are performed on symbols. The symbols that you can use in Genran:Director extended rules are variables, constants, form components/ internal storage, arrays, and accumulators. You can address existing form components and you also have the ability to create additional instances of form components, as long as the form component is originally defined in internal storage. You can use this function to create, for example, extra line items when one line item field is already defined in internal storage. You must address each type of symbol in the proper syntax.

- To address a string constant, you must enclose the constant value in quotes:
#fieldname = "HDR";
 where HDR is the constant value
- To address a field or create a field in internal storage, within the scope of the current action, the syntax is #FIELD_NAME
#field_1 = 2;
 where 2 is a numeric constant value
- 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
- 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:

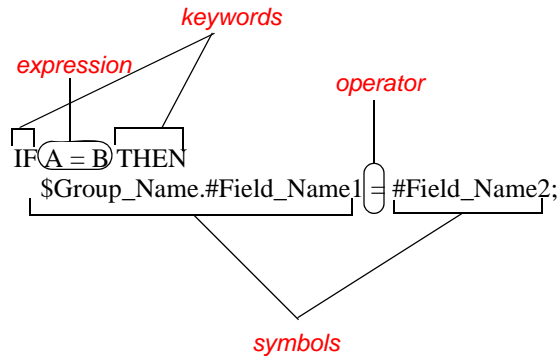


- To address an array (of any type), you address each element of the array individually. For example, if array_1 is an array (of integers) and is declared:
integer array_1[5]
 with variables 0 through 4, each element of the array is addressed individually as follows:
 array_1[0]
 array_1[1]


```
array_1[2]
array_1[3]
array_1[4]
```

- 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;

An example of a simple statement is as follows:



Common Statements and Examples

Some of the common statements are as follows:

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

Assignment 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
```

A numeric expression can consist of number, numeric fields, numeric variables, and numeric functions combined with the standard arithmetic operators. A string expression can consist of string constants, string fields, string variables, and string functions concatenated with the “+” operator. A datetime expression can consist of a datetime constant, datetime field or datetime variable.

Some examples of assignment expressions are as follows:

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

Datetime Expressions

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 functions are written as follows (month specified as 0-11):

```
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 ("95/4/6") you want to convert. You use this function if you are using non-standard syntax and need to specify the syntax you are using.

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 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.
```

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

Gentran:Director 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*.

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:Director interprets the value as either *true* or *false*. The system interprets a zero value as false and a nonzero value as true.



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:Director 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:Director 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. The IF/THEN/ELSE statement is written as follows:

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

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

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

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. The right function extracts a specified number of character from the right of the string. The mid functions extracts from a specified position in the string to the right, either to the end of the string or for a specified number of characters. These statements are used as follows:

```
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);  
s4 = mid(s,3);
```

The *strdate* functions 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. The syntax is as follows:

```
datetime d;  
string[8] s;  
  
strdate(d,"%y/%m/%d",s);
```

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);
```

The *strstr* function finds a substring inside a string. This function returns the position of the first instance of the designated substring. The syntax is as follows:

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

Numerical Functions

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`
- `aton`
- `ntoa`

The *len* function counts and returns the number of characters in a string. An example is as follows:

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

The *atoi* function converts strings into integers. The *aton* function converts string into real numbers. The *ntoa* function converts real numbers into strings. An example is as follows:

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

Remove Field Value Function (EMPTY)

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. The syntax is as follows:

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

The above example sets the value of the specified instance of the VATC element to null. When this function is used in expressions, the null value is propagated if any part of the expression is evaluated to be null. This function is typically used to test whether or not an error occurred.

Existence of Data Function (EXIST)

The *exist* function returns a non-zero (true) value if there is data in a specified field in internal storage. If there is not data in the specified field, this function returns a zero (false) value. This function is typically used as a part of a condition. The syntax is as follows:

```
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 (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 = " "
```

Count Function (COUNT)

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

```
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)

The *delete* function deletes the specified iteration of a repeating record or group. The syntax is as follows:

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

Select Function (SELECT)

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.

The syntax is as follows:

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

Update Function (UPDATE)

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. The syntax is as follows:

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

Alphabetic Language Reference

This section contains an alphabetic reference for the Gentran:Director proprietary programming language keywords and commands. This enables you to quickly and easily refer to information on specific keywords or commands. This section uses the following conventions:

- | | |
|------------------|--|
| While | Bold typeface is used to express what you need to type. |
| <i>condition</i> | In syntax, italic letters indicate information that you need to provide. |
| [1] | In syntax, items inside square brackets are optional. |

The following programming guidelines are utilized in this section:

- Keywords and commands are shown in all lower case letters.
- Form 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

The **atoi** function is a numerical function that converts strings into integers. The numerical functions enable you to convert one data type to another.

The syntax is as follows:

```
int = atoi(string);
```

where: int = integer variable
string = string variable

An example is as follows:

```
integer a;  
string[20] s;  
s = "5";  
a = atoi(s);  
// "a" contains the value 5
```

aton

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.

The syntax is as follows:

```
real = aton(string);
```

where: real = real number variable
string = string variable

An example is as follows:

```
real b;  
string[20] s;  
s = "5.5";  
a = aton(s);  
// "a" contains the value 5.5
```

begin

Gentran:Director 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**.



Do not end conditions with a semicolon (;) – this terminating syntax is necessary for statements *only*.

The syntax is as follows:

```
if condition then  
begin  
    statement1;  
    statement2;  
end
```

break

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.

An example is as follows:

```
integer i  
integer j;
```

```
i = j = 0;  
while i<10 do  
begin  
    if (i = 8) then  
        continue;  
    if (i = 9) then  
        break;  
end
```

```
//As long as "i" has a value less than "10" the loop repeats. If "i" has  
//a value of "i", the loop continues. If "i" has a value  
//of "9" the loop terminates.
```

concat

The **concat** function concatenates a specified number of characters from one string onto the end of another string.

The syntax is as follows:

```
concat(string,string,num_char);
```

```
where: string      = string variable  
       num_char    = number of characters from the second string onto the  
                   end of the first string
```

An example is as follows:

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

```
//Concatenate five characters from string "s2" onto the end of string  
/"s1"
```

continue

The **continue** function continues the execution of the innermost loop without processing the statements in the loop that follow the **continue** statement.

An example is as follows:

```
integer i  
integer j;
```

```
i = j = 0;  
while i<10 do  
begin  
  if (i = 8) then  
    continue;  
  if (i = 9) then  
    break;  
end
```

```
//As long as "i" has a value of "8" the innermost loop repeats. If "i"  
//does not have a value of "8" the next statement (break statement)  
//is processed.
```

count

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.
```

delete

The **delete** function deletes a specified occurrence of a group.

The syntax is as follows:

```
delete($GROUP_NAME[N]0;
```

where [n] is the occurrence of the group that you want to delete.

An example is as follows:

```
delete($ILD[2]);
```

```
//Deletes the second occurrence of the ILD group.
```

empty

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.

The syntax is as follows:

```
empty($GROUP_NAME[index][index][index]. #FIELD_NAME)
```

An example is as follows:

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

```
//Set the value of the specified instance of the VATC element to null
```

end

Gentran:Director 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**.



Do not end conditions with a semicolon (;) – this terminating syntax is necessary for statements *only*.

The syntax is as follows:

```
if condition then  
begin  
    statement1;  
    statement2;  
end
```

exist

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

The syntax is as follows:

```
if exist($GROUP_NAME[index][index][index]. #FIELD_NAME) then
```

An example is as follows:

```
if exist($ILD[0][1][1].#VATC) then
```

//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).

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 = " "

get

The **get** function enables you to access individual components of a datetime variable.

The syntax is as follows:

```
integer_variable = get datetime_component (datetime_variable);
```

where: integer_variable	=	integer variable
datetime_component	=	individual component of the datetime variable
datetime_variable	=	datetime variable of which you want to access a component part

An example is as follows:

```
integer a;
integer b;
datetime d;

a = get days (d);
b = get hours (d);

//Accesses the days from the datetime variable "d" and loads into
//variable "a". Accesses the hours from the datetime variable "d" and
//loads into variable "b".
```

if...then...else

The **if**, **then**, and **else** keywords allow the use of conditional logic. Genran:Director 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:Director interprets the value as either *true* or *false*. The system interprets a zero value as false and a nonzero value as true.

Gentran:Director 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:Director 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.



Do not end conditions with a semicolon (;) – this terminating syntax is necessary for statements *only*.

An example is as follows:

```
if condition then  
begin  
    statement1;  
    statement2;  
end  
else  
begin  
    statement3;  
    statement4;  
end
```

index

The **index** function enables you to determine which instance of a particular loop the translator is currently accessing.

The syntax is as follows:

```
index(integer_variable);
```




where: integer_variable

= integer variable that indicates the hierarchical level for which you want to determine the loop count

An example is as follows:

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 
```

left

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

An example is as follows:

```
string [25]name;
string [5]temp_variable;
name = "Acme Shipping Company"
temp_variable = left(name,4);
// "temp_variable" would contain "Acme"
```

len

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.

The syntax is as follows:

```
number_char = len(string);
```

where: num_char = integer variable

An example is as follows:

```
integer a;
a = len("hello");
// "a" contains the value 5
```


messagebox

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.

The syntax is as follows:

```
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)

An example is as follows:

```
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.

The defined_numbers for the button and icon types are as follows:

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

The message box return values are as follows:

Return Value	Action Selected
1	OK selected
2	CANCEL selected
6	YES selected
7	NO selected

mid

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.

The syntax is as follows:

```
string_variable = mid(string_variable,start_pos,num_char)
```

where: start_pos = integer variable
num_char = integer variable

An example is as follows:

```
string [25]name;  
string [10]temp_variable;  
name = "Acme Shipping Company"  
temp_variable = left(name,5,8);  
// "temp_variable" would contain "Shipping"
```

ntoa

The **ntoa** function is a numerical function that converts real numbers into strings. The numerical functions enable you to convert one data type to another.

The syntax is as follows:

```
string = ntoa(real, string);
```

where: real = real number variable
string = string variable

An example is as follows:

```
real b;  
string[8] s;  
b = 5.5;  
ntoa(5.5, s);  
//The variable "s" contains the string "5.5".
```

right

The **right** function extracts a specified number of characters from the right side of a string variable or field.

The syntax is as follows:

```
string_variable = right(string_variable,num_char)
```

where: num_char = integer variable

An example is as follows:

```
string [25]name;  
string [10]temp_variable;  
name = "Acme Shipping Company"  
temp_variable = left(name,7);  
// "temp_variable" would contain "Company"
```

select

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. The syntax is as follows:

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

An example is as follows:

```
string[10]temp_var1,temp_var2,temp_var3;
select Text1, Text2, Text3 into temp_var1, temp_var2, temp_var3 from
ÂPartnerlookup where Tablename = "Test" and Item = "widgit"
//Example assumes that a Lookup table named "Test" has been
//created.
```

set

The **set** function enables you to define individual components of a datetime variable.

The syntax is as follows:

```
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

An example is as follows:

```
integer a;
integer b;
datetime d;
set days (d,a);
set hours (d,a);
//Defines the days of the datetime variable "d" from variable "a".
//Defines the hours of the datetime variable "d" from variable "b".
```

strdate

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.

The syntax is as follows:

```
strdate(datetime,"format",string);
```

where: datetime = datetime variable (month specified as 0-11)
 format = desired date format
 string = string variable

An example is as follows:

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

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

**//Converts a datetime variable into an eight character string in the
//format "year/month/day".**

The format specifiers are as follows:

Format Specifier	Description
%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)
%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

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”.

The syntax is as follows:

```
integer = strstr("string","substring");
```

where: integer = integer variable
 string = string
 substring = part of the string

An example is as follows:

```
integer d;
```

```
d = strstr("mississippi","is");
```

```
//Finds the first instance of the substring "is" inside the string  

//"mississippi" and returns the position of that first substring.
```

update

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. The syntax is as follows:

```
update tablename set fieldname =expression [fieldname =expression] where  

key = expression [and key = expression];
```

An example is as follows:

```
string[15]temp_variable;
```

```
temp_variable = "5 Pound Hammer";
```

```
update PartnerXref set Text1 = temp_variable where MyItem =  

&PN5lbHMR" and Tablename = "Tools";
```

```
//Example assumes that a cross-reference table named "Tools" has been  

//created.
```

while...do

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.



Do not end conditions with a semicolon (;) – this terminating syntax is necessary for statements *only*.

An example is as follows:

```
integer i;

while i < 10 do
begin
    if (i = 8) then
        continue;
    if (i = 9) then
        break;
end
```

**//While "i" is less than ten, execute the loop. If "i" is equal to or greater
//than ten, terminate the loop.**

winexec

The **winexec** function enables you to exit Gentran:Director and execute another program. 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. These error codes are the same as the Windows equivalent, and are therefore not listed in this section.

The syntax is as follows:

```
winexec("program",window_display)
```

where: program = executable program name string
window_display = number that indicates how you want the program window displayed

An example is as follows:

```
winexec("program.exe", 3)
```

**//Exits Gentran:Director and executes the "program.exe"
//program asynchronously. The program window is displayed
//maximized (3).**

The window_display numbers that control the appearance of the program window are as follows:

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.
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.

Select and Update Available Options

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.

Document Table The field names that are available when using the select or update extended rules on the Document Table are the following:

- TESTMODE
- AGENCY
- VERSION
- TRANSACTIONSETID
- RELEASE
- DOCUMENTTYPE
- REFERENCEDATA
- DOCUMENTNAME
- APPFIELD1
- APPFIELD2
- APPFIELD3
- APPFIELD4
- APPFIELD5
- APPFIELD6
- DOCUMENTPARTNERKEY
- CONTROLNUMBER
- PARTNERKEY

Partner Table The field names that are available when using the select or update extended rules on the Partner Table are the following:

- PARTNERNAME
- EDICODE
- APPLICATIONPARTNERKEY

The alternate keys that are available for the Partner Table are the following:

- PARTNERKEY
- ALTERNATEKEY (refers to Application Key)
- APPLICATIONPARTNERKEY (refers to Application Key)

Location Table The field names that are available when using the select or update extended rules on the Location Table are the following:

- CONTACTNAME
- NAME
- ADDRESS1
- ADDRESS2

- ADDRESS3
- CITY
- STATE
- ZIP
- COUNTRY
- TELEPHONE
- PRIMARYREFCODE
- SECONDARYREFCODE
- FAX

Lookup Table The field names that are available when using the select or update extended rules on the Lookup Table are the following:

- ITEM
- DESCRIPTION
- TEXT1
- TEXT2
- TEXT3
- TEXT4

The alternate key that is available for the Lookup Table is the following:

- TABLENAME

Cross-reference Table The field names that are available when using the select or update extended rules on the Cross-reference Table are the following:

- MYITEM
- PARTNERITEM
- DESCRIPTION
- TEXT1
- TEXT2
- TEXT3
- TEXT4

The alternate key that is available for the Cross-reference Table is the following:

- TABLENAME

Interchange Table The field names that are available when using the select or update extended rules on the Interchange Table are the following:

- TESTMODE
- CONTROLNUMBER
- VERSION
- APPFIELD1
- APPFIELD2
- APPFIELD3
- APPFIELD4
- APPFIELD5
- APPFIELD6
- AGENCY
- PARTNERKEY

Group Table The field names that are available when using the select or update extended rules on the Group Table are the following:

- TESTMODE
- CONTROLNUMBER
- FUNCTIONALGROUPID
- VERSION
- APPFIELD1
- APPFIELD2
- APPFIELD3
- APPFIELD4
- APPFIELD5
- APPFIELD6
- AGENCY

Generic Envelope Segment Table The field names that are available when using the select or update extended rules on the Generic Envelope Segment Table are the following:

- 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

Customizing Screen Entry Forms

Screen entry forms provide an organized format for entering an EDI file into the Document Editor for translation and communication to your partners. The screen entry form ensures that your users enter all the data necessary to create the required EDI documents. You can customize screen entry forms to your organization's specific needs. For example, you might want to customize the way data is arranged in fields, or the format of numbers, dates, or times. You can even prevent fields in the screen entry form from being displayed when the translation object (compiled form) is used.

This chapter explains how to customize screen entry forms, including the following topics:

- Customizing field labels
- Setting an initial value for a field
- Setting a constant value for a field
- Creating definitions for a list box
- Creating Help text for fields and list boxes
- Creating Help text for frames
- Formatting entry fields
- Formatting display-only (non-editable) fields
- Hiding fields
- Formatting numbers
- Formatting dates and times
- Formatting strings
- Storing EDI code value descriptions

Customizing Field Labels

Field labels identify the contents of the fields included in a screen entry form. The system uses the element name from the EDI standard as the default to label a field. You can customize the default field label to be any name you choose.

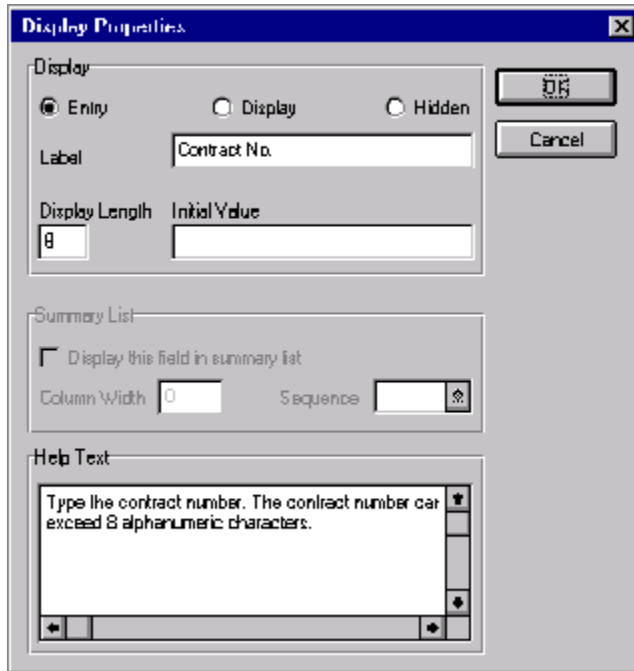
We recommend that you label every field. If you decide *not* to label a field, follow the steps below, but delete the existing field label. For example, you might decide not to label the fields of a company address to be entered in a standard address format, because it is obviously an address. In this instance, the name of the company is on the first line; the street address is on the second line; and the city, state, and zip code are on the third line and do not need to be labeled.

Complete the following steps to customize a field label:


1. Double-click the element for which you want to customize the field label. The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	1	8	String	X	X

- Click DISPLAY... to access the Display Properties dialog.



The screenshot shows the 'Display Properties' dialog box for a field. The dialog has a title bar with 'Display Properties' and a close button. It is divided into three sections: 'Display', 'Summary List', and 'Help Text'.
- The 'Display' section has three radio buttons: 'Entry' (selected), 'Display', and 'Hidden'. Below them is a 'Label' text box containing 'Contract No.', a 'Display Length' spinner box set to '8', and an 'Initial Value' text box.
- The 'Summary List' section has a checkbox 'Display this field in summary list' which is unchecked. Below it are 'Column Width' and 'Sequence' spinner boxes.
- The 'Help Text' section contains a text area with the text: 'Type the contract number. The contract number can exceed 8 alphanumeric characters.' The text area has scroll bars on the right and bottom.

- In the Label box, modify the text to customize the label.
- Click OK to save the new label and exit the Display Properties dialog.
- Click CLOSE to exit the Field Properties dialog.
-  Click Generate Layout on the Main Toolbar to refresh the display of the screen entry form.

Setting an Initial Value for a Field

If you usually need the same value in a field each time that field is used, you might want to set an initial value for the field. After you set an initial value, that value is displayed in the field every time that field is used in the form.

For a screen entry form, the user can override the initial value for a field if you want to enter data that is different from the initial value. If you do *not* want the user to be able to override the field value, then you should use a constant value for the field, instead of an initial value. See *Using a Constant Value for a Field* on page 210 for more information on setting a constant value.

For example, if you know that one person at your company is nearly always the person who should be contacted by telephone, you might set an initial value for a Telephone Contact field on an invoice, by setting the telephone contact's name as the initial value. There are occasions when someone else at your company is the contact person, however, so you want to be able to override the initial value in the Telephone Contact field when those exceptions occur.

Complete the following steps to set an initial value for a field:

1. Double-click the element for which you want to set an initial value. The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	1	8	String	X	X

2. Click DISPLAY... to access the Display Properties dialog.

The screenshot shows the 'Display Properties' dialog box for a field named 'Contract No.'. The dialog has a title bar with 'Display Properties' and a close button. It is divided into three main sections: 'Display', 'Summary List', and 'Help Text'.
1. **Display Section:** Contains three radio buttons: 'Entry' (selected), 'Display', and 'Hidden'. Below them is a 'Label' field containing 'Contract No.'. Underneath are two input fields: 'Display Length' with the value '8' and an empty 'Initial Value' field.
2. **Summary List Section:** Features a checkbox labeled 'Display this field in summary list' which is currently unchecked. Below it are 'Column Width' and 'Sequence' input fields, with a small icon to the right of the 'Sequence' field.
3. **Help Text Section:** Contains a text area with the text: 'Type the contract number. The contract number can exceed 8 alphanumeric characters.' The text area has a scroll bar on the right and a horizontal scroll bar at the bottom.
On the right side of the dialog, there are two buttons: 'OK' and 'Cancel'.

2. In the Initial Value box, type the value that you want to be the default displayed for this field.
3. Click OK to save the initial value for this field and exit the Display Properties dialog.
4. Click CLOSE to exit the Field Properties dialog.

Using a Constant Value for a Field

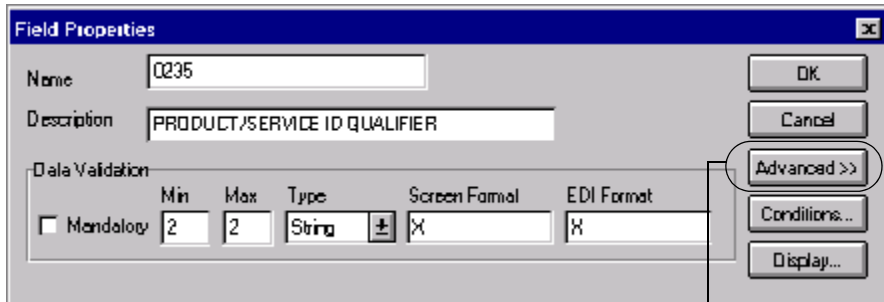
If you always use the same value in a field each time that field is used, you might want to use a constant value for the field so the user does not have to enter the same information each time. After you set the constant value, that value is displayed in the field every time the field is used.

You *cannot* override the constant value for a field. If you want to be able to override the field value, then you should set an initial value for the field, instead of a constant value. See *Setting an Initial Value for a Field* on page 208 for more information on setting an initial value.

For example, you might set a constant value for a qualifier field on an invoice if you know that you never need to override the value in the field. In this instance, you would probably hide the field, as well. See *Hiding Fields* on page 221 for more information on hiding fields.

Complete the following steps to use a constant value for a field:

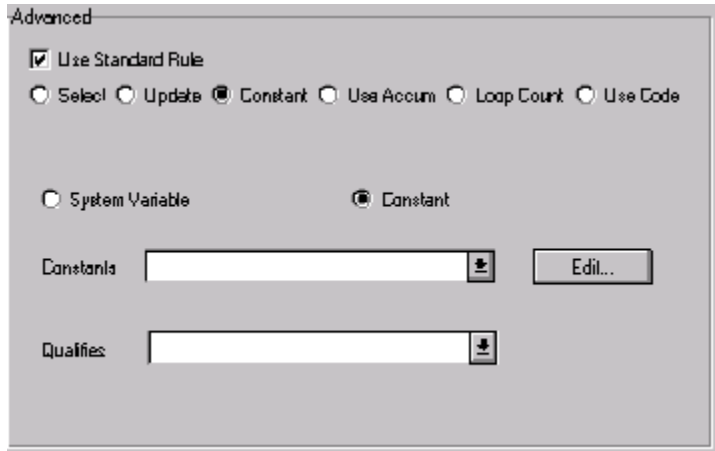
1. Double-click the element for which you want to set a constant value. The Field Properties dialog is displayed.



Click Advanced >>

2. Click the Use Standard Rule check box.
3. Click the Constant option.

- Click the second Constant option.



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

- Use Standard Rule
- Select
- Update
- Constant
- Use Account
- Loop Count
- Use Code

Below these options, there are two radio buttons:

- System Variable
- Constant

There are two text input fields:

- A field labeled "Constants" with a dropdown arrow on the right and an "Edit..." button to its right.
- A field labeled "Qualifies" with a dropdown arrow on the right.

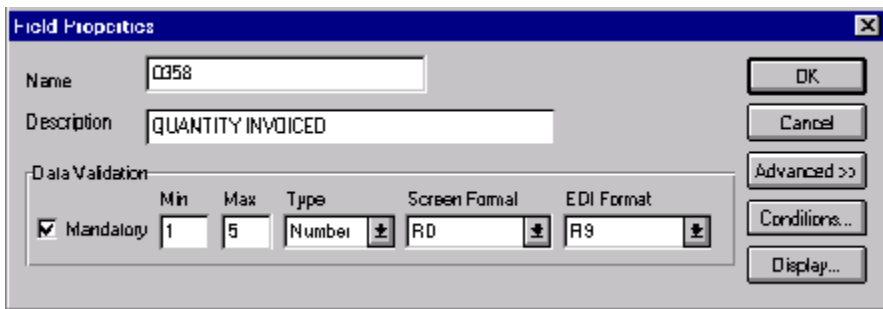
- From the Constants list, select the constant that you want to use for this field. If the constant that you want to use for this field is not in the list, you need to create or edit a constant. Please see *Defining Literal Constants* on page 115 for more information on creating a literal constant. Please see *Editing Literal Constants* on page 116 for more information on editing a literal constant.
- Click OK and the data that was stored in the selected constant is loaded in the field.

Creating Definitions for a List Box

When Gentran:Director identifies a group in a transaction set, it creates a separate frame to contain the data within the group. It also creates a list box on the parent frame to allow the user access to the items in the group. To identify the item in the group, at least one field needs to be displayed in the list box. Each set of data can be displayed as a column in the list box. For example, you might have a list box of items on a purchase order. Within the Items list box, you can set up three columns of data: Quantity, Description, and Unit Price. To define the list box columns, you would perform the steps below for the Quantity field, then repeat the steps for the Description and Unit Price fields.

Complete the following steps to create a definition for a list box:

1. Double-click the element for which you want to define a list box. The Field Properties dialog is displayed.



- Click DISPLAY... to access the Display Properties dialog.

- In the Display section, click the Display option.
- In the Summary List section, click the Display this field in summary list check box.
- In the Column Width box, type the width of this column in number of characters allowed.



Ensure that the column width is long enough to accommodate the average length (in characters) of the values expected for this column of data.

- In the Sequence box, select the sequence of this column in the list box. For example, if this is to be the first column in the list box, select "1."
- Click OK to save the list box definition and exit the Display Properties dialog.
- Click CLOSE to exit the Field Properties dialog.
- Repeat steps one through nine for each element in the group that contains data you want to define in the list box.

To label the columns of data set up in the list box, you must add column headings (static text) to the form.

Creating Help Text for Fields and List Boxes

You can create context-sensitive Help text for a field or list box to aid screen entry translation object users in entering the information required (in the Document Editor). The Help text is displayed during data entry when the a user selects a field or an entry in the list box and presses F1.

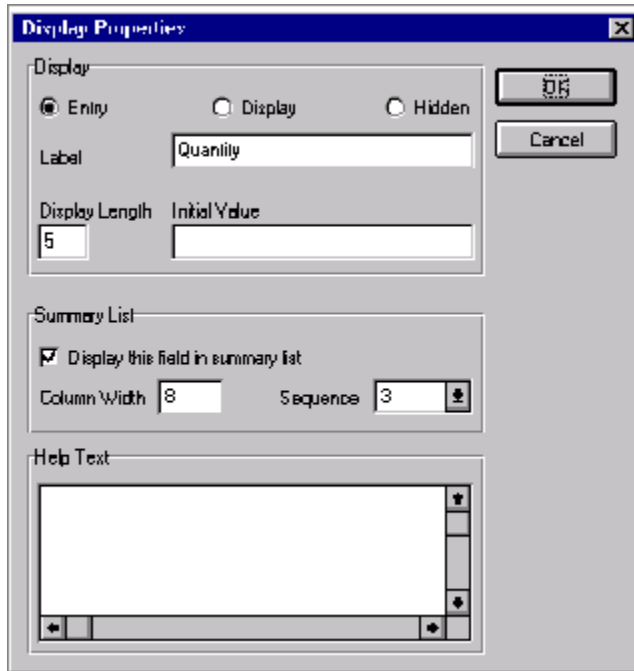
Creating Help Text for Fields

Complete the following steps to create Help text for a field:

1. Double-click the element for which you want to create Help text. The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input checked="" type="checkbox"/> Mandatory	1	5	Number	R0	R9

2. Select DISPLAY... to access the Display Properties dialog.



3. In the Help Text list, type the Help text.



Gentran:Director does not automatically wrap Help text when users access context-sensitive help in Document Editor. Therefore, you may want to insert a hard return each time the Help text you are typing reaches the right side of the Help Text list, rather than allowing the list to scroll to the right.

4. Click OK to save the Help text and exit the Display Properties dialog.
5. Click CLOSE to exit the Field Properties dialog.

Creating Help Text for List Boxes

List boxes contain information from selected elements in a repeating group. The information from each element is displayed as a column in the list box.

Complete the following steps to create Help text for a list box:

1. Click the group for which you want to create Help text with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu to access the Group Properties dialog.

The screenshot shows the 'Group Properties' dialog box. The 'Identification' section contains 'Name' (D400_JT1) and 'Desc' (Line Items). The 'Loop Control' section contains 'Min Usage' (0) and 'Max Usage' (3392). The 'Display' section contains a 'Promote group to parent frame' checkbox, 'Frame Caption' (Line Items), and 'List Caption' (Line Items). The 'Listbox Help Text' area is empty. The 'OK' button is highlighted.

2. In the Listbox Help Text list, type the Help text.
3. Click OK to save the Help text and exit the Group Properties dialog.

Creating Help Text for Frames

The Help text is displayed during data entry when the a user clicks `FRAME ?` or selects Frame Help from the Document Editor Help menu, when the frame is currently in focus.

Complete the following steps to create Help text for a frame:

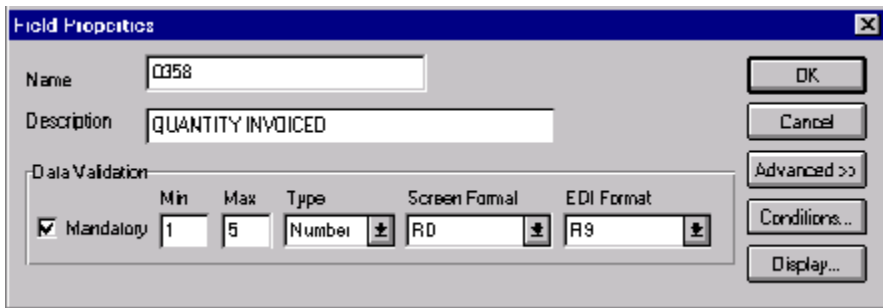
1. Click the group for which you want to create Help text with the right mouse button to access the shortcut menu. Select Properties... from the shortcut menu to access either the File Format Properties dialog or the Group Properties dialog (depending on the level of the group).
2. In the Frame Help Text list, type the Help text. The Frame Help Text list is only be active if the Max Usage is greater than “1.”
3. Click OK to save the Help text and exit the Properties dialog.

Formatting Entry Fields

An entry field is a field that can be edited. When you create a form, Gentran:Director makes every field an entry field by default. For entry fields, you can change the field type to Display (the field is display-only) or Hidden (the field is not displayed at all).


Complete the steps below to change a display-only or hidden field to an entry field:

1. Double-click the element you want to format as an entry field. The Field Properties dialog is displayed.



2. Click DISPLAY... to access the Display Properties dialog.


The screenshot shows the 'Display Properties' dialog box. The 'Display' section has three radio buttons: 'Entry' (selected), 'Display', and 'Hidden'. The 'Label' field contains 'Quantity'. The 'Display Length' field contains '5'. The 'Summary List' section has a checked checkbox 'Display this field in summary list', a 'Column Width' field with '8', and a 'Sequence' field with '3'. The 'Help Text' section has a text area with the text: 'Type the quantity of this invoiced item. The quantity exceed 5 digits.' There are 'OK' and 'Cancel' buttons on the right side of the dialog.

2. In the Display section, click the Entry option.
3. Click OK to exit the Display Properties dialog.
4. Click CLOSE to exit the Field Properties dialog.
5.  Click the Generate Layout button on the Main Toolbar to refresh the display of the screen entry form.

Formatting Display-Only (Non-Editable) Fields

When you create a form, Gentran:Director makes every field an entry field by default. You can change the default to Display if you want a field to be display-only (non-editable). Data can be viewed but not edited in display-only fields.

Complete the steps below to format a display-only field:

1. Double-click the element you want to format as a display-only field. The Field Properties dialog is displayed.
2. Click DISPLAY... to access the Display Properties dialog.
3. In the Display section, click the Display option.
4. Click OK to exit the Display Properties dialog.
5. Click CLOSE to exit the Field Properties dialog.
6.  Click the Generate Layout button on the Main Toolbar to refresh the display of the screen entry form.

Hiding Fields

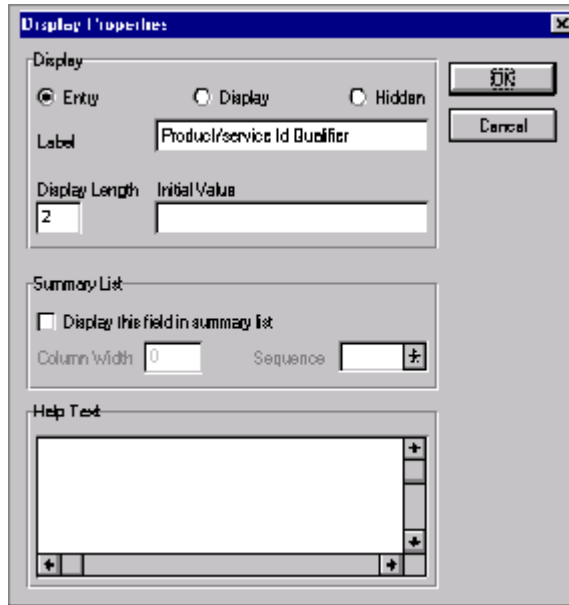
Usually, you want all EDI data to be displayed. However, there may be times when you want to prevent the data in a field from being displayed. For example, you might want to hide a field when it contains a qualifier that is explained by the label of the associated data field. In this instance, you might set a constant for the field (so that the field always contains the appropriate qualifier without it having to be entered each time), then hide the field. You also might want to hide a field if it is a mandatory segment that must be kept activated, but does not need to be displayed.


Complete the following steps to hide a field in a screen entry form:

1. Double-click the element you want to hide. The Field Properties dialog is displayed.

Field Properties							
Name		0235					OK
Description		PRODUCT/SERVICE ID QUALIFIER					Cancel
Data Validation							
	Min	Max	Type	Screen Format	EDI Format		
<input type="checkbox"/> Mandatory	2	2	String	X	X	Advanced >>	
						Conditions...	
						Display...	

2. Click DISPLAY... to access the Display Properties dialog.



3. In the Display section, click the Hidden option.
4. Click OK to exit the Display Properties dialog.
5. Click CLOSE to exit the Field Properties dialog.
6.  Click the Generate Layout button on the Main Toolbar to refresh the display of the screen entry form.

Formatting Numbers

A number-type element contains either an integer or a real number. If a field is a number type, you must specify the format (either “R” or “N”) and the number of decimal places.

An “N”-formatted number is a number that has an implicit decimal point (e.g., “2.01” is formatted as “201”). An “R”-formatted number has an explicit decimal point (e.g., “2.01” is formatted as “2.01”), and truncates trailing zeroes. Whether you use the “N” or “R” format depends on your requirements. Regardless of whether you use the “N” or “R” format, you must also indicate the number of decimal places in the field.

You can format numbers that are written to an EDI file in various numeric formats.

Complete the following steps to format numbers in a screen entry form:

1. Double-click the element that you want to format. The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	1	10	Number	R9	R9

2. In the Type list, verify that Number is selected, indicating that the element is an integer or real number.
3. From the Screen Format list, select the format for the number to be displayed on the frame.
 - Number formats beginning with “R” represent numbers with an explicit decimal point. For example, if the format you select is R2, then numbers are formatted with two places to the right of the decimal point (e.g., “2.01”).
 - Number formats beginning with “N” represent numbers with an implicit decimal point. The decimal point is not displayed, but the system still processes the number correctly. For example, if the format you select is N2, then numbers are formatted with two places to the right of an implicit decimal point.

Valid number formats are the following:

- R0 Number formatted with an explicit decimal point and no decimal places
- R1 Number formatted with an explicit decimal point up to 1 decimal place
- R2 Number formatted with an explicit decimal point up to 2 decimal places
- R3 Number formatted with an explicit decimal point up to 3 decimal places
- R4 Number formatted with an explicit decimal point up to 4 decimal places
- R5 Number formatted with an explicit decimal point up to 5 decimal places
- R6 Number formatted with an explicit decimal point up to 6 decimal places
- R7 Number formatted with an explicit decimal point up to 7 decimal places
- R8 Number formatted with an explicit decimal point up to 8 decimal places
- R9 Number formatted with an explicit decimal point up to 9 decimal places
- N0 Number formatted with an implicit decimal point and no decimal places
- N1 Number formatted with an implicit decimal point up to 1 decimal place
- N2 Number formatted with an implicit decimal point up to 2 decimal places
- N3 Number formatted with an implicit decimal point up to 3 decimal places
- N4 Number formatted with an implicit decimal point up to 4 decimal places
- N5 Number formatted with an implicit decimal point up to 5 decimal places
- N6 Number formatted with an implicit decimal point up to 6 decimal places
- N7 Number formatted with an implicit decimal point up to 7 decimal places
- N8 Number formatted with an implicit decimal point up to 8 decimal places
- N9 Number formatted with an implicit decimal point up to 9 decimal places



Usually, the number formats in the Screen Format list and in the EDI Format list are the same. However, if

there is an “N” format selected in the EDI Format list, you might want to select an “R” format in the Screen Format list so that the decimal point in the numbers is displayed.

4. In the EDI Format list, verify that the number format selected is the correct format that is written to the EDI file.



A number format must be designated for number-type elements in the EDI Format list, even for hidden fields.

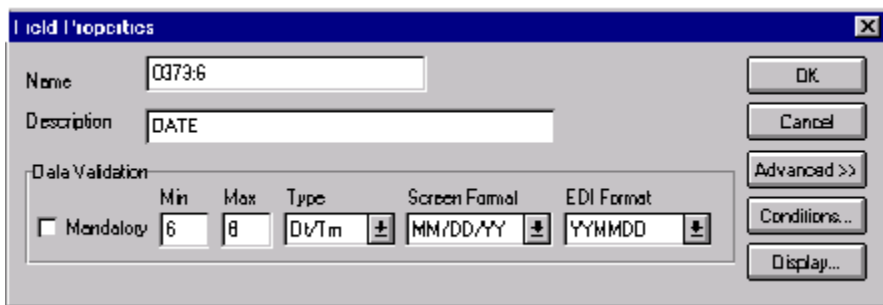
5. Click OK to exit the Field Properties dialog.

Formatting Dates and Times

A date/time-type element contains either a date or a time. If an element is a date/time type, you must designate how the date or time is formatted in the screen entry form. You can format dates and times that are written to an EDI file in various formats.

Complete the following steps to format dates and times in a screen entry form:

1. Double-click the element you want to format. The Field Properties dialog is displayed.



2. If the screen format for the date or time is longer than the length in the Max box, type a new maximum field length in the Max box. This value should be the maximum number of digits, including separators (/), that you expect for the date or time.



The value in the Max box should be the same as the value in the Display Length box on the Display Properties dialog (click DISPLAY... on the Field Properties dialog to access the Display Properties dialog).

3. In the Type list, verify that Dt/Tm is selected, indicating that the field value must be a date or a time.
4. From the Screen Format list, select the format for how you want the date or time to be formatted on the frame.

Valid date/time formats are the following:

MM/DD/YY Two-digit month, separator, two-digit day, separator, last two digits of year (example: 12/15/97 for December 15, 1997)

YYYYMMDD	Four-digit year, two-digit month, two-digit day (example: 19971215)
YYYY/MM/DD	Four-digit year, separator, two-digit month, separator, two-digit day (example: 1997/12/15)
DD/MM/YYYY	Two-digit day, separator, two-digit month, separator, four-digit year (example: 15/12/1997)
MM/DD/YYYY	Two-digit month, separator, two-digit day, separator, four-digit year (example: 12/15/1997)
HHMM	Two-digit hour, two-digit minutes (example: 0330 for thirty minutes past three o'clock)
HHMMSS	Two-digit hour, two-digit minutes, two-digits seconds (example: 033045 for thirty minutes and forty-five seconds past three 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)
DDMMYY	Two-digit day, two-digit month, last two-digits of year (example: 151297)
MMDDYY	Two-digit month, two-digit day, last two-digits of year (example: 121597)
DDMMYYYY	Two-digit day, two-digit month, four-digit year (example: 15121997)
MMDDYYYY	Two-digit month, two-digit day, four-digit year (example: 12151997)
YYDDD	Last two-digits of year, three-digit Julian day (example: 97349 for the 349th day of 1997)
DDYY	Three-digit Julian day, last two digits of year (example: 34997)
YYYYDDD	Four-digit year, three-digit Julian day (example: 1997349)
DDYYYY	Three-digit Julian day, four-digit year (example: 3491997)
YY/DDD	Last two digits of year, separator, three-digit Julian day (example: 97/349)
DDD/YY	Three-digit Julian day, separator, last two digits of year (example: 349/97)
YYYY/DDD	Four-digit year, separator, three-digit Julian day (example: 1997/349)

DDD/YYYY	Three-digit Julian date, separator, four-digit day (example: 349/1997)
MONTH	The month (example: December)
DAY	The day of the week (example: Friday)

5. In the EDI Format list, verify that the date or time format selected is the correct format that is written to the EDI file.



A date/time format must be designated for date/time-type elements in the EDI Format list, even for hidden fields.

6. Click OK to exit the Field Properties dialog.

Formatting Strings

A string-type field or element contains one or more printable characters. If you specify that an 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/or numbers that are allowed to be used for a string-type element. You can use syntax tokens in the Screen Format and EDI Format fields of the Field Properties dialog to designate defined ranges of characters and/or numbers that are allowed to be used for a string-type element.

Complete the following steps to format strings in a screen entry form:

1. Double-click the element you want to format. The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	1	8	String	X	X

2. In the Type list, verify that String is selected, indicating that the field value must contain a string of printable characters.
3. In the Screen Format box, verify that the string format selected specifies how you want the string to be formatted on the frame.
4. In the EDI Format box, verify that the string format selected is the correct format that is written to the EDI file.



A string format must be designated for string-type elements in the EDI Format box, even for those that you hide.

5. Click OK to exit the Field Properties dialog.

You can create syntax tokens to define ranges of characters and/or numbers that are allowed to be used for a string-type field or element. See *Using the String Type* on page 45 for information on how to create or edit syntax tokens.

Storing EDI Code Value Descriptions

Gentran:Director enables you to store code value descriptions in code storage fields. This feature is useful for elements that contain an EDI code, and for which you want to print the description of the code instead of or in addition to the code itself. You set up a standard rule to direct the system to look up an EDI code value, find the description for that value, and display the description in another field.

You might use this feature, for example, if you have a code list that contains first names and you want the corresponding last names loaded in the descriptive element.

Complete the following steps to store an EDI code value description in a code storage field:

1. Double-click the element for which you want to use an EDI code value description. The Field Properties dialog is displayed.

Data Validation					
	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	1	8	String	X	X

2. In the Advanced section, click the Use Standard Rule check box.

- Click the Use Code option.

The screenshot shows a dialog box titled "Advanced" with the following controls:

- Use Standard Rule
- Select Update Constant Use Accum Loop Count Use Code
- Table: [Text Field] [Dropdown Arrow] [Edit Table... Button]
- Exclusive
- Store Field: [Text Field] [Dropdown Arrow]

- From the Table list, select the code list table against which the data in this field is validated. If the Tables list is empty, you need to load a code table. See *Loading a Code List Table from the Standard* on page 157 for more information.
- If you need to specify that, for compliance reasons, the field/element *must* contain one of the codes from the specified table (nothing else is valid for that field), click the Exclusive check box. (If you receive a code that is not in the specified table, the system flags it as an error.)
- From the Store Field list, select the field/element to which you want the description of the code item to be mapped.
- Click OK to add the standard rule to the field and exit the Field Properties dialog.

Now that you have set up the EDI code value description, and directed the system to display the description of the EDI code value in an element, you can format and customize that element. If you want to hide the EDI code value field, see *Hiding Fields* on page 221.

See *Using Code List Tables* on page 145 for information on using code list tables and using code item descriptions.

Customizing Print Forms

Print forms organize and format EDI data that is printed on a printout. You can customize the format of the EDI data as it is generated by Gentran:Director after you select the segments, groups, and elements for the print form. For example, you might want to customize the way data in fields are arranged on a printed printout, or the formats of numbers, dates, or times. You can even prevent fields in the EDI data from being printed on the printout.

This chapter explains how to customize print forms in Gentran:Director, including the following topics:

- Customizing field labels
- Hiding fields
- Formatting numbers
- Formatting dates and times
- Formatting strings

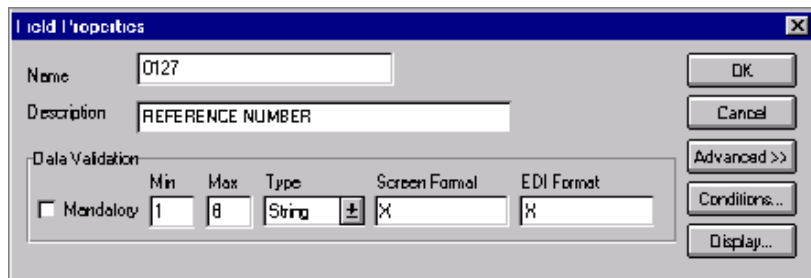
Customizing Field Labels

Field labels identify the contents of the fields included in a print form. The system uses the element name from the standards as a default to label each field. You can customize the default field label to be any name you choose.

We recommend that you label every field. If you decide *not* to label a field, follow the steps below, but delete the field label. For example, you might decide not to label the fields of a company address to be printed in a standard address format. In this instance, it is obvious that the name of the company is on the first line; the street address is on the second line; and the city, state, and zip code are on the third line, and do not need to be labeled.

Complete the following steps to customize a field label:

1. Double-click the element for which you want to customize the field label. The Field Properties dialog is displayed.



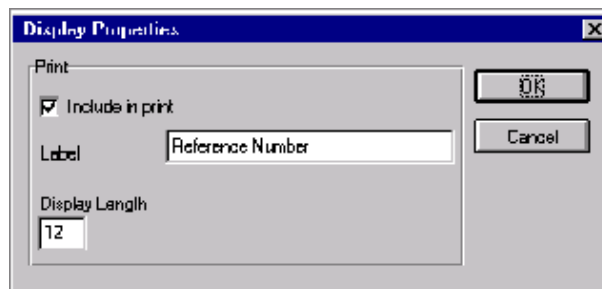
The Field Properties dialog box is shown with the following fields and options:

- Name: 0127
- Description: REFERENCE NUMBER
- Data Validation section:

	Min	Max	Type	Screen Format	EDI Format
<input type="checkbox"/> Mandatory	1	8	String	X	X

Buttons on the right: OK, Cancel, Advanced >>, Conditions..., Display...

2. Click DISPLAY... to access the Display Properties dialog.



The Display Properties dialog box is shown with the following fields and options:

- Print section:
 - Includes in print
 - Label: Reference Number
 - Display Length: 12

Buttons on the right: OK, Cancel

3. In the Label box, modify the text to customize the label.
4. Click OK to save the new label and exit the Display Properties dialog.
5. Click CLOSE to exit the Field Properties dialog.
6. Click the Generate Layout button on the Main Toolbar to refresh the display of the print form.

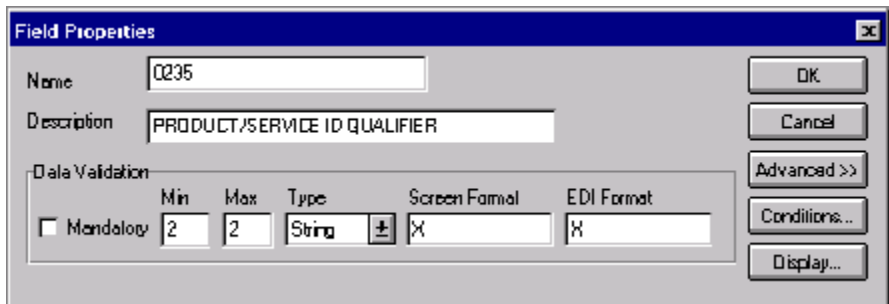


Hiding Fields

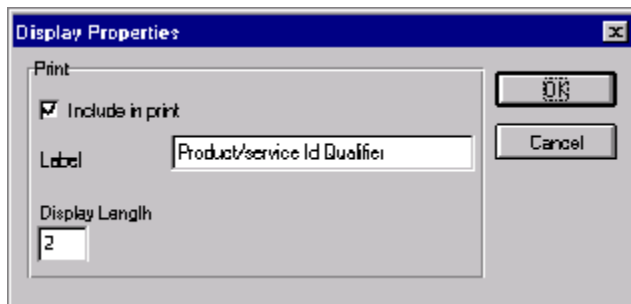
Usually, you want all EDI data to be printed on a printout. However, there may be times when you want to prevent a field of data from being printed (hide the field). For example, you might not want to print a field if it contains a qualifier that is explained by the label of an associated data field. You also might want to hide a field if it is a mandatory segment that must be kept activated, but the field does not need to be printed.

Complete the following steps to hide a field in a print form:

1. Double-click the element you do not want displayed on the printout. The Field Properties dialog is displayed.



2. Click DISPLAY... to access the Display Properties dialog.



3. Click the Include in print check box to turn the option off.
4. Click OK to exit the Display Properties dialog.
5. Click CLOSE to exit the Field Properties dialog.



6. Click the Generate Layout button on the Main Toolbar to refresh the display of the print form.

Formatting Numbers

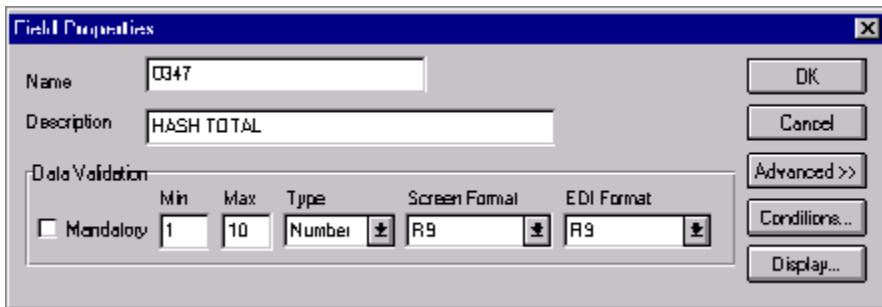
A number-type element contains either an integer or a real number. If a field is a number type, you must specify the format (either “R” or “N”) and the number of decimal places.

An “N”-formatted number is a number that has an implicit decimal point (e.g., “2.01” is formatted as “201”). An “R”-formatted number has an explicit decimal point (e.g., “2.01” is formatted as “2.01”), and truncates trailing zeroes. Whether you use the “N” or “R” format depends on your requirements. Regardless of whether you use the “N” or “R” format, you must also indicate the number of decimal places in the field.

You can format numbers from the EDI data that are displayed on a printout in various numeric formats.

Complete the following steps to format numbers in a print form:

1. Double-click the number element that you want to format. The Field Properties dialog is displayed.



2. In the Type list, verify that Number is selected, indicating that the element is an integer or real number.
3. From the Screen Format list, select the format for the number to be displayed on the printout.
 - Number formats beginning with “R” represent numbers with an explicit decimal point. For example, if the format you select is R2, then numbers from the EDI data are formatted with two places to the right of the decimal point (e.g., “2.01”).
 - Number formats beginning with “N” represent numbers with an implicit decimal point. The decimal point is not displayed on the printout, but the system still processes the number correctly. For example, if the format you select is N2, then numbers from the EDI

data are formatted with two places to the right of an implicit decimal point.

Valid number formats are the following:

- R0 Number formatted with an explicit decimal point and no decimal places
- R1 Number formatted with an explicit decimal point up to 1 decimal place
- R2 Number formatted with an explicit decimal point up to 2 decimal places
- R3 Number formatted with an explicit decimal point up to 3 decimal places
- R4 Number formatted with an explicit decimal point up to 4 decimal places
- R5 Number formatted with an explicit decimal point up to 5 decimal places
- R6 Number formatted with an explicit decimal point up to 6 decimal places
- R7 Number formatted with an explicit decimal point up to 7 decimal places
- R8 Number formatted with an explicit decimal point up to 8 decimal places
- R9 Number formatted with an explicit decimal point up to 9 decimal places
- N0 Number formatted with an implicit decimal point and no decimal places
- N1 Number formatted with an implicit decimal point up to 1 decimal place
- N2 Number formatted with an implicit decimal point up to 2 decimal places
- N3 Number formatted with an implicit decimal point up to 3 decimal places
- N4 Number formatted with an implicit decimal point up to 4 decimal places
- N5 Number formatted with an implicit decimal point up to 5 decimal places
- N6 Number formatted with an implicit decimal point up to 6 decimal places
- N7 Number formatted with an implicit decimal point up to 7 decimal places
- N8 Number formatted with an implicit decimal point up to 8 decimal places
- N9 Number formatted with an implicit decimal point up to 9 decimal places



Usually, the number formats in the Screen Format box and in the EDI Format box are the same. However, if there is an “N” format selected in the EDI Format box, you might want to select an “R” format in the Screen Format box so that the decimal point in the numbers is displayed on the printout.

4. In the EDI Format list, verify that the number format selected is the correct format for the EDI data expected in the file.



A number format must be designated for number-type elements in the EDI Format list, even for hidden fields.

5. Click OK to exit the Field Properties dialog.

Formatting Dates and Times

A date/time-type element contains either a date or a time. If an element is a date/time type, you must designate how the date or time is formatted on the printout. You can format dates and times from the EDI data that are displayed on a printout.

Complete the following steps to format dates and times in a print form:

1. Double-click the date/time element you want to format. The Field Properties dialog is displayed.

2. If the screen format for the date or time is longer than the length in the Max box, type a new maximum field length in the Max box. This value should be the maximum number of digits, including separators (?), that you expect for the date or time.



The value in the Max box should be the same as the value in the Display Length box on the Display Properties dialog (click DISPLAY... on the Field Properties dialog to access the Display Properties dialog).

3. In the Type list, verify that Dt/Tm is selected, indicating that the field value must be a date or a time.
4. From the Screen Format list, select the format for how you want the date or time to be formatted on the printout.

Valid date/time formats are the following:

MM/DD/YY	Two-digit month, separator, two-digit day, separator, last two digits of year (example: 12/15/97 for December 15, 1997)
----------	---

YYYYMMDD	Four-digit year, two-digit month, two-digit day (example: 19971215)
YYYY/MM/DD	Four-digit year, separator, two-digit month, separator, two-digit day (example: 1997/12/15)
DD/MM/YYYY	Two-digit day, separator, two-digit month, separator, four-digit year (example: 15/12/1997)
MM/DD/YYYY	Two-digit month, separator, two-digit day, separator, four-digit year (example: 12/15/1997)
HHMM	Two-digit hour, two-digit minutes (example: 0330 for thirty minutes past three o'clock)
HHMMSS	Two-digit hour, two-digit minutes, two-digits seconds (example: 033045 for thirty minutes and forty-five seconds past three 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)
DDMMYY	Two-digit day, two-digit month, last two-digits of year (example: 151297)
MMDDYY	Two-digit month, two-digit day, last two-digits of year (example: 121597)
DDMMYYYY	Two-digit day, two-digit month, four-digit year (example: 15121997)
MMDDYYYY	Two-digit month, two-digit day, four-digit year (example: 12151997)
YYDDD	Last two-digits of year, three-digit Julian day (example: 97349 for the 349th day of 1997)
DDDY	Three-digit Julian day, last two digits of year (example: 34997)
YYYYDDD	Four-digit year, three-digit Julian day (example: 1997349)
DDYYYY	Three-digit Julian day, four-digit year (example: 3491997)
YY/DDD	Last two digits of year, separator, three-digit Julian day (example: 97/349)
DDD/YY	Three-digit Julian day, separator, last two digits of year (example: 349/97)
YYYY/DDD	Four-digit year, separator, three-digit Julian day (example: 1997/349)

DDD/YYYY	Three-digit Julian date, separator, four-digit day (example: 349/1997)
MONTH	The month (example: December)
DAY	The day of the week (example: Friday)

5. In the EDI Format list, verify that the date or time format selected is the correct format for the EDI data expected in the file.



A date/time format must be designated for the date/time-type elements in the EDI Format list, even for hidden fields.

6. Click OK to exit the Field Properties dialog.

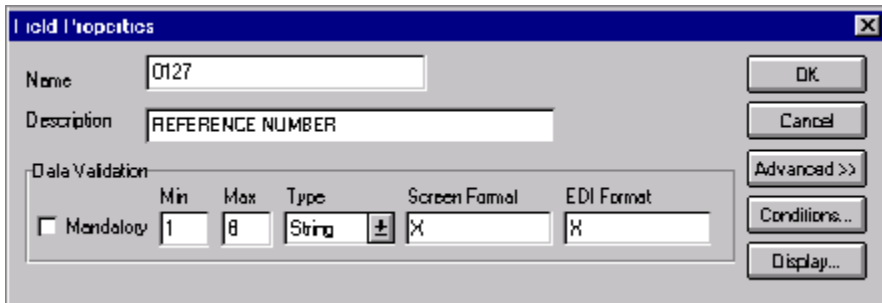
Formatting Strings

A string-type field or element contains one or more printable characters. If you specify that an 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/or numbers that are allowed to be used for a string-type element. You can use syntax tokens in the Screen Format and EDI Format fields of the Field Properties dialog to designate defined ranges of characters and/or numbers that are allowed to be used for a string-type element.

Complete the following steps to format strings in a print form:

1. Double-click the string element you want to format. The Field Properties dialog is displayed.



2. In the Type list, verify that String is selected, indicating that the field value must contain a string of printable characters.
3. In the Screen Format field, verify that the string format selected specifies how you want the string to be formatted on the printout.
4. In the EDI Format field, verify that the string format selected is the correct format that is written to the EDI file.



A string format must be designated for string-type elements in the EDI Format field, even for those that you hide.

5. Click OK to exit the Field Properties dialog.

You can create syntax tokens to define ranges of characters and/or numbers that are allowed to be used for a string-type field or element. See *Using the String Type* on page 45 for information on how to create or edit syntax tokens.

Formatting Screen Entry and Print Forms

After you customize screen entry and print forms, you can format the organization and size of the fields in the form. This chapter explains how to format screen entry and print forms, including the following topics:

- Selecting and grouping fields
- Resizing fields
- Arranging fields
- Resizing and positioning a frame
- Resizing drop-down combo boxes
- Setting tab sequences
- Adding and positioning static text
- Modifying frame titles
- Modifying list box names
- Adding column headings to list boxes

Selecting and Grouping Fields

You must select or group fields and field labels to be able to resize or arrange them. This section includes the following topics:

- Selecting one or more fields or field labels
- Grouping fields and field labels

Selecting One or More Fields or Field Labels

You must know how to select one or more items (fields and field labels) in a form to be able to move, resize, align, or change the spacing between them.

To select a field or a field label, click anywhere on the field or label. The field or field label is outlined by diagonal lines. The selected field also displays *handles*, which mark the corners and sides.

The screenshot shows a form titled "810 Invoice" with various input fields. The "Invoice Date" field is selected, indicated by a diagonal border and a hatched pattern. To the right of the form, there are three legends: "* = Required Field", "** = Required Field if Cash Discounts are Given", and "\$ = Required Field if Units Shipped is Used". The form includes fields for Invoice Date, Invoice Number, P.O. Number, Qualifier, ID Code, Ship-To Name, Address, City, State, and Zip Code.

If you want to move, resize, align, or adjust the spacing of several fields or field labels at the same time, you must select all those items first.

Complete the steps below to select two or more fields and field labels:

1. Select a field or field label in a form.



The Lock labels function enables you to specify that when you click a field (in the Layout Window), its corresponding label is selected at the same time. Click Lock Labels on the Main Toolbar to activate this function.

2. Press **CTRL** and click another field or field label.

Repeat this step for all other fields and field labels in the form you want to select. The last item you select is outlined by diagonal lines. This item is called the *reference*. All other items you selected are now outlined by a solid line. When you arrange a group of items, the reference item is significant because the alignment, size, and spacing between all other items in the group is relative to the alignment, size, and spacing of the reference item.



When you select two or more items, make sure the item you want as the reference is the last item you select.

This is the reference

If you select an item by mistake, you can easily deselect the item. To deselect an item, press **CTRL** and click the item you want to deselect.



If the item you deselect is the reference, the item that was selected immediately prior to the last item (the reference) becomes the new reference.

Grouping Fields and Field Labels

You can group fields and field labels that are in the same area of a form.

To group fields and field labels, click and drag the mouse across the area of the form containing the items you want to select. As you drag the mouse, the grouped area is designated by dotted lines. When you release the mouse button, all items *completely* contained within the selection area are selected. The last item in the group is the reference. All the selected items are now a group, and can be moved together, as well as resized or spaced relative to the reference.

The screenshot shows a form titled "810 Invoice" with a dotted selection box around the top three fields: Invoice Date, Invoice Number, and P.O. Number. A legend on the right side of the form explains the asterisk symbols: a single asterisk (*) indicates a "Required Field", a double asterisk (**) indicates a "Required Field if Cash Discounts are Given", and a dollar sign (\$) indicates a "Required Field if Units Shipped is Used". The form includes the following fields: Invoice Date, Invoice Number, P.O. Number, Qualifier (with a dropdown arrow), ID Code, Ship-To Name, Address, City, State, and Zip Code. A mouse cursor is positioned over the ID Code field, and a dotted line extends from the legend to it.

Resizing Fields

When you generate a form, each field is automatically sized to the length specified in the Display Length field on the Display Properties dialog. If you change the display length of a field after the form is generated for the first time, you must resize that field in the Layout Window. You can resize a field in one of these two ways:

- Use the Size to Length function
- Use the mouse to reshape the field manually

In addition, you can resize groups of fields or field labels so that they are all the same size.



Resizing the field graphically maintains the integrity of the form.

Using Size to Length

You can resize one or more fields according to the display length specified on the Display Properties dialog for each selected item.

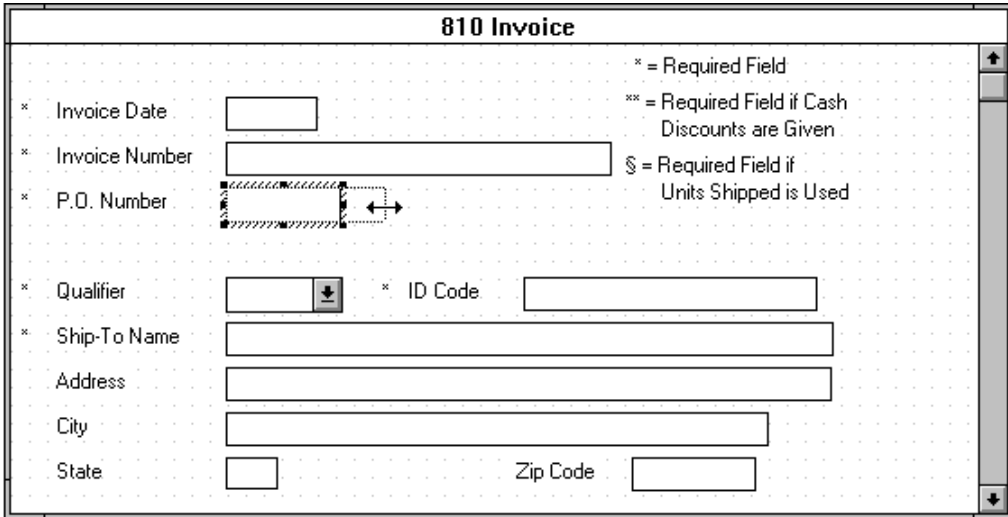
Complete the following steps to resize a field or group of fields with the Size to Length function:

1. Select the fields you want to resize. See *Grouping Fields and Field Labels* on page 246 for more information on grouping items.
2. Select Size to Length from the Display menu to resize according to the display length specified on the Display Properties dialog for each selected item.

Resizing a Field Manually

Complete the following steps to resize one field manually:

1. Select a field in a form.
2. Click and drag one of the handles.



Resizing Groups of Fields Manually

You can resize several fields at the same time if you want them all to be the same size. For example, you might want to resize a group of fields that are similar in size to give the form a more consistent appearance. See *Grouping Fields and Field Labels* on page 246 for more information on grouping items.

Complete the following steps to resize a group of fields manually:

1. Select the fields you want to resize.



Ensure that the field that is to be used as the reference for resizing all other selected fields is selected last.

2. Select Size Equally from the Display menu to equally resize all selected items. The reference is used as the basis for determining the size of all other selected items.

Arranging Fields

To arrange the fields and field labels in a screen entry or print form, you must move, align, and space the fields and field labels in the Layout Window. This section includes the following topics:

- Changing grid settings
- Using alignment lines
- Moving fields and field labels
- Aligning fields and field labels
- Spacing fields and field labels

Changing Grid Settings

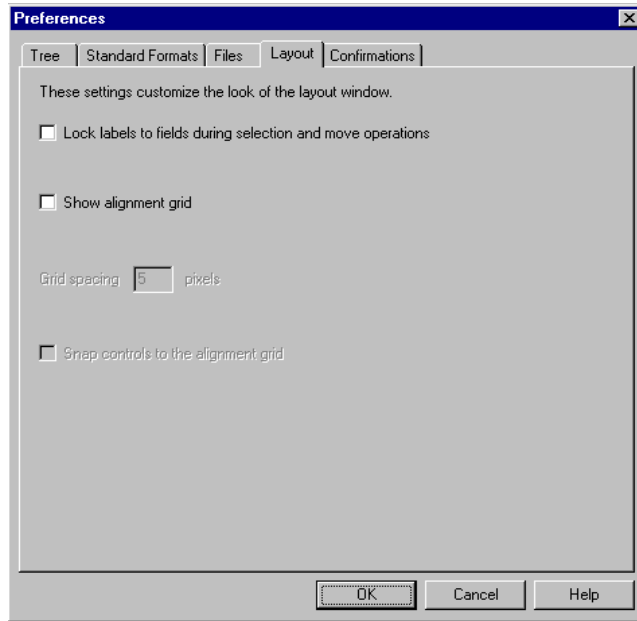
Gentran:Director provides a grid, indicated by the dots in the background of the form, that you can use to assist you in formatting the layout of a screen entry or print form. The dots on the grid form horizontal and vertical lines that guide you in aligning and spacing fields and field labels. The dots on the grid are evenly spaced, with the spacing between them measured in pixels. The grid is not printed (for print forms) – it is only displayed in the background of the template to aid you in arranging fields and field labels.

Sometimes it is useful to have items snap to the grid when you move them. With the “Snap controls to the alignment grid” feature turned on, as you move, align, or space items, they snap to the lines of the grid. This feature can help you align items with one another or evenly space them.

Complete the following steps to change the grid settings:

1. Select Preferences... from the Options menu to display the Preferences dialog (the Tree tab is displayed by default). The Preferences dialog is a property sheet that enables you to set global defaults for Gentran:Director.

2. Click the Layout tab to access the layout options.



3. If you want the grid to be displayed, click the “Show alignment grid” check box to turn the option on. The “Snap controls to the alignment grid” check box and the Grid spacing box are activated. If you do not want the grid to be displayed, continue with step 6.
4. If you want the fields and field labels to snap to the grid, click the “Snap controls to the alignment grid” check box to turn the option on.



If you are using alignment lines, you should not use the “Snap controls to the alignment grid” function, as this hinders the effectiveness of the alignment lines.

5. If you want to change the spacing between the dots on the grid, type the number of pixels in the Grid spacing box. The default setting is five pixels. Increasing the number of pixels increases the spacing between the dots; decreasing the number of pixels decreases the spacing between the dots.
6. Click OK to exit the Preferences dialog.

Using Alignment Lines

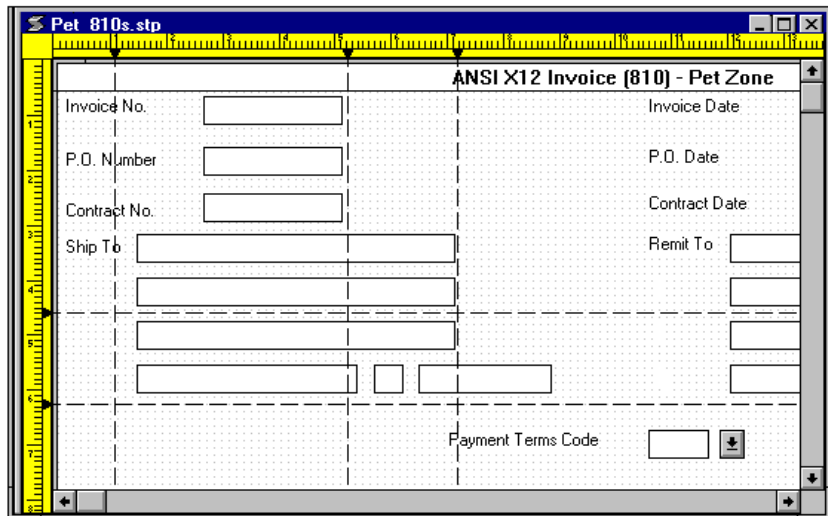
Genran:Director enables you to create alignment lines (lines that guide you in aligning the components of the Layout Window).

Complete the following steps to use alignment lines:

1. Double-click the vertical or horizontal ruler just below the graduation where you want the alignment line to be located. The alignment line is displayed in the Layout Window.



If you are using alignment lines, you should not use the Snap Controls To Grid function (on the Grid Settings dialog), as this hinders the effectiveness of the alignment lines.



2. After an alignment line is created, you can move it anywhere in the Layout Window by clicking and dragging the black triangle that connects the alignment line to the ruler.
3. Double-click an alignment line to remove it.

Moving Fields and Field Labels

You can move fields or field labels from their default positions to other locations in a form. You can move either individual items or groups of items.

If you intend to resize fields, we recommend that you do so before moving them, since the arrangement of the fields might depend on the size of the

fields. For example, you might be able to arrange more fields in a row if the fields are smaller. Therefore, you should resize the fields before you arrange them, so you can accurately determine how many fields can fit in a row. See *Using Size to Length* on page 247 or *Resizing a Field Manually* on page 248 for details on resizing fields.

To move a field or field label, click and drag it to the new location in the form. If you are moving a group of selected items, all items in the group stay in position relative to each other as they are moved from one location to the next.

Aligning Fields and Field Labels

When fields and field labels are created, they are automatically left-aligned in the screen entry or print form, with one field per line. However, you can align particular fields and field labels with one another in different ways to change the way they are arranged in the form.

If you intend to resize or move fields or field labels, we recommend that you do so before you align them. If you align fields first, then move or resize them, the fields might move out of alignment. See *Using Size to Length* on page 247 or *Resizing a Field Manually* on page 248 for details on resizing fields. See *Moving Fields and Field Labels* on page 251 for more information on moving fields or field labels.

Complete the following steps to align fields and field labels:

1. Select two or more fields that you want to align in a form.



Ensure that the field or field label to be used as the reference for aligning all other selected items is selected last.

2. Select Align Controls from the Display menu.
3. From the Align Controls cascading menu, select how you want the selected items to be aligned. Valid aligning methods are the following:

Top Aligns all selected items with the top of the reference item.

Bottom Aligns all selected items with the bottom of the reference item.

Left Aligns all selected items with the left side of the reference item.

Right Aligns all selected items with the right side of the reference item.



Use caution in selecting the alignment method to ensure that items do not overlay each other. Use Top or

Bottom alignment for aligning a horizontal row of items; use Left or Right alignment for aligning a vertical column of items.

Spacing Fields and Field Labels

When fields and field labels are resized and moved, the spacing between them can become uneven. You can change the spacing between items to give the form a more consistent appearance.

If you intend to resize or move fields or field labels, we recommend that you do so before you space them. If you space fields first, then move or resize them, the spacing between the fields might change. See *Using Size to Length* on page 247 or *Resizing a Field Manually* on page 248 for details on resizing fields. See *Moving Fields and Field Labels* on page 251 for more information on moving fields or field labels.

Complete the following steps to evenly space several fields and field labels:

1. Select three or more items that you want to be evenly spaced in a form.



Ensure that the field or field label to be used as the reference for evenly spacing all other selected items is selected last.

2. Select Space Evenly from the Display menu.
3. From the Space Evenly cascading menu, select the type of spacing you want for the selected items. Valid spacing methods are the following:

Across Evenly spaces all selected items with the next selected item to the left or right (horizontal spacing).

Down Evenly spaces all selected items with the next selected item above or below (vertical spacing).

Resizing and Positioning a Frame

You can define how a frame is sized and positioned in the Document Editor by resizing and positioning that frame in the Forms Integration subsystem. When a form is created, the default location for frames is the upper left corner of the Layout Window. When a screen entry translation object is opened in Document Editor, its frame is sized and positioned the same as it was when the source form was last compiled. For example, if you want the frame to be centered on the user's desktop, then you must center the frame in the Layout Window in the Forms Integration subsystem before compiling it.



When you resize the frames of a screen entry form, you must resize based on the resolution of the target monitor (i.e., the monitor on which the end user views the translation object). For example, if you size the frames of a screen entry form on a Super VGA monitor, the compiled translation object may not fit in the display area of a VGA monitor.

Complete the following steps to resize and position a frame:

1. To resize the frame, click and drag a side or corner.
2. To position the frame in the Layout Window, click and drag the frame title to the new position in the Layout Window.

Resizing Drop-Down Combo Boxes

A drop-down combo box consists of a text box and a drop-down list. When the combo box is closed, the text box and the drop-down arrow are displayed. When the combo box is opened, the available field values (and a scroll bar, if there are more values than can fit in the drop-down list) are displayed. You can resize a drop-down combo box in a screen entry form to make the size appropriate for the field values contained in the list.

Please remember the following tips about drop-down combo boxes:

- When you change the width of the combo box, the width of the text box changes. (The width is the same whether the drop-down combo box is opened or closed.)
- When you change the height of the combo box, the height of the drop-down list is changed. When closed, the drop-down combo box is long enough to contain one item. When opened, the drop-down combo box is long enough to contain several items.

The following are some guidelines for resizing drop-down combo boxes:

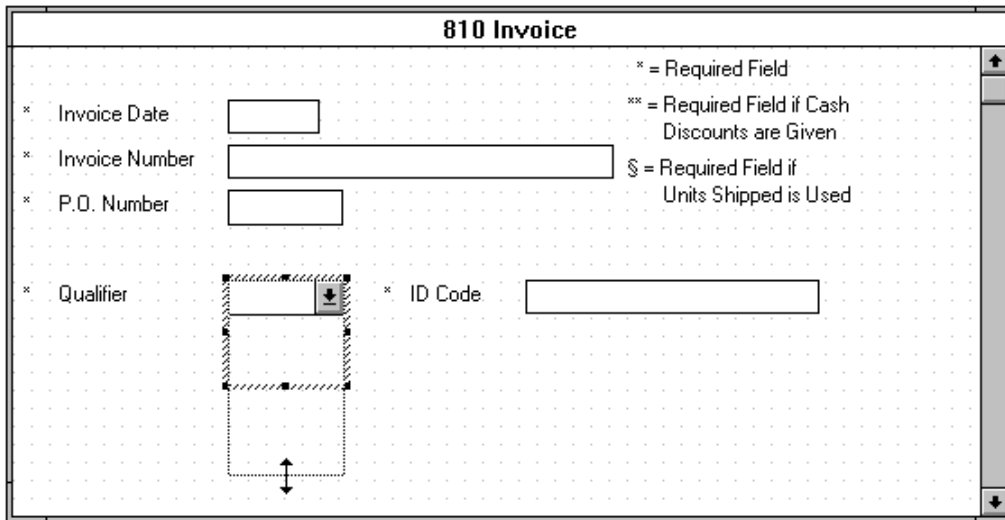
- The width of the drop-down combo box should be appropriate to the width of the values available in the list. You should make the width a few spaces wider than the longest value available. For example, if the values available are two-character codes, the combo box does not need to be twenty characters wide. However, if the values in the list consist of several words or a long string of characters, then the combo box should be wide enough for users to read the longest field value.
- The height of the drop-down combo box should be appropriate to the number of values available in the list. For example, if there are just four values available, you might make the drop-down combo box long enough to contain all of the values, so that users do not have to scroll through a short list. However, if there are one hundred values available, you might make the drop-down combo box long enough to view up to eight values at once.



The scroll bar is automatically added to the drop-down combo box when the number of available values exceeds the vertical space provided in the combo box.

Complete the following steps to resize a drop-down combo box:

1. Click the drop-down combo box you want to resize in the form.
2. Drag a handle of the combo box.



Setting Tab Sequences

The tab sequence is the order in which the cursor moves from field to field in a screen entry form when a user presses **TAB**. When you create a screen entry form, the system automatically positions the fields and sets the tab sequence for those fields. However, you usually want to modify the tab sequence of the fields to meet your screen entry requirements.



You can set tab sequences for screen entry forms *only*.

Complete the following steps to set the tab sequence of fields in a screen entry form:

1. Select Set Tab Order from the Display menu. Numbers are displayed in the fields in the form to indicate the current tab sequence.

810 Invoice

* = Required Field
 ** = Required Field if Cash Discounts are Given
 \$ = Required Field if Units Shipped is Used

* Invoice Date **1**

* Invoice Number **2**

* P.O. Number **3**

* Qualifier **4** * ID Code **5**

* Ship-To Name **6**

Address **7**

City **8**

State **9** Zip Code **10**

2. Click the fields sequentially, in the order you want the tab sequence to be set. For example, if you want a field that is currently set as third in the tab sequence to become first in the tab sequence, then click that field first. The field that was first in the tab sequence is now the second, the second is now the third, etc. After you have clicked all the fields, the tab sequence numbers are no longer displayed in the fields.



If you want to change the tab sequence for some of the fields without having to click them all, then select Set Tab Order from the Display menu when you have finished setting the tab sequence for the fields you want to change. The tab sequence numbers are no longer displayed.

Adding and Positioning Static Text

Static text is text that is always included in the screen entry or print form. Examples of static text include legends and column headings. Field labels, which usually accompany fields to identify the contents of the fields, are not considered static text.

Complete the following steps to add and position static text to a screen entry or print form:

1. Click Add Text from the Display menu. Text is added to the current frame, displayed in the upper left corner of the frame. If the template is larger than the frame (there is a scroll bar on the right side of the frame), the static text is displayed in the upper left corner of the part of the template that is displayed, not at the top of the template.
2. Type the static text in the new selected text field.
3. To move the static text, click and drag it to a new location in the frame.

Modifying Frame Titles

The system automatically titles the frames in a screen entry form when the form is created, based on the name of the file, segments, or groups. You can modify the frame title of any repeating group (EDI file, segment, or group) in a screen entry form. For example, if you use several purchase order forms, each for a different trading partner, you might give each purchase order form a slightly different title to distinguish one form from the others.

Complete the steps below to modify a frame title:

1. Highlight the form component (EDI file, segment, or group) of the frame title you want to modify.
2. Select Properties from the Edit menu to display the appropriate dialog for the form component you selected.
 - If you select the EDI file, the File Format Properties dialog is displayed.
 - If you select a segment, the Segment Properties dialog is displayed.
 - If you select a group, the Group Properties dialog is displayed.
3. In the Frame Caption box, type the new frame title.
4. Click the OK button. The new frame title is displayed in the appropriate frame.

Modifying List Box Names

When Gentran:Director identifies a group in a transaction set, it creates a separate frame to contain the data within the group. It also creates a list box in the parent frame to allow the screen entry translation object user access to the items in the group. The system automatically sets the names for list boxes in a form when the form is created, based on the name of the segment or group. You can modify the names of the list boxes in screen entry forms. For example, you might want to modify a list box name to give it a name that is more meaningful to you than the default name that the system gives it.

Complete the steps below to modify a list box name:

1. Highlight the form component (segment or group) containing the list box name you want to modify.
2. Select Properties from the Edit menu to display the appropriate dialog for the form component you selected.
 - If you select a segment, the Segment Properties dialog is displayed.
 - If you select a group, the Group Properties dialog is displayed.
3. In the List Caption box, type the list box name.
4. Click the OK button. The new list box name is displayed in the form.

Adding Column Headings to List Boxes

A single list box in a form may contain two or more columns of data, depending on which fields have been designated list box fields. Although the system automatically generates field labels for list boxes, it does not automatically generate column headings for them. If you want the columns in a list box to be labeled, you must add the column headings by using static text.

Complete the following steps to add column headings to a list box:

1. Click Add Text from the Display menu. Text is added to the current frame, displayed in the upper left corner of the frame. If the template is larger than the frame (there is a scroll bar on the right side of the frame), the static text is displayed in the upper left corner of the part of the template that is displayed, not at the top of the template.
2. Type the static text in the new selected text field.
3. To move the column heading, click and drag the column heading to a new location in the frame.

Error Messages

The Gentran:Director error messages and other informational messages are noted in the Compile Errors dialog when you compile the translation object or in the Error section of the Extended Rules dialogs when you compile an extended rule containing errors prior to compiling the translation object, and when you commit an erroneous action in Gentran:Director. 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. This chapter explains how to determine the action you should take when you receive an error message, including the following topics:

- **Compile Error Messages** (when you compile the translation object or compile an extended rule containing errors prior to compiling the translation object)
- **Gentran:Director Error Messages** (if you perform an erroneous action in Gentran:Director)

Compile Error Messages

The Compile Error Messages are displayed in the Compile Errors dialog if you compile a translation object with errors. Or, error messages are displayed in the Error section of an Extended Rule dialog 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.

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.

100 Mandatory Element Missing

Description: An element that the translation object designated as “Mandatory” was not created in an Outbound document or was not received in an Inbound document.

1000 expected '.'

Description: The rule does not have the required “.” between a group name and a field name.

User Action: Insert a “.” between the group name and field name.

1001 no statement to compile

Description: The rule is does not contain any statements.

User Action: Add a statement or statements to the rule.

1002 unexpected end of program

Description: The rule was not complete.

User Action: Finish the rule.

1003 expected ','

Description: The rule does not have the required “,” between parameters.

User Action: Insert a “,” between parameters.

1006 no statements to compile

Description: The body of an IF/ELSE or WHILE condition was empty.

User Action: Complete the body of the unfinished condition.

1008 expected '#'

Description: The rule does not have the required “#” before a field name.

User Action: Insert a “#” before the field name.

2000 group ... undefined

Description: The rule references a group that does not exist.

User Action: Change the reference to an existing group or delete the reference.

2001 ... is not a member of ...

Description: The rule references a field that does not belong to the specified group.

User Action: Change the reference to an existing field in the specified group.

2002 insufficient indices to access group ...

Description: The rule does not give the full addressing for a group.

User Action: Complete the addressing for the group.

2003 too many indices to access group...

Description: The rule uses too many addresses for the group.

User Action: Address the group correctly.

2004 ... has not been defined

Description: The rule references an undefined variable.

User Action: Define the variable in the declarations section.

2005 out of temporary variables

Description: The rule could not be compiled because some expressions are too complex.

User Action: Simplify the expressions and compile the rule again.

2008 field type unknown

Description: The compiler was unable to determine the type of a field.

User Action: Verify that a Data Type is selected for this field.

2009 cannot reference a local field with no current group

Description: The rule (probably pre- or post-session) references a local field, but is not associated with any group.

User Action: Reference the field using the proper addressing.

2010 instances of '%1' are transient and cannot be accessed

Description: The rule references an output group using full addressing. This form of addressing is only appropriate for input groups.

User Action: Reference the output group with the proper address.

2011 no field specified

Description: The rule omits a field reference.

User Action: Add the field reference to the rule.

20700 only one binary data and one binary length field are permitted

Description: You have more than one binary data and/or more than one binary length element in one segment.

User Action: Remove the additional binary data and/or binary length elements from the segment.

20701 binary length must precede binary data

Description: The binary length element must be sequenced *before* the binary data element in the segment so the translator knows how much data to expect.

User Action: Move the binary length element to before the binary data element.

20702 incomplete binary data

Description: You marked a segment as binary, but did not include either a binary length element or a binary data element (or both).

User Action: Add a binary length element and/or a binary data element to the segment.

20703 group [group] has no active child objects

Description: You tried to compile the translation object, but the specified group is empty.

User Action: Activate at least one child object in the group.

2100 ... is not an array variable

Description: The rule uses array indexing for a variable that is not an array.

User Action: Use the proper indexing for the variable.

2101 ... : array index required

Description: The rule uses an array variable without using the necessary array indexing.

User Action: Add the necessary indexing to the array variable.

2102 ... : array overflow

Description: The rule uses an invalid array index.

User Action: Use the proper array index for the array variable.

2103 only one wildcard index is permitted

Description: The rule uses more than one wildcard index.

User Action: Only one wildcard index is permitted per rule.

2104 a wildcard index must be specified

Description: The rule does not specify a wildcard index when required.

User Action: Add a wildcard index where necessary.

2200 expected a string

Description: A required string or string variable is not supplied.

User Action: Add the required string or string variable.

2201 string overflow

Description: A string overflow occurred.

User Action: Verify that the size of a target string is equal to or greater than the source string.

3000 array size expected in declaration of ...

Description: Invalid array declaration.

3001 declaration of ... missing ']'

Description: Invalid array declaration.

3002 string size expected

Description: Invalid string declaration.

3003 string size missing ']'

Description: Invalid string declaration.

3004 variable name expected

Description: Invalid variable declaration.

3005 ... already defined

Description: Two variables with the same name are defined at the same scope.

User Action: Rename one of the two variables.

3006 expected an accumulator number, found ...

Description: Invalid accumulator reference.

3007 ... is not a valid accumulator number

Description: Invalid accumulator reference.

4000 expected a numeric expression, found ...

Description: You specified something other than the expected numeric expression.

User Action: Specify the correct numeric expression.

4001 expected a term, found ...

Description: You specified something other than the expected term.

User Action: Specify the correct term.

4002 expected '+' or '-'

Description: You specified something other than the expected “+” or “-”.

User Action: Specify “+” or “-”.

4003 expected '*' or '/'

Description: You specified something other than the expected "*" or "/".

User Action: Specify "*" or "/".

4004 expected ')'

Description: You specified something other than the expected ")".

User Action: Specify ")".

4005 expected a factor, found ...

Description: Invalid numeric expression.

4006 expected '('

Description: You specified something other than the expected "(".

User Action: Specify "(".

4007 ... is of incorrect type

Description: The specified expression is of an incorrect type.

User Action: Specify the correct type for the expression.

4008 expected a relational operator

Description: You specified something other than the expected relational operator.

User Action: Specify the correct relational operator.

4009 missing argument

Description: A required parameter was omitted.

User Action: Add the required parameter.

4010 assignment expected

Description: The assignment operator was omitted from an assignment statement.

User Action: Add the correct assignment operator to the statement.

4011 operator ... requires two arguments

Description: Only one parameter was supplied for a binary operator.

User Action: Supply a second parameter for the binary operator.

4100 expected a date, found ...

Description: A date is required but not supplied.

User Action: Specify a date.

4101 expected a date modifier, found ...

Description: A date modifier was required but not supplied.

User Action: Specify a date modifier.

4102 ... is not a date

Description: Invalid date expression.

5000 THEN expected

Description: The compiler expected a THEN condition.

5001 DO expected

Description: The compiler expected a DO condition.

5002 END expected

Description: The compiler expected a END condition.

5003 IF expected

Description: The compiler expected a IF condition.

5004 FROM expected

Description: The compiler expected a FROM condition.

5005 INTO expected

Description: The compiler expected a INTO condition.

5006 END without BEGIN

Description: An END statement was found without a corresponding BEGIN statement.

User Action: Insert a BEGIN statement in the correct location.

5018 too many parameters

Description: Too many parameters were supplied for a function.

User Action: Remove the unnecessary parameters.

5019 too few parameters

Description: Too few parameters were supplied for a function.

User Action: Add the necessary parameters.

6000 expected a database table name

Description: The compiler expected a database table name.

User Action: Insert a database table name in the necessary location.

6001 expected a database column name

Description: The compiler expected a database column name.

User Action: Insert a database column name in the necessary location.

6002 too many column receivers specified

Description: The number of columns specified did not match the number of receivers in a SELECT statement.

User Action: Correlate the number of columns specified in the SELECT statement to the number of receivers.

6003 too few column receivers specified

Description: The number of columns specified did not match the number of receivers in a SELECT statement.

User Action: Correlate the number of columns specified in the SELECT statement to the number of receivers.

6004 WHERE expected

Description: The compiler expected a WHERE statement.

6005 expected a database key

Description: An invalid database key was specified in a WHERE statement.

User Action: Specify the correct database key in the WHERE statement.

6006 WHERE not allowed with this database table

Description: A WHERE statement is not necessary to access the specified database table.

User Action: Remove the WHERE statement.

6007 invalid key combination

Description: The combination of keys specified in the WHERE statement is not a valid unique compound key to the table.

User Action: Specify a valid combination of keys in the WHERE statement.

Gentran:Director Error Messages

The Gentran:Director 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.

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.

A code list entry must have a code value

Description: You attempted to add a code without an associated code value.

A code list must have an element id

Description: 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

Description: You attempted to create a code list with the same element ID as an already existing code list.

User Action: Either change the element ID or delete the original code list.

A condition field is required

Description: You established a conditional relationship that requires a condition field, but did not specify a condition field.

A data type is required.

Description: You did not specify the Data Type of a field.

A field must have a name that is unique within its parent group

Description: You tried to give a field a name that is already in use by another field within the same group (not necessarily the same segment).

A group must have a unique name

Description: You tried to give a group a name that is already used by another group or segment.

A group or record with a maximum usage of 1 cannot be split or promoted

Description: You attempted to split or promote a single-occurrence group or segment. This is not a valid action.

A key field has been selected but no key value has been specified.

Description: You established a key field without specifying a key value

A Memory Exception has occurred.

Description: A system error occurred.

User Action: Exit Gentran:Director and restart Windows.

A name must be entered

Description: You did not specify a name for an object.

A problem occurred while attempting to close loop This is probably due to incorrect standards.

Description: An error occurred when reading from the standards.

A record must have a unique name

Description: You tried to give a segment a name that is already used by another group or segment.

A Resource Exception has occurred.

Description: A system error occurred.

User Action: Exit Gentran:Director and restart Windows.

A serious error was encountered whilst accessing the clipboard and the action was abandoned

Description: The system aborted a cut, copy, or paste operation because a serious error occurred.

User Action: Perform the cut, copy, or paste operation again.

A system error was encountered while compiling the translation object

Description: While compiling a translation object, a system-related problem, such as lack of memory, prevented the compile from completing.

User Action: Close unnecessary applications to free memory, and reboot your computer (if necessary), and then recompile the translation object.

A TDF could not be generated due to a lack of available memory. Either restart Windows or close down other applications, then try again

Description: While generating a TDF, the system ran out of memory.

User Action: Exit Gentran:Director, restart Windows and regenerate the TDF.

An accumulator must be chosen for this entry

Description: You did not specify the accumulator to be used in a Use Accum standard rule.

An invalid usage count has been entered

Description: You entered an invalid usage count for a segment or group.

No actions have been chosen for this accumulator entry

Description: You selected an accumulator but did not specify any actions for it in a Use Accum standard rule.

No alternate accumulator has been selected

Description: 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.

No character ranges have been specified for this token.

Description: You did not specify the characters allowed for a syntax token.

No fields have been used in this relationship

Description: You established a conditional relationship but did not specify the fields involved.

No more accumulator entries can be created

Description: You attempted to create more than six accumulator entries in a Use Accum standard rule.

No more field mappings can be created

Description: You attempted to create more than eight field mappings for a Select standard rule.

The constant ID must be unique.

Description: You entered a literal constant identifier in the ID box that is already used in an existing constant.

User Action: Type a unique literal constant identifier in the ID box.

The contents of the clipboard cannot be pasted into the translation object at this point

Description: Either the Clipboard does not contain copied or cut data, or you are trying to paste EDI information into a positional file format or paste position information into an EDI file format.

The file format cannot be deleted

Description: You tried to delete a file format object.

The Maximum Usage must be greater then zero and not less than the Minimum Usage.

Description: You entered an invalid maximum usage count.

The number of entries to be split must be greater than zero and less than the maximum number of entries in the original

Description: You entered an invalid number in the Split dialog. The number to be split must be greater than zero and less than the maximum number of entries in the original.

The translation object could not be compiled

Description: The translation object is not ready to be compiled.

The token code must be unique.

Description: You attempted to create a syntax token with the same token identifier as an existing syntax token.

User Action: Type a unique token identifier in the Token box.

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.

Description: 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.

These fields cannot be linked because the loop which the input field belongs to is shallower than the loop which the output field belongs to.

Description: 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.

This code already exists. You must use another code or delete the original code first

Description: You attempted to add a code to a code list that already contained that code.

This field cannot be linked to another field

Description: Due to an unknown error, Gentran:Director was unable to begin creating the link.

This file is not a valid GENTRAN translation object

Description: You attempted to open (load) a file that is not a translation object.

You have entered an invalid decimal separator or character code.

Description: You entered an invalid decimal separator.

You have entered an invalid element delimiter character or character code

Description: You entered an invalid element delimiter.

You have entered an invalid Pad character or character code

Description: You entered an invalid pad character for a positional field.

You have entered an invalid release character or character code

Description: You entered an invalid release character.

You have entered an invalid segment delimiter character or character code

Description: You entered an invalid segment delimiter.

You have entered an invalid sub-element delimiter character or character code

Description: You entered an invalid sub-element delimiter.

You have entered an invalid tag delimiter character or character code

Description: You entered an invalid tag delimiter.

You must enter a description of the translation object.

Description: You attempted to save the translation object without entering the name of the translation object description on the Translation Object Details dialog.

User Action: Type the translation object description in the Description box on the Translation Object Details dialog. This description must be unique because it is used by the system to identify the form. We recommend that you use the following identifying characteristics: the partner that this form is used for, the standard, the version, the type of transaction this form uses, and the type of form).

You must enter a subtable name.

Description: For a Select or Update standard rule, the selected table requires you to specify a table name in the Sub Table box.

You must enter a valid character range.

Description: You entered an invalid character range for a syntax token.

You must enter a valid syntax token.

Description: You did not enter a valid token identifier in the Token box when creating or editing a syntax token.

You must enter a value for this constant.

Description: You did not specify a value for the constant in the Value box when creating or editing a syntax token.

You must enter an ID.

Description: You did not specify a literal constant identifier for the constant in the ID box when creating or editing a syntax token.

You must enter the name of the author of this translation object.

Description: You attempted to save the translation object without entering the name of the translation object author on the Translation Object Details dialog.

You must select a constant type.

Description: You did not select a constant type from the Type list when creating or editing a constant.

You need to specify Input and Output File Types for the new translation object

Description: You clicked OK on the Create New Translation Object dialog without selecting both the Input Format Type and Output Format Type.

Glossary

?In Documents This Gentran:Director 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:Director 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, *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.

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 the Application Integration subsystem. In Gentran:Director 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.

- 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:Director 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 form and creates a translation object. The form that you create using Gentran:Director is a *source form*. When that source form is compiled, the result is a *compiled translation object*. This translation object must be registered with the Gentran:Director 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 relational conditions.
- 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 *unless* you specifically override them.
- dialog box** A dialog box (or dialog) 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.

double-click This indicates that you should position your cursor on top of the designated item and quickly press and release your primary mouse button twice. 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.

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 File Format Window The EDI File Format Window contains the EDI file format. When you click a segment or group in the EDI file format, the corresponding frame is displayed in the Layout Window. When you click an element in the EDI file format, the corresponding layout component is highlighted in the Layout Window.

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.

Data Element	Definition
Component data element	A simple data element that belongs to a composite data element. Component data elements are also called sub-elements.

envelope 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.

There are three levels of envelopes, as follows:

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.



A communications session could easily include a number of interchange envelopes. For this reason, the standards-setting bodies are considering a fourth level of envelope to cover an entire communications session.

equalize 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.

export map 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.

export translation object See export map.

extended rules These rules enable you to use a Gentran:Director proprietary programming language to perform virtually any mapping operation you require.

- field** 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.
- fixed-format file** See application file.
- font** This feature enables you to globally change the font that is used in the display of all maps. The default font that Gentran:Director 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.
- form** A form is a set of instructions you define in the Forms Integration subsystem that indicates how the system should format data
- frame** A frame contains the groups, repeating segments, and elements at that level (single segments are not represented on the frame). Gentran:Director uses frames to format the EDI file for the translation object.
- 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:Director.
- 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 import map.

- In Documents** This Gentran:Director 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.
- 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 you partners. To translate inbound data, you need to create an export map.
- In Drawer** This Gentran:Director 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:Director 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.
- Layout Window** The Layout Window contains the actual format of the translation object. Each group (with a maximum usage greater than one) and repeating segment has a corresponding frame in the Layout Window. When you click a segment or group in the EDI file format, the corresponding frame is displayed in the Layout Window. When you click an element in the EDI file format, the corresponding layout component is highlighted in the Layout Window.
- list** A list displays a list of choices. Scroll bars are provided if there are more choices than can fit in the box.
- 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 *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).
- 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 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). A service, such as COMMERCE:Network, 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.
- Out Documents** This Gentran:Director 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.
- 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 Application Integration subsystem.
- Out Drawer** This Gentran:Director 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:Director function enables you to define, edit, and delete all partner information for your company and all of your trading partners.
- positional file** See application file.
- post** This Gentran:Director 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.
- print translation object** A print translation object organizes and formats the printout of EDI documents that are received from or sent to the trading partners for which you have established a trading relationship (inbound or outbound) that utilizes that print translation object. The print translation object enables you to view the EDI document in an easily readable format.
- 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:Director 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:Director function enables you to manually initiate a communications session to receive data from COMMERCE:Network through an invocation of the Connection Manager.
- 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:Director 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.
- screen entry translation object** A screen entry translation object provides a standardized format for keying an EDI document into the Document Editor, for translation and transmission to your trading partners, for which you have established an outbound trading relationship that utilizes that screen entry translation object. The screen entry translation object ensures that your users key all the data necessary to create the required EDI document.
- 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:Director 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:Director function enables you to manually start a communications session to send data to COMMERCE:Network through an invocation of Connection Manager. 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.
- simple mapping** See link.

- 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).
- static text** Static text is text that is always included in the screen entry or print translation object. Examples of static text include legends and column headings. Field labels, which usually accompany fields to identify the contents of the fields, are not considered static text.
- 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. 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 system import map.
- 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:Director system.

TDCC See EDIA.

TDF TDF is the Transaction Data File. This file serves as a filter between your document files and the Gentran:Director 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:Director 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 Network.

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 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.

Inbound Translation Objects:

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.

Export File: This translation object indicates that when a document is received, it is exported to a specified file format.

Print: This translation object is used to print documents.

Outbound Translation Objects:

Import: This translation object is used to import data from an application file.

Print: This translation object is used to print documents.

Data Entry: This translation object is used to enter data into the Document Editor facility.

- translator** This is the engine that processes data for the Gentran:Director, Application Integration subsystem and Forms Integration subsystem.
- transmission** See Communications Session.
- 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.



Please note that this function updates the internal Gentran:Director 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:Director 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.

Index

Symbols

- ?In Documents 277
- ?Out Documents 277

A

- accumulators 105
 - calculating a value total 137
 - calculating hash totals 133
 - calculating sums 126
 - using 126
- acknowledgements 277
- activation 30, 67, 277
- AIAG 277
- alphabetic language reference
 - atoi 185
 - atol 186
 - begin 186
 - break 187
 - concat 187
 - continue 188
 - count 188
 - delete 188
 - do 198
 - else 190
 - empty 189
 - end 189
 - exist 189
 - get 190
 - if 190
 - index 191
 - left 192
 - len 192
 - messagebox 193
 - mid 194
 - ntoa 194
 - right 195
 - select 195

- set 196
- strdate 196
- strstr 197
- then 190
- update 198
- while 198
- winexec 199

- ANA 277
- ANSI 277
- application file 277
- application system 278
- atoi 185
- atol 186

B

- begin 186
- break 187
- browser 278
- button 278

C

- check box 278
- click 278
- code list entry
 - deleting 150
- code list table 109
- code list tables 145, 278
 - copying and pasting 160
 - defining 146
 - deleting 150
 - exporting 155
 - importing 153
 - loading 157
 - loading code item descriptions 163
 - modifying 148
 - validating data against 162
- code table 109

- code tables
 - copying and pasting 160
 - defining 146
 - deleting 150
 - exporting 155
 - importing 153
 - loading code item descriptions 163
 - modifying 148
 - using 145
 - validating data against 162
- code value descriptions
 - storing 230
- colours 278
 - customizing 61
- column headings
 - adding 261
- communications session 278
- Communicator 278
- compile 278
- compliant 278
- component data element 279
- composite 6
 - customizing properties 88
- composite data element 279
- concat 187
- conditions 279
 - defining 85
 - modifying 85
- constant 279
 - using 210
- constants
 - defining 115
 - deleting 117
 - editing 116
 - using 98, 100, 114, 118
- continue 188
- control menu 9
- copy 33, 69
- count 188
- cross-reference table 279
- cross-reference tables
 - using information from 124
- customer service 4
- cut 33, 69

D

- data
 - formatting 45
- data element 280
- date
 - system 113
- date and time
 - formatting 226, 239
 - using the system 113
- date format
 - setting default 65
- date/time 279
- date/time type
 - using 57
- deactivation 279
- default 279
- delete 188
- dialog box 279
- display
 - customizing 60
- do 198
- document name
 - setting up 71
- documents 279, 288
 - turnaround 289
- double-click 280
- drop-down combo box
 - resizing 255

E

- EANA 280
- EDI 280
 - standard 280
- EDI file
 - activating 30
 - defining 20
 - loading a form 38
 - modifying composite properties 28
 - modifying element properties 29
 - modifying group properties 25
 - modifying segment properties 26
 - opening a form 38
 - promoting 32
 - splitting 32

- using 20
- using copy, cut, and paste 33
- EDI file format 5
- EDI File Format Window 280
 - customizing 64
- EDI file format window 11
- EDIA 280
- EDIFACT 280
- element 6, 280
 - component 279
 - composite 279
- else 190
- empty 189
- end 189
- entry fields
 - formatting 218
- envelope 281
- equalize 281
- exist 189
- export map 281
- export translation object 281
- extended rules 281

F

- field 282
 - arranging 249
 - creating help text 214
 - formatting display-only 220
 - hiding 221, 235
 - resizing 247
 - selecting and grouping 244
- field labels
 - customizing 206, 234
- fixed-format file 282
 - creating a group 23
- font 282
- fonts
 - customizing 63
- form 5, 282
 - compiling 42
 - creating 16
 - loading 38
 - naming 41

- opening 38
- saving 39
- formatting
 - data 45
- formatting data
 - date/time type 57
 - number type 54
 - string type 45
- forms
 - introducing 5
- frame 282
 - modifying titles 259
 - positioning 254
 - resizing 254
- functional group 282

G

- GENTRAN:Director
 - navigation 8
- get 190
- getting started 1
- global display options
 - customizing 60
- group 6, 282
 - creating 23
 - customizing properties 76
 - modifying 25
 - promoting 32
 - splitting 32

H

- hash totals
 - calculating 133

I

- if 190
- import 282
- import translation object 282
- In Documents 283
- In Drawer 283
- inbound mapping 283
- index 191
- initial value
 - setting 208

integer 283, 284
interchange 283

K

keyboard techniques 11

L

Layout Window 11, 283
 customizing 64
layout window 11, 283
left 192
len 192
list box 283
 adding column headings 261
 creating definitions 212
 creating help text 216
 modifying names 260
literal constant 283
literal constants
 defining 115
 deleting 117
 editing 116
 using 98, 114, 118
location table 283
location tables
 using information from 122
lookup table 283
lookup tables
 using information from 123
loop count 284
 using 107
loop end 284
 defining inbound 83
loop start 284

M

Main Menu Bar 10
Main Toolbar 10
map 284
 building 12
maximize 9
menu 284
Menu Bar 10, 284
message 288

messagebox 193
mid 194
minimize 9

N

network 284
ntoa 194
number 284
number type
 using 54
numbers
 formatting 223, 236

O

ODETTE 284
Out Documents 284
Out Drawer 284
outbound mapping 284

P

partner 285
partner definition
 using information from 121
Partner Editor 285
partner-related information
 using 121
paste 33, 69
positional file 285
posting 285
predefined 285
print translation object 285
 customizing field labels 234
 formatting dates and times 239
 formatting numbers 236
 formatting strings 242
 hiding fields 235
printout 285
promote 285

Q

qualifier 285
qualifiers
 generating 119

qualifying relationships
using 114

R

real number 285
real) 284
receiving 285
record 285
reference
 setting up 71
relational conditions 286
 defining 85
 modifying 85
responding 286
responsible agency 286
right 195

S

screen entry translation object
 creating help text for list boxes 216
screen entry translation object 286
 creating definitions for a list box 212
 creating help text for fields 214
 customizing field labels 206
 formatting dates and times 226
 formatting display-only fields 220
 formatting entry fields 218
 formatting numbers 223
 formatting strings 229
 hiding fields 221
 setting an initial value 208
 storing code value descriptions 230
 using a constant 210
segment 6
 customizing properties 77
 modifying 26
 promoting 32
 splitting 32
segments 286
select 195, 286
 using 90
sending 286
set 196, 286
simple mapping 286

split 287
standard rule
 loop count 107
standard rules 287
 constant 98
 select 90
 update 94
 use accum 105
 use code 109
 using 89
static text 287
 adding and positioning 258
status bar 11, 287
strdate 196
string 287
 formatting 229, 242
string type 45
strstr 197
sums
 calculating 126
support 4
syntax token 287
 creating for Eastern European languages 48
 creating for Western European languages 45
 deleting 52
 editing 50
 using 53
system date 113
system import map 287
system import translation object 287
system time 113
system translation object 287
system variables
 using 98

T

tab sequence
 setting 257
TDCC 288
TDF 288
technical support 4
then 190
third-party network 288

time

system 113

title bar 9, 288

Toolbar 10, 288

TRADACOMS 288

trading partner 288

transaction set 288

translation object

adding and positioning static text 258

adding column headings to list boxes 261

arranging fields 249

modifying frame titles 259

modifying list box names 260

resizing and positioning a frame 254

resizing drop-down combo boxes 255

resizing Fields 247

selecting and grouping fields 244

setting tab sequences 257

translation objects 288

translator 289

transmission 289

chain 289

turnaround document 289

turnaround map 289

type

date/time 57

number 54

string 45

U

UCS 289

update 198, 289

using 94

use accum 289

using 105

use code 290

using 109

user translation object 290

User's Guide

conventions 3

using 145

V

value total

calculating 137

version 290

VICS 290

W

while 198

winexec 199

Workspace 290

X

X12 290