GENTRAN:Director [®]

for Windows [®]

Version 5.3

Application Integration

User's Guide

Sterling Commerce
An IBM Company

June 2002

C O N T E N T S

Chapter 3   Customizing the Properties of Map Components 99

Chapter 4   Using Standard Rules. 119

Glossary

Index

1

Getting Started

GENTRAN:Director® for Windows® Application Integration subsystem provides you with the tools you need to translate your application files to EDI standard formats for documents you send to your partners (outbound mapping), and to translate EDI standard formats to your application format for documents that you receive from your partners (inbound mapping).

What's in this Manual

The majority of the GENTRAN:Director Application Integration documentation is contained in the online Help system. This manual is intended to support the online Help by assisting you in performing various tasks in the Application Integration subsystem. This task-oriented approach is intended to answer any questions you may have about GENTRAN: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 the GENTRAN:Director™ product.**

The *Application 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 to get technical support, provides a brief outline of how to build maps using the Application Integration subsystem, and gives you an introduction to mapping and GENTRAN:Director for Windows.

■ *Chapter 2, Mapping Basics* explains how to perform basic functions in the Application Integration subsystem, including creating a new map; using fixed-format and EDI files; opening, saving, and naming a map; compiling a map; and printing a mapping report. This chapter also explains customizing display options; setting the default date format; selecting which map components to use; cutting, copying, and pasting; setting up the document name and reference; formatting your data; and using field-to-field linking.

■ *Chapter 3, Customizing the Properties of Map Components* explains how to customize the properties of record, segments, looping structure, fields, elements, 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 Application Integration 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, Setting Up the Import Process* explains all the facets of the import process, defines import file restrictions, and describes how to create a system import map.

■ *Chapter 6, Setting Up the Export Process* explains all the facets of the export process and describes how to use envelope information.

■ *Chapter 7, Using Extended Rules* defines what extended rules are and when they are used, and explains the proprietary programming language syntax.

■ *Chapter 8, Error Messages* describes GENTRAN:Director error messages and specifies corrective actions.

■ The *Glossary* contains definitions of mapping 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:

Creating a new map
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., **CTRL** + **C**).

Click this button
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 Mapping

A map is a set of instructions you define in the GENTRAN:Director Application Integration subsystem that indicates the corresponding relationship between your application file and the EDI standards, and defines how the system should translate data.
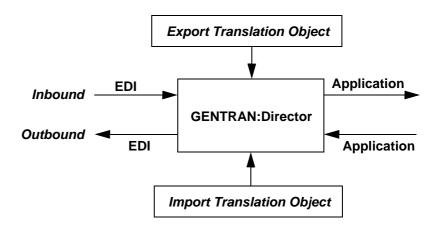
For inbound mapping, the system translates your trading partner's EDI standard formatted business document to your application file format. To translate inbound data, you need to create an export map in GENTRAN:Director Application Integration (because the system is *exporting to* your application file).

For outbound mapping, the system translates your application file format to EDI standard formats, so you can send documents to your partners. To translate outbound data, you need to create an import map and a system import map in the GENTRAN:Director Application Integration subsystem. You need to create an import map because the system is *importing from* your application file. The import map defines how to move data from your application file (flat file definition), which may contain multiple documents, to the EDI standard-formatted documents that your partners expect to receive from you. You need to create a system import map to determine which trading relationship (established in Partner Editor) corresponds to each document in the application file, so the system knows which import map to use to process the document. Please see *Creating a System Import Map* on page 217 for more information on creating a system import map.

After you create a map in the Application Integration subsystem, this map is the *source* from which you compile a translation object. A translation object is a set of rules that instruct the translator how to convert a file from one format to another. Every function performed during translation requires a translation object. Translation objects are used to control all the functions of GENTRAN:Director EDI translation. Translation objects can belong to the following two categories: system translation objects or user translation objects. System translation objects control the creation and separation of interchanges, functional groups, and transaction sets. They are also used to generate and reconcile functional acknowledgements. User translation objects control data entry, importing, exporting, document turnaround, and creating printed reports. All the required system translation objects are automatically installed with the GENTRAN:Director system.

The GENTRAN:Director Application Integration subsystem enables you to generate import, export, and document turnaround translation objects. The GENTRAN:Director Forms Integration subsystem enables you to generate data entry and print translation objects. Once you generate a translation object using the GENTRAN:Director Application Integration subsystem (or the GENTRAN:Director Forms Integration subsystem), you must register the translation object with GENTRAN:Director (see the GENTRAN:Director *User's Guide* for more information). In addition, you must establish correct trading relationships for all trading partners from which you receive or to which you send data (see the GENTRAN:Director *User's Guide* for more information). During the process of establishing a trading relationship, you need to specify the name of the appropriate translation object that the translator uses to translate the data (see the GENTRAN:Director *User's Guide* for more information).

The diagram below illustrates the inbound and outbound translation process:

When you create an import map, you need to define to GENTRAN:Director your application file format (your data that you sends to your partners) and the EDI standard format in which your partners expect to receive the documents. When you create an export map, you need to define to GENTRAN:Director the EDI standard format in which your partners sends documents to you and your application file format (how the data needs to be formatted for your application to process it).

Application File Format

If you are creating an import or export map, you must define your application to the GENTRAN:Director 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.

The map components that you use to define your application file are as follows:

■   A **group** is a looping structure that contains related records and/or groups that repeat in sequence until either the group data ends or the maximum number of times that the loop is allowed to repeat is exhausted. 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 is a group and therefore, it is visually represented the same way as other groups and sub-groups in the Application Integration subsystem:

■   A **record** contains a group of related fields. A record can occur once or can repeat multiple times. A record is visually represented in the Application Integration subsystem as the following:

■   A **field** is the smallest piece of information defined in the application file. A field is the application map component that is mapped (linked) to a corresponding EDI element to move data to and from your application file. A field is visually represented in the Application Integration subsystem as the following:

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

Before you define your application, you should obtain a layout of the necessary records, fields, and groups. Each map component is arranged sequentially in the order that it is most logical for the system to process. Therefore, each level of your application must be created sequentially. For example, your application file contains records and groups. The records contain fields and the groups contain records and/or sub-groups. This means that you must create records and groups before you create the subordinate fields. Please see *Using a Fixed-Format File* on page 23 for more information on defining your application file format.

EDI File Format

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

The GENTRAN:Director Application Integration subsystem generates an EDI file for you, based on the standard (agency), version, transaction set, and release (for the TRADACOMS standard *only*) you selected. The system includes all the groups, segments, composites, and elements that are defined by the standards agency for the version of the document you selected. Please see *Creating a New Map* on page 18 for more information on generating an EDI file when you create a new map. If you are creating an import or export map, you typically need to customize the system-generated EDI file by modifying the properties of the map components, and using specialized GENTRAN: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 map components that you use depend on the type of map you are creating. This includes the standard, version, and transaction set (document) selected, and which groups, segments, composites, and elements your company requires. We recommend that you determine which map components you are using before generating or defining an EDI file.

The map 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 the Application Integration subsystem:

■ 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 the Application Integration subsystem 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 the Application Integration subsystem as the following:

■ An **element** is the smallest piece of information defined by the EDI standards. An individual element can have different meanings depending on the context. Therefore, elements are normally not considered to have useful meaning until they are combined into segments. 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. An element is visually represented in the Application Integration subsystem as the following:

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

Please see *Using an EDI File* on page 35 for more information on customizing the properties of the map components, and using specialized GENTRAN:Director functions to manipulate the EDI file structure.

Introducing the Application Integration Subsystem

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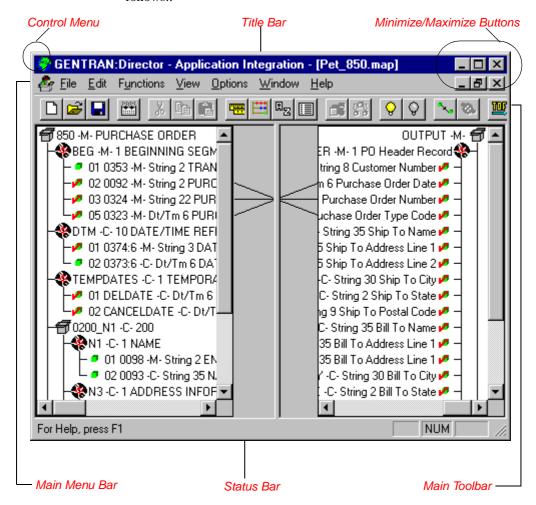
Navigation

You can access the GENTRAN:Director Application Integration Window in one of four different ways:

■   Click the menu option from the Main Menu Bar.

■   Click the appropriate button on the Main Toolbar.

■   Click the appropriate part of the map.

■   Click a map component with the right mouse button to access a shortcut menu, which contains all the functions that are currently available for that map component. The content of the shortcut menus varies, depending on the type and level of the selected map component.
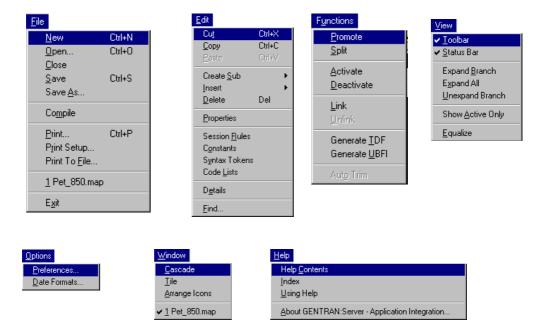
When you start GENTRAN:Director Application Integration, the Main Menu Bar contains a subset of menu items. A full set of menu items is displayed after you create a new map or open (load) an existing map.

The GENTRAN:Director - Application Integration Window appears as follows::

Control Menu   *Title Bar*   *Minimize/Maximize Buttons*

Main Menu Bar   *Status Bar*   *Main Toolbar*
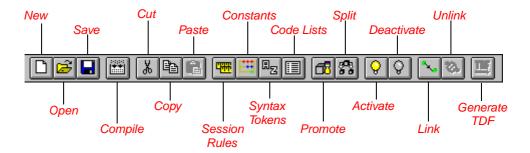
- ■ The **Control Menu** contains the following commands: **Restore**, **Move**, **Size**, **Minimize**, **Maximize**, and **Close**.

- ■ The **Title Bar** contains the name of the application. You can use this to move the position of the EDI Director window on your desktop.

- ■ The **Minimize** button reduces the window to an icon.

  The **Maximize** button 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.

File	
New	Ctrl+N
Open...	Ctrl+O
Close	
Save	Ctrl+S
Save As...	
Compile	
Print...	Ctrl+P
Print Setup...	
Print To File...	
1 Pet_850.map	
Exit	

Edit	
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Create Sub	▶
Insert	▶
Delete	Del
Properties	
Session Rules	
Constants	
Syntax Tokens	
Code Lists	
Details	
Find...	

Functions
Promote
Split
Activate
Deactivate
Link
Unlink
Generate TDF
Generate UBFI
Auto Trim

View	
✔ Toolbar	
✔ Status Bar	
Expand Branch	
Expand All	
Unexpand Branch	
Show Active Only	
Equalize	

Options
Preferences...
Date Formats...

Window
Cascade
Tile
Arrange Icons
✔ 1 Pet_850.map

Help
Help Contents
Index
Using Help
About GENTRAN:Server - Application Integration...

■   The **Main Toolbar** gives you access to some of the most common
    operations in the Application Integration subsystem. 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 **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).

Keyboard Techniques

The procedures in this manual are designed to be executed with a mouse. If you prefer to use the keyboard, 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 me   that option, you may press that key combination as well.**
>
> **Instead of clicking a button to execute a command on a dialog or browser, press TAB to highlight the desired button, and press ENTER to execute that command.**

Building a Map with Application Integration

This section defines how to proceed with building a map using the
GENTRAN:Director Application Integration subsystem.

1. **Prepare and Analyze**
   Obtain a layout of your application file and determine how it
   corresponds with the EDI standard you are using. Determine how you
   move data to or from each application field.

2. **Set Global Defaults (first time only)**
   The first time you use the GENTRAN:Director Application Integration
   subsystem, you should establish the default date format that the system
   uses. Please see *Setting Default Date Format* on page 93 for more
   information on setting the global date default.

3. **Create, Save, and Name a New Map**
   Please see *Creating a New Map* on page 18 for more information on
   creating, saving, and naming a map.

4. **Activate the Appropriate EDI Groups, Segments, and Elements or
   Auto Trim Based on a Sample EDI File**
   Please see *Selecting Which Map Components to Use* on page 94 for
   more information on activation. Please see *Using Auto Trim* on page 46
   for more information on auto trim.

5. **Define your Application**
   Please see *Using a Fixed-Format File* on page 23 for more information
   on defining your application.

6. **Map the Appropriate Data for Each Application Field**
   Please see *Using an EDI File* on page 35 for more information on
   customizing your EDI file. Please see *Creating Simple Field-to-Field
   Links* on page 84 for more information on simple mapping (linking).
   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.

7. **Compile the Translation Object**
    Please see *Compiling a Map* on page 62 for more information on
    compiling the translation object and translation object naming
    conventions.

8. **Print the Mapping Report**
    Validate and review the map, and make modifications as needed. Please
    see *Printing a Mapping Report* on page 65 for more information on
    printing the mapping report.

9. **Register the Translation Object with GENTRAN:Director**
    Please see the GENTRAN:Director *User's Guide* for more information
    on registering translation objects.

10. **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.

11. **Testing the Translation Object**
    Obtain test data from your partners and process the data. Verify
    acknowledgement processing (if applicable). Verify communications
    with your network.

2

Mapping Basics

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

This chapter includes the following topics:

- Creating a New Map
- Using a Fixed-Format File
- Using an EDI File
- Opening a New Map
- Saving a Map
- Naming a Map
- Compiling a Map
- Printing a Mapping Report
- Setting the Default Date Format
- Selecting Which Map Components to Use
- Setting Up the Document Name and Reference
- Formatting Your Data
- Creating Simple Field-to-Field Links
- Customizing Your Display
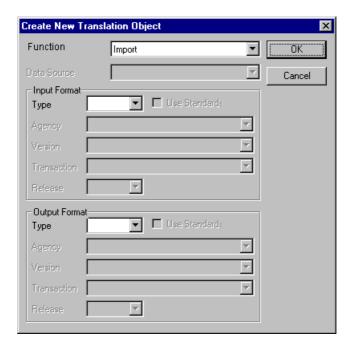
Creating a New Map

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

Complete the following steps to create a new map:

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

2. Complete the Input Format section of the dialog. This is the format of the data that is translated by the GENTRAN:Director system.

3.   Select the type of format for the input data from the Type list. Valid values are "EDI" or "Positional."

■   For **Inbound** maps (Export), the Input Format Type is "EDI."

■   For **Outbound** maps (System Import or Import), the Input Format Type is "Positional."

■   For **Turnaround** maps, the Input Format Type is "EDI."

4.   If you selected "EDI" in the Type box and if you want to use the EDI standard, click the Use Standards check box. If you select "Positional" in the Type box, proceed to step 9.

5.   If you want to select a data source name from which Application Integration accesses EDI standards, select one from the Data Source list. This DSN is prompted the next time you create a map.

6.   Select the EDI standard you are using for the input side of the map from the Agency list. Valid values are:

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

7.   Select the version of the standard from the Version list. The versions that are available depend on which standard you selected.

8.   Select the transaction set (document or message) from the Transaction list. The transactions that are available depend on which standard and version you selected.

9.   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.

10. Select the type of format for the output data from the Type list. Valid values are "EDI" or "Positional."

■   For **Inbound** maps (Export), the Output Format Type is "Positional."

■   For **Outbound** maps (System Import or Import), the Output Format Type is "EDI."

■   For **Turnaround** maps, the Output Format Type is "EDI."

11. If you selected "EDI" in the Type box, click the Use Standards check box. If you selected "Positional" in the Type box, proceed to step 15.

12. Select the EDI standard you are using for the input side of the map from the Agency list. Valid values are:
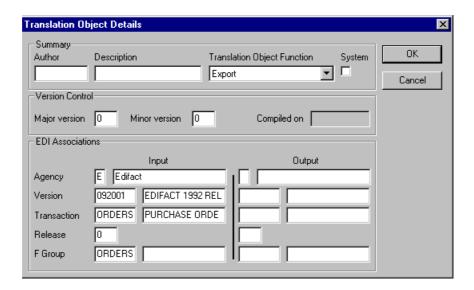
[A] = TRADACOMS
[E] = EDIFACT
[O] = ODETTE
[T] = TDCC
[U] = UCS/WINS
[X] = ANSI X12

13. Select the version of the standard from the Version list. The versions that are available depend on which standard you selected.

14. Select the transaction set (document or message) from the Transaction list. The transactions that are available depend on which standard and version you selected.

15. 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.

16. Click OK to generate a new map (this may take a few seconds). The new map is displayed in the GENTRAN:Director Application Integration Window.

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

18. Complete the Author box. This box typically contains the name of the person who created the map.

19. Complete the Description box. This box must be unique because it is used by the system to identify the map. We recommend that you use the following identifying characteristics: the partner that this map is used for, the standard, the version, the type of transaction this map uses, and the direction of the map (e.g., Import, Export). For example, "MWT X 3030 850 Export" is the description of a map used with partner MWT, for an ANSI X.12 version 003030 Purchase Order (850), that is sent Inbound (Export).

20 Verify the type of translation object in the Translation Object Function list.

21. 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).
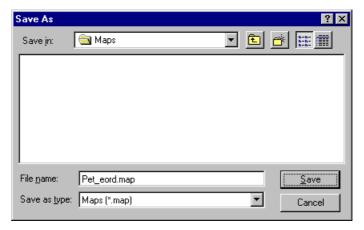
STOP **You should only select this check box if you are certain you want to define this translation object as a system translation object. Once a translation object is registered with GENTRAN:Director, it cannot be deleted.**

22. 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.

23. The Input and Output boxes (Agency, Version, Transaction, Release, and F Group) contains information only if the Input and/or Output sides of the map are of EDI format. You can change the information in these boxes, but it does not alter the content of the map. You may wish to alter these boxes if, for example, you want the map to reflect a standards version that is not loaded on your system. You can change the version on this dialog, and then alter the map to be compliant with that version.

24. Click OK to save the translation object details and access the Save As dialog.

25. We recommend that you store maps in the Maps folder where GENTRAN:Director is installed.

26. Type the name of the map (one to eight characters) in the "File name" box. The default file extension is .MAP.

27. Click SAVE to save the map.

After you finish creating and saving a new map, you need to define the Input and Output sides of the map. The steps you take are different, depending on whether the map is an Import, System Import, Export, or Turnaround map.

Using a Fixed-Format File

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

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

For example, your application contains records and groups. The records contain fields and the groups contain records, sub-groups, or both. This means that you must create records and groups before you create the subordinate fields.

Loading a File Definition

GENTRAN:Director enables you to load an individual file format definition that you previously saved. This feature provides you with a quick way to build the fixed-format or EDI sides of your map. Please see *Saving a File Definition* on page 24 for more information on saving a file definition.
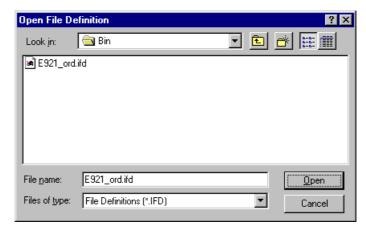
STOP **Loading a file definition replaces the selected side of the map. Please be certain that is your intent before performing this task.**

Complete the following steps to load a 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 Open File Definition... from the shortcut menu.

If you already used GENTRAN:Director to create a fixed-format file on that side of the map, you are prompted with a message that warns you that the existing file format will be replaced. Click YES to continue.

The Open File Definition dialog is displayed.

3.   The "Look in" list should reflect the drive\folder where
      GENTRAN:Director is installed. The default is c:\Program
      Files\Sterling Commerce\GENTRAN Director. If the current subfolder
      is not the correct one, select the appropriate folder.

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

5.   Select the file definition that you want to load into the system from the
      list or type it in the "File name" box.

6.   Double-click the filename or click OPEN to load the selected file format
      definition.

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 maps. This provides you with a
quick way to build the fixed-format or EDI sides of your map. Please see
Loading a File Definition on page 23 for more information on loading a file
definition.

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.

Save File Definition		? X
Save in:	⬛ Bin	▼ 🔼 📂 📰 📰

File name:	E921_ORD.IFD	Save
Save as type:	File Definitions (*.IFD)	▼	Cancel

3.   The "Save in" list should reflect the drive\folder where
      GENTRAN:Director is installed. The default is c:\Program
      Files\Sterling Commerce\GENTRAN Director. If the current subfolder
      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 the First Record

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

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

Complete the following steps to create the first application record:

1.   Click the Positional File icon to highlight it (OUTPUT for an inbound
      map and INPUT for an outbound map).

2.   Select Create Sub from the Edit menu. Select Record from the submenu.

The Positional Record Properties dialog is displayed.

Positional Record Properties

Identification

Name _____

Desc _____

Tag _____ Position [1] ☐ Use Key Field [Key Field...]

Looping

Min Usage [0]   ⊙ Normal
                   ○ Loop Start
Max Usage [1]   ○ Loop End

Special Properties

⊙ Normal

○ Floating

○ Wildcard   Alternate Tag [____]   Position [1]

[OK]
[Cancel]
[Promote]
[Split...]
[Rules...]
[Edit Fields...]

3.   In the Name box, type the record name.

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

4.   In the Desc box, type a description of the record. This box is used to provide a brief explanation of the record that allows you to differentiate it from similar records.

5.   In the Tag box, type the record identification code <TAG>. For example, a record is recognized by the system as:

     <TAG>[Field_1][Field_2]

     The record TAG for each record enables the system to recognize that record and then determine the mapping requirements.

> We recommend that you define the record tag on the
> **Positional Record Properties dialog, instead of defining**
> **fields with the purpose of explicitly containing the**
> **record tag.**

6. In the Position box, type the starting column position of the tag in the
   data record.

The Looping section of the Positional Record Properties dialog enables you
to specify the minimum and maximum number of times this record can
repeat, and whether a record is mandatory or conditional. Alter this section
only if the record is a looping structure. Otherwise, accept the defaults.

7. In the Min Usage box, type the minimum amount of times the record
   must repeat.

   We recommend that you type "1" to make the header record mandatory.

> **If the Min Usage box contains a "0" (zero), the record**
> **is "conditional." If the Min Usage box contains a "1" or**
> **greater, the record is "mandatory."**

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

   For the header record, we recommend that you accept the default of "1"
   so it remains a record that only occurs once (not a looping structure).

9. Click OK to create the record.
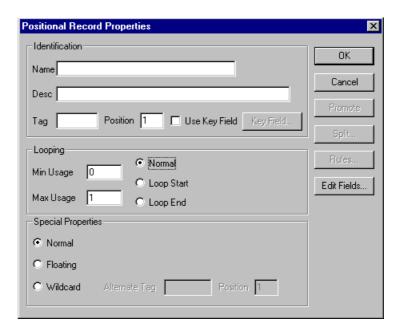
Creating Subsequent Records

After you create the first record, you can define subsequent records at the same level. You can also define groups, if your application includes looping structures. You may need to define a loop if your application requires a group of related records (and/or sub-groups) that repeat in sequence. Please see *Creating a Group* on page 30 for more information on defining groups.

Complete the following steps to create additional application records:

1. Highlight the record which precedes the record you are creating in the file layout.

2. Select Insert from the Edit menu. Select Record from the submenu.

    The Positional Record Properties dialog is displayed.

3. In the Name box, type the record name.

4. In the Desc box, type a description of the record. This box is used to provide a brief explanation of the record that allows you to differentiate it from similar records.

5. In the Tag box, type the record identification code <TAG>. For example, a record is recognized by the system as:

    <TAG>[Field_1][Field_2]

    The record TAG for each record enables the system to recognize that record and then determine the mapping requirements.

6. In the Position box, type the starting column position of the tag in the data record.

7. In the Min Usage box, type the minimum amount of times the record must repeat.

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

9. Click OK to create the record. Use this procedure to create any additional records.

Creating a Group

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

Complete the following steps to create a group:

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

    OR

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

The Group Properties dialog is displayed.

2. In the Name box, type the group name.

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

3. In the Description box, type the loop description. This box is used to provide a brief explanation of the loop.

4. In the Minimum 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 Maximum Usage box, type the maximum number of times the loop can be repeated.

6. Click OK to create the group.

Creating Fields

Each record you create contains a group of logically-related application fields. These fields define the structure and content of the data to the system.
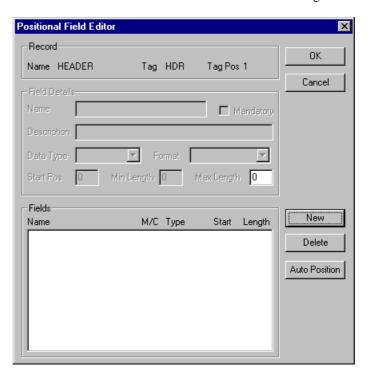
The easiest way to add application fields to a record is to use the Positional Field Editor. Generally, you create the fields for the first record in the application file, and then proceed with each sequential record.

> **We recommend that you do not define fields with the purpose of using the field to explicitly contain the record tag. This is because the system takes the tag that you define in the record into account when the automatic sequencing (Auto Position) feature is used. We recommend that you define the record tag on the Positional Record Properties dialog. Please see page 26 for more information.**

Complete the following steps to create the application fields for a record:

1. Right-click the application record and select Edit Fields... from the shortcut menu to access the Positional Field Editor dialog.

2. If the field you are creating is the first field in the record, click NEW. Highlight the field that precedes the field you are creating in the record layout and click NEW. A highlight bar is displayed in the Fields section where the new field is positioned. You need to fill in the field values in the Field Details section.

3. In the Name box, type the field name. Each application field must have a unique name. It is useful to tag the end of the fields that occur in multiple records with a suffix that identifies the record that contains it. For example, "HDR" for the Header record.

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

4. If you want to designate the field as "mandatory" (must be present), click the Mandatory check box.

5. In the Description box, type a description of the field. The description is used to provide a brief explanation of the field that allows you to differentiate it from similar fields.

6. From the Data Type list, select the type of the field. Valid values are:

   String = alpha element
   Number = numeric or real element
   Dt/Tm = date or time element

7. From the Format list, select how the data element is formatted. The choices for this field depend on the type of field you selected from the Data Type list. If you choose Number or Dt/Tm in the Data Type box, you can select the data format from the Format list. If you selected String from the Data Type box, you should type a syntax token to denote that this field must be formatted as the specified syntax token dictates (the default syntax token is "X"). Please see *Using Syntax Tokens* on page 77 for more information on defining and using syntax tokens.

8. If you want to indicate the exact position of the field in the record, type the starting position of the field in the Start Pos box. You want to specify field start positions if, for example, you are only using a few fields but you want them positioned exactly in the record.

    The alternative to specifying the start position of each field, is to add the fields sequentially in the record and then use the Auto Position function. Please see step 13 below for more information on the Auto Position function.

9. In the Min Length box, type the minimum number of characters that is acceptable for this field.

10. In the Max Length box, type the maximum length of the field.

11. To add the field and to create another field positioned after it, click NEW. A new field with blank values ready for you to identify is positioned below the field.

12. Create the rest of the fields according to your record layout.

13. After adding the last field, if you want the system to automatically position the fields in the record, click AUTO POSITION. This automatically calculates the start position in the record of each field based on the criteria that each field is positioned directly after the previous field and is of the length specified in the Max Length box. Click YES to acknowledge the warning message that fields are sequenced in order.

> **STOP** **It is only valid to use the Auto Position function if you define a record tag, and if you define every field in the record in the sequence that each field occurs. Please see page 26 for more information on defining record tags.**

14. When you are completely finished adding fields to the record, click CLOSE. Complete this procedure to add fields to the other records you defined.

Using a Constant with a Field

For inbound maps, you may want to use a literal constant to move a value into an application field that has no corresponding EDI data (you cannot link the field directly to an EDI element). Please see *Defining Literal Constants* on page 154 for more information on defining constants. Please see *Mapping Literal Constants* on page 159 for more information on mapping a constant to a field.

Using an EDI File

GENTRAN:Director generates an EDI file for you, based on the standard (agency), version, transaction set, and release you select. The system includes all the groups, segments, composites, and elements that are defined by the standards agency for the version of the document you select. Please see *Creating a New Map* on page 18 for more information on generating an EDI file when you create a new map. You can modify the system-generated EDI file by modifying the properties of the map components, and using the promote, split, copy, cut, and paste functions.

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

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

Verifying EDI Delimiters
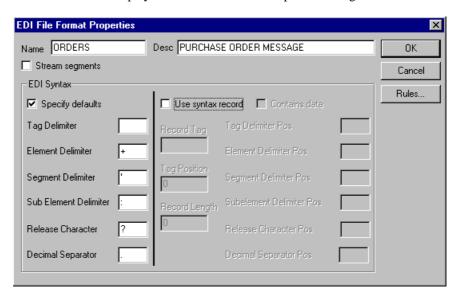
If you are using an EDI standard that contains composite elements or sub-elements, you must verify that GENTRAN:Director is specifying the correct delimiters. Delimiters are flags that you define to the system as separating specific EDI components. Delimiters are necessary for all variable field-length 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.

STOP **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 you perform this task regardless of which standard you use.**

Complete the following steps to verify EDI delimiters:

1. Right-click the EDI file icon and select Properties... from the shortcut menu to display the EDI File Format Properties dialog.

2. In the EDI Syntax section, verify that the Specify defaults check box is selected.

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 EDI File Format Properties dialog.

Loading a File Definition

GENTRAN:Director enables you to load a previously-saved individual file format definition. This provides you with a quick way to build the EDI or fixed-format sides of your map. Please see *Saving a File Definition* on page 38 for more information on saving a file definition.

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

Complete the following steps to load a file format definition:
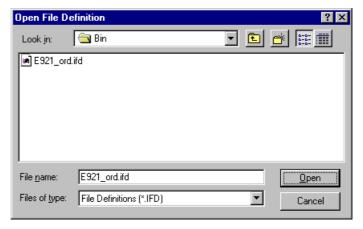
1. Right-click the EDI icon (either the INPUT or OUTPUT side of the map) and select Open File Definition... from the shortcut menu.

> **i** **If you already used GENTRAN:Director to generate an EDI file or if you created an EDI file on that side of the map, you are prompted with a message that warns you that the existing file format is replaced. Click** YES **to continue.**

The Open File Definition dialog is displayed.

2. The "Look in" list should reflect the drive\folder where GENTRAN:Director is installed. The default is GENTRAN Director\BIN. 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 .IFD.

4. Select the file definition that you want to load into the system from the list or type the name in the "File name" box.

5. Double-click the filename or click OPEN to load the selected file format definition.

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 maps. This provides you with a quick way to build the EDI or fixed-format sides of your map. Please see *Loading a File Definition* on page 37 for more information on loading a file definition.

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

1. Right-click the EDI icon (either the INPUT or OUTPUT side of the map) and select Save File Definition As... from the shortcut menu. The Save File Definition dialog is displayed.

Save File Definition			? X
Save in:	Bin		

```
File name:     E921_ORD.IFD                         Save
Save as type:  File Definitions (*.IFD)             Cancel
```
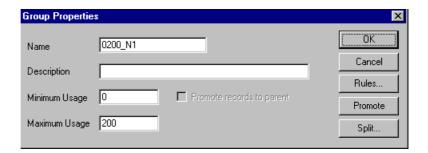
2. The "Save in" list should reflect the drive\folder where GENTRAN:Director is installed. The default is GENTRAN Director. If the current folder is not the correct one, select the appropriate folder.

3. Select the name of 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.

4. Click SAVE to save the file definition.

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. Right-click the group you want to modify and select Properties... from the shortcut menu The Group Properties dialog is displayed.

```
┌─────────────────────────────────────────────────────────────┐
│ Group Properties                                         [X] │
│                                                              │
│                                            ┌──────────────┐  │
│  Name          [0200_N1            ]       │     OK       │  │
│                                            └──────────────┘  │
│  Description   [                        ]   ┌────────────┐   │
│                                             │   Cancel   │   │
│  Minimum Usage [0    ]  □ Promote records   └────────────┘   │
│                           to parent         ┌────────────┐   │
│                                             │   Rules... │   │
│  Maximum Usage [200  ]                      └────────────┘   │
│                                             ┌────────────┐   │
│                                             │  Promote   │   │
│                                             └────────────┘   │
│                                             ┌────────────┐   │
│                                             │   Split... │   │
│                                             └────────────┘   │
└─────────────────────────────────────────────────────────────┘
```

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

    If a segment/record or element/field occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where "n" is the number of the occurrence in the map.

3. If you want to modify the group description, type the new description in the Description box. This box is used to provide a brief explanation of the loop.

4. In the Minimum 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 Maximum 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 105 for more information on using loop start and loop end segments.

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

1.   Right-click the segment you want to modify and select Properties... from the shortcut menu. The EDI Segment Properties dialog is displayed.

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

> **If a segment/record or element/field occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where "n" is the number of the occurrence in the map.**

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. If you want to modify the segment tag, type the segment identification code <TAG> in the Tag box. For example, a record is recognized by the system as:

    <TAG>[Delimeter]<DATA>[Delimeter]<DATA>.......<DATA> [Segment Terminator]

    The segment TAG for each segment enables the system to recognize that segment.

5. 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."**

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

7. 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

8. 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.

i **Composite elements are only used for EDI maps.**

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

1. Right-click the composite you want to modify and select Properties... from the shortcut menu. The Composite Properties dialog is displayed.

Composite Properties			✕
Name	C002	☐ Mandatory	OK
Description			Cancel

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

i **If a segment/record, composite, or element/field occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where "n" is the number of the occurrence in the map.**
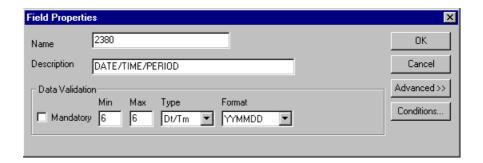
3. Click the Mandatory check box to indicate whether this composite must be present.

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.

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/record or element/field occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where "n" is the number of the occurrence in the map.**

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 Format list, select how the data element ID is formatted. The choices for this field depends 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 77 for more information on defining and using syntax tokens.

9. Click OK to save the changes to the element.

Activating EDI Map Components

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

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

> **As an alternative to activation, you can use the Auto Trim feature if you have a sample EDI file defined. Please see *Using Auto Trim* on page 46 for more information on the Auto Trim feature.**

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

1.   Select Activation from the Functions menu or the Main Toolbar.

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

2.   Open (double-click) each group that contains segments or groups that need to be activated.

3.   Click each inactive group that you need to use. This activates the groups.

> **If you accidentally click a group, segment, composite, or element that you did not mean to activate, right-click the map component and select Deactivate from the shortcut menu.**

4.   Click each inactive segment that you need to use. This activates the segments.

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. This activates the composites.

7.   Open (double-click) each composite that contains elements that need to be activated.

8.   Click each inactive element that you need to use. This activates the elements.

9.   Once you have activated all the necessary groups, segments, and elements, click the Activation button on the Main Toolbar to turn activation mode off.

Using Auto Trim
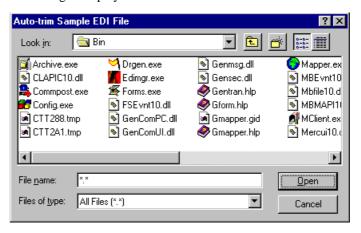
Instead of activating map components manually, you can use the Auto Trim feature to modify the EDI side of the map according to a sample EDI file that you select (you must have previously created this EDI file). This feature examines the EDI file that you specify and then activates and deactivates map components on the EDI side of the map, so that the two files match. The sample EDI file must be the same standard, version, and transaction set (message) as the map for Auto Trim to match map components. To use Auto Trim, you must define the Segment and Element Delimiters. If you do not supply the Tag Delimiter, the system substitutes the Segment Delimiter. If you want to use composite elements, you must specify the Sub Element Delimiter. If you want to use release characters, you must specify the Release Character. Please see *Verifying EDI Delimiters* on page 35 for more information on specifying EDI delimiters.

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

Complete the following steps to use Auto Trim:

1. Select a map component on the EDI side of the map. The EDI side of the map must be selected for the Auto Trim function to be available.

2. Select Auto Trim from the Functions menu. The Auto-trim Sample EDI File dialog is displayed.

3. The "Look in" list should reflect the drive\folder where the EDI file is stored. The default is GENTRAN Director\BIN. If the current subfolder is not the correct one, select the appropriate subfolder.

4. If necessary, select the type of file that you want to display from the Files of Type list.

5. Select the sample EDI file that you want to load into the system from the list.

6. Double-click the filename or click OPEN to begin the auto trim process. When auto trim is complete, the system displays a message stating whether or not auto trim was successful. Click OK to acknowledge the message.

Promoting Groups and Repeating Segments

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

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 50 for more information on splitting groups and repeating segment. Please see *Using Copy, Cut, and Paste* on page 54 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.

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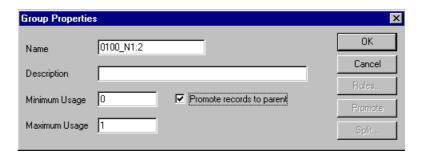
In this example, your application file contains a Ship To record that groups all the shipping information and a Bill To record that groups all the billing information. Your Ship To and Bill To records both occur only once in the application. The EDI file contains a N1 group (for ANSI) or a NAD group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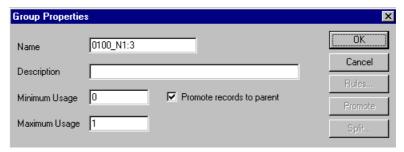
Complete the following steps to map unique data to your application file:

1. Activate the N1/NAD loop and the appropriate segments and elements, if you have not already done so. Please see *Activating EDI Map Components* on page 44 for more information on activating. If you already activated the N1/NAD group, just highlight it.

2. Click the PROMOTE button on the Main Toolbar to extract a single instance of the N1/NAD group.

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

Group Properties			
Name	0100_N1:2		OK
Description			Cancel
			Rules...
Minimum Usage	0	☑ Promote records to parent	Promote
Maximum Usage	1		Split...

4. In the Description box, type a specialized definition to identify this group as containing billing information.

5. Click OK to save the change and exit the Group Properties dialog.

6. Highlight the original N1/NAD group (that now repeats 199 times) again.

7. Click the PROMOTE button on the Main Toolbar to extract another single instance of the N1/NAD.

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

Group Properties			
Name	0100_N1:3		OK
Description			Cancel
			Rules...
Minimum Usage	0	☑ Promote records to parent	Promote
Maximum Usage	1		Split...

9. In the Description box, type a specialized definition to identify this group as containing shipping information.

10. Click OK to save the change and exit the Group Properties dialog.

11. Click the DEACTIVATE button on the Main Toolbar and then click the original N1/NAD group (that now repeats 198 times), so the system does not process it.

12. Now you can link the elements in the single occurrence of the N1/NAD that contains billing information to the fields in your Bill To application record. And, you can link the elements in the single occurrence of the N1/NAD that contains shipping information to the fields in your Ship To application record.

Splitting Groups and Repeating Segments

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

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.

> **STOP**   **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.

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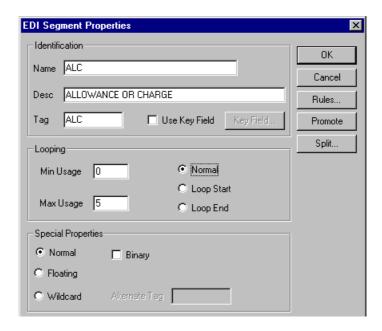
Complete the following steps to split the ALC segment:

1. Activate the ALC segment and the appropriate elements, if you have not already done so. Please see *Activating EDI Map Components* on page 44 for more information on activating. If you already activated the ALC segment, just highlight it.

2. Click the SPLIT button on the Main Toolbar. The Split dialog is displayed.
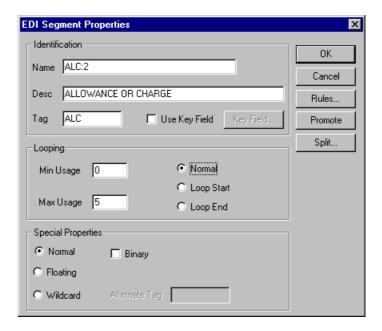
3. In the First Loop Entries box, type "5." This is the number of sequential iterations where you want the repeating segment split.

4. Click OK to exit the Split dialog.

5.   Select Properties from the Edit menu to access the EDI Segment Properties dialog.

6.   In the Desc box, type a specialized definition to identify this segment as containing allowance information.

7.   Click OK to save the change and exit the EDI Segment Properties dialog.

8.   Highlight the original ALC segment (that now repeats 5 times).

9. Select Properties from the Edit menu to access the EDI Segment Properties dialog.

10. In the Desc box, type a specialized definition to identify this segment as containing charge information.

11. Click OK to save the change and exit the EDI Segment Properties dialog.

12. Now you can link the elements in the ALC segment that contains allowance information to the fields in your Allowance application record. And, you can link the elements in the ALC segment that contains charge information to the fields in your Charge application record.

Using Copy, Cut, and Paste

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

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

You can also cut, copy, and paste a map component from one map to another.
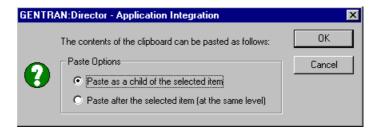
Complete the following steps to cut, copy, and paste a map component:

1. Select the map component that you want to cut or copy.

2. Click the Copy button on the Main Toolbar.

3. If you are pasting the map component in another map, open that map, if it is not already open. Please see *Opening a Map* on page 57 for more information on opening a map.

4. Select the map component that you want the cut or copied selection pasted *after*.

    If you cut or copied an EDI component, you can only paste that component into the EDI side of a map. Conversely, if you cut or copied a positional component, you can only paste that component into the positional side of a map.

5. Select Paste from the Edit menu or the Main Toolbar to paste the contents of the Clipboard. If the map 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.
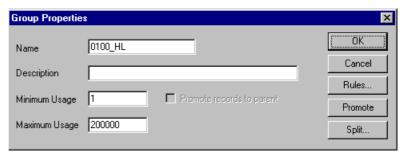
Example   In this example, you are defining an ANSI X12 Ship Notice/Manifest (856). When the system generates the 856 document, it contains one HL group. The HL group identifies dependencies between and the content of hierarchically-related groups of segments.

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 Properties from the Edit menu to access the Group Properties dialog.
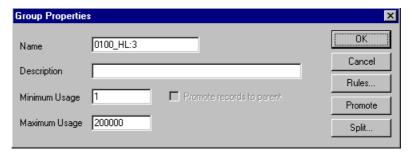
3.   In the Description box, type a specialized definition to identify this group as containing shipment information.

4.   Click OK to save the change and exit the Group Properties dialog.

5.   Select the map component in the HL that you want the nested HL group (order information) pasted *after*.

6.   Click the Paste button on the Main Toolbar to paste the contents of the Clipboard. The second HL group is added as a nested loop to the original HL loop.

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

8. In the Description box, type a specialized definition to identify this group as containing order information.

9. Click OK to save the change and exit the Group Properties dialog.

10. Double-click the second HL group (order information) to open it.

11. Select the map component in the HL that you want the nested HL group (item information) pasted *after*.

12. Click the Paste button on the Main Toolbar to paste the contents of the Clipboard. The third HL group (item information) is added as a nested loop of the second HL group.

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

14. In the Description box, type a specialized definition to identify this group as containing line item information.

15. Click OK to save the change and exit the Group Properties dialog.

16. Now you can activate the appropriate groups, segments, and elements in the new HL nested looping structure. Please see *Activating EDI Map Components* on page 44 for more information on activating.
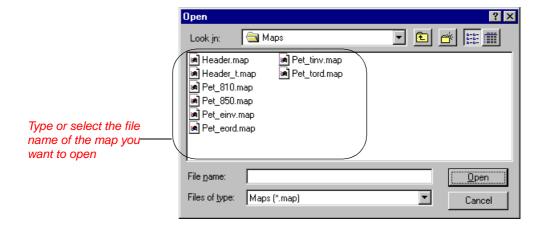
Opening a Map

Opening a map loads it into GENTRAN:Director and enables you to view or edit it. GENTRAN:Director allows you to have multiple maps open, but only one map has focus.

Complete the following steps to open a map:

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

Type or select the file name of the map you want to open

2. The "Look in" list should reflect the drive\folder where GENTRAN:Director is installed. The default is GENTRAN Director\BIN\Maps. 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 .MAP.

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 map.

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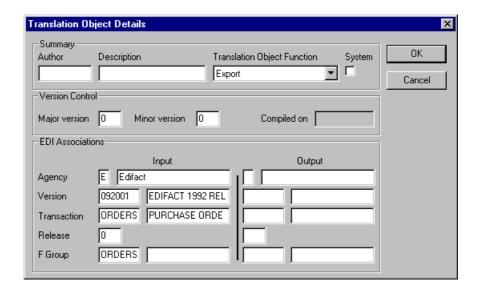
The Save function writes (saves) the currently selected map to disk. If the map was saved before and you completed the Translation Object Details dialog, complete the following step to save the current map:

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

To save a new map, complete the following steps:

1. Select Save from the File menu or the Main Toolbar. The Translation Object Details dialog is displayed.

2. Complete the Author box. This box typically contains the name of the person who created the map.

3. Complete the Description box. This box must be unique because it is used by the system to identify the map. We recommend that you use some combination of the following identifying characteristics: the type of map (e.g., Import, Export), the partner that this map is used for, and what type of transaction (document) this map uses. For example, "Export 850 MWT" is the description of an export map for an ANSI Purchase Order (850) for partner MWT.

4. Verify that the correct type of map is selected in the Translation Object list.

5. 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).

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

6. 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.

7. The Input and Output boxes (Agency, Version, Transaction, Release, and F Group) contain information only if the Input and/or Output sides of the map are of EDI format. You can change the information in these boxes, but it does not alter the content of the map. You may wish to alter these boxes if, for example, you want the map to reflect a standards version that is not loaded on your system. You can change the version on this dialog, and then alter the map to be compliant with that version.

8. Click OK to save the translation object details and access the Save As dialog.

9. The "Save in" list should reflect the drive where GENTRAN:Director is installed. The default is GENTRAN Director\Maps. If the current folder is not the correct one, select the appropriate folder.

10. Type the name of the map (one to eight characters) in the "File name" box. The default file extension is .MAP.

11. Click SAVE to save the map.

Naming a Map

You allocate a filename to a map when you save the map for the first time or use the Save As function to save a copy of the map under a different name. The name of the map must be 1-8 unique characters that define the map in a meaningful way. For example, "850_MWT.MAP" is the name of a map for an inbound ANSI Purchase Order (850) for partner MWT.

The naming conventions in GENTRAN:Director are as follows:

■   The source map (map created in GENTRAN:Director) should be named using the file extension **.MAP** (i.e., MWT_850.MAP)

■   The compiled translation object (translation object that is the result of compiling the source map) should be named using the same filename that was allocated to the source map BUT with the file extension **.TPL** (i.e., MWT_850.TPL)

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

i   **Be very careful not to overlay the source map 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 map under a different name:

1. Select Save As... from the File menu to display the Save As dialog.

2. In the "File name" box, type the name of the map copy (using the default .MAP file extension) and click SAVE to save the map copy under the new name and exit the Save As dialog.

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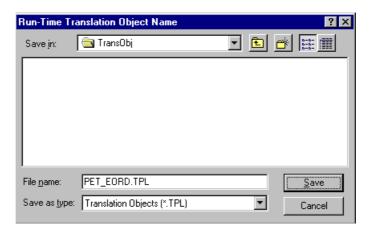
The Compile function compiles the map and creates a translation object. The map that you create using GENTRAN:Director is a *source map*. When that source map is compiled, the result is a *compiled translation object*.

This translation object needs to be registered with the GENTRAN:Director system before you can use it. You use the Compile function after the map is completed and saved. Please see *Saving a Map* on page 58 for more information on saving a map.

Complete the following steps to compile a map 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. Type the name of the translation object in the "File name" box. The name of the translation object is 1-8 characters. Use the default .TPL file extension.

    We recommend that you name the translation object (.TPL file) the same filename (one to eight characters long) as you named the map (.MAP file). Preserving the same filename (with different file extensions) means that the relationship between the source map and the compiled translation object remains evident. For example, if the source map name is MWT_850.MAP, the compiled translation object name is MWT_850.TPL.
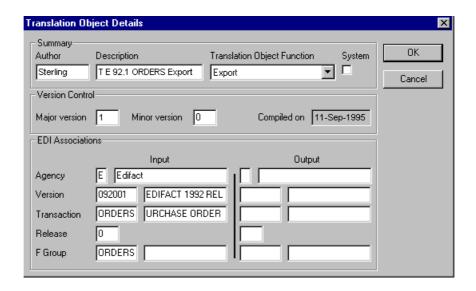
STOP **Be very careful not to overlay the source map with the compiled translation object. Use the .TPL file extension to distinguish the translation object.**

3. From the "Save in" list, change the drive\folder where the compiled translation object is stored, if necessary.

STOP **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.**

4. Click SAVE and the system compiles the map and generate a translation object. The Compile Error dialog is displayed. Verify that no errors occurred and click OK to exit the dialog. The date that the translation object was compiled on is automatically loaded into the "Compiled on" box on the Translation Object Details dialog.

5.   Select Save from the File menu to save the source map with the Compiled on date.

6.   The translation object needs to 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 Mapping Report

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

> **You can print a mapping report to a file by selecting**
> **Print to File... from the File menu. Change the defaults**
> **on the Print Options dialog 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 mapping 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 mapping 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
     mapping 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 mapping report is printed.

Formatting Your Data

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

Using the String Type

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

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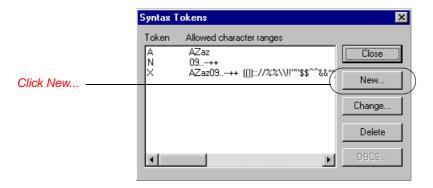
Creating a Syntax Token (Western European languages)

Complete the following steps to create a syntax token:

ℹ️ **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.
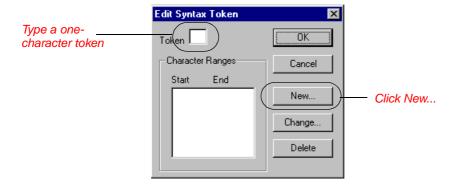
Click New...

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.

Type a one-character token

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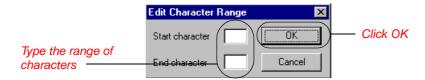
3.   In the Token box, type the unique one-character alphanumeric value that the system recognizes as containing the allowed range of characters you designate.

> **STOP** **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 alphabetics.

4. 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.

Type the range of characters   *Click OK*

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

5. 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.

> **STOP** **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 map 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.
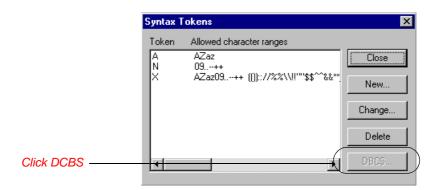
> **i** **When you set up a DBCS token, it applies only to that map. You may need to set one up for each map that you create.**
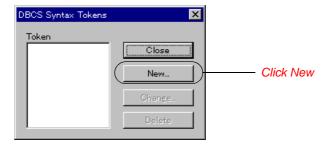
Complete the following steps to create a DBCS syntax token:

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

Click DCBS

2. Click DCBS to access the DCBS Syntax Token dialog.

Click New

3. 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.**

4. 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.
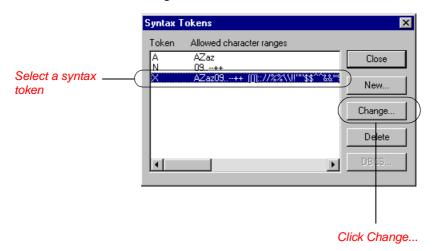
    If *no*, continue with step 6.

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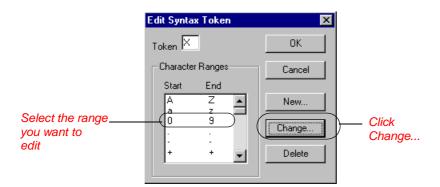
1. Select Syntax Tokens from the Edit menu to display the Syntax Tokens dialog.

Select a syntax token

Click Change...

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 a field/element, dictate the way that field/element must be formatted. If the field or 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 Tokens dialog.

Select the range you want to edit   *Click Change...*

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.

The Character Ranges list contains the character range or ranges that you defined 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" allows upper-case *and* lower-case alphabetics only.

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.

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.
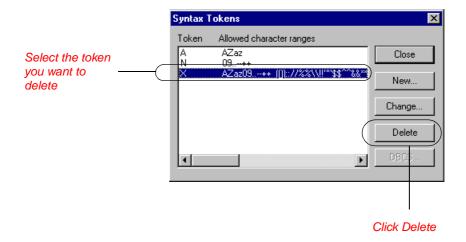
9.   Click OK to save the syntax token and return to the Syntax Tokens
     dialog.

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.

*Select the token
you want to
delete*

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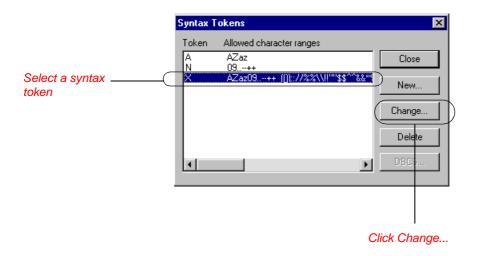
2.   Highlight the token that you want to delete and click DELETE.

STOP   **The selected entry is deleted without warning.**
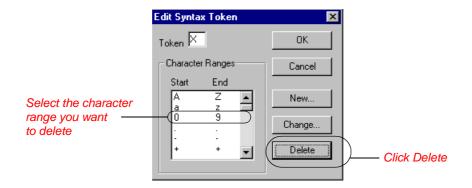
Complete the following steps to delete a character range from a syntax token:

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

Select a syntax token

Click Change...

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

Select the character range you want to delete

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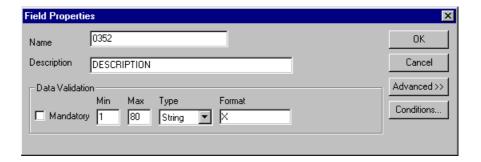
3. Highlight the character range that you want to delete and click DELETE.

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 or field. The use of a syntax token enables you to define what characters should be used while compliance checking this element/field (i.e., alphanumeric within a certain range, numeric within a certain range, etc.).

Complete the following steps to use syntax tokens:

1.   Double-click an existing element or field, or create a new element or field. Please see *Using a Fixed Format File* and *Using an EDI File* for more information on creating fields and elements. The Field Properties dialog is displayed.

2.   From the Type list, select String. This indicates that the element/field contains characters.

3.   In the Format box, type an existing syntax token to denote that this element/field must be formatted as the specified syntax token dictates. Please see *Creating a Syntax Token (Western European languages)* on page 67 for more information on creating syntax tokens.

4.   Click OK to exit the Field Properties dialog.
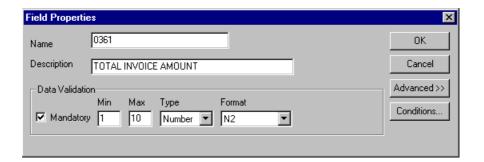
Using the Number Type

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

Complete the following steps to use the number type:

1.   Double-click an existing element or field, or create a new element or field. Please see *Using a Fixed Format File* and *Using an EDI File* for more information on creating fields and elements. The Field Properties dialog is displayed.

2.   From the Type list, select Number. This indicates that the element/field is a numeric or real number that can be mathematically manipulated.

3. From the Format list, select the appropriate real or numeric option. The valid selections are as follows:

■   R0 (this field/element has an explicit decimal point and no decimal places)

■   R1 (this field/element has an explicit decimal point and up to one decimal place)

■   R2 (this field/element has an explicit decimal point and up to two decimal places)

■   R3 (this field/element has an explicit decimal point and up to three decimal places)

■   R4 (this field/element has an explicit decimal point and up to four decimal places)

■   R5 (this field/element has an explicit decimal point and up to five decimal places)

■   R6 (this field/element has an explicit decimal point and up to six decimal places)

■   R7 (this field/element has an explicit decimal point and up to seven decimal places)

■   R8 (this field/element has an explicit decimal point and up to eight decimal places)

■   R9 (this field/element has an explicit decimal point and up to nine decimal places)

■   N0 (this field/element has an implied decimal point and no decimal places)

■   N1 (this field/element has an implied decimal point and up to one decimal place)

■   N2 (this field/element has an implied decimal point and up to two decimal places)

■   N3 (this field/element has an implied decimal point and up to three decimal places)

■   N4 (this field/element has an implied decimal point and up to four decimal places)

■   N5 (this field/element has an implied decimal point and up to five decimal places)

■   N6 (this field/element has an implied decimal point and up to six decimal places)

■   N7 (this field/element has an implied decimal point and up to seven decimal places)

■ N8 (this field/element has an implied decimal point and up to eight decimal places)

■ N9 (this field/element has an implied decimal point and up to nine 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".**
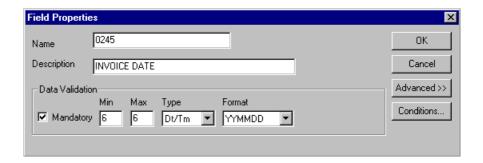
4. Click OK to exit the Field Properties dialog.

Using the Dt/Tm Type

A Dt/Tm type field or element contains a date or time. If you specify that a field or 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:

1. Double-click an existing element or field, or create a new element or field. Please see *Using a Fixed Format File* and *Using an EDI File* for more information on creating fields and elements. The Field Properties dialog is displayed.

Field Properties					☒
Name	0245				OK
Description	INVOICE DATE				Cancel
Data Validation					Advanced >>
	Min	Max	Type	Format	Conditions...
☑ Mandatory	6	6	Dt/Tm ▼	YYMMDD ▼	

2. From the Type list, select Dt/Tm. This indicates that the element/field is a date or time.

3. From the Format list, select the appropriate date or time option.

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)
DDDYY	Three-digit Julian day, last two digits of year (example: 34997)
YYYYDDD	Four-digit year, three-digit Julian day (example: 1997349)
DDDYYYY	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)

4. Click OK to exit the Field Properties dialog.

Creating Simple Field-to-Field Links

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

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

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

Complete the following steps to link two fields:

1.   Select Link from the Functions menu or the Main Toolbar.

2.   Click a field on the Input side of the map. The cursor changes to a "Link Arrow":
     LINK

3.   Click the field on the Output side of the map that the data from the field selected on the Input side of the map is mapped to.

4.   A line connects the two fields to visually represent the link.

> **If a line does not appear when the two fields are linked, check the Preferences dialog. Please see *Customizing Global Display of Links* on page 91 for more information. On the Links tab of the Preferences dialog, you should specify "Show all links possible."**

Customizing Your Display

The GENTRAN:Director Application Integration subsystem allows you to customize the display of maps in several different ways:
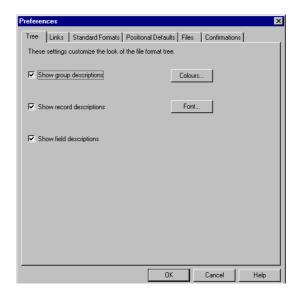
■ Customizing global display options

■ Customizing colours (global option)

■ Customizing fonts (global option)

■ Displaying links (global option)

■ Making the two sides of a map equal

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

Customizing Global Display Options

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

1. Select Preferences... from the Options menu to display the Preferences dialog (the Tree tab is displayed by default). The Preferences dialog 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 remove the checkmark.

    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 map if the descriptions are not displayed.

    You may want to experiment with shrinking the size of the font for the map (see *Customizing Global Fonts* on page 89) before you turn off the display of descriptions.

3. Click the OK button to save changes and exit the Preferences dialog.
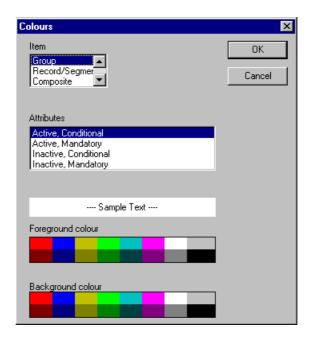
Customizing
Global Colours

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

Complete the following steps to customize colours for all maps:

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

2.   Click COLOURS... to display the Colours dialog.
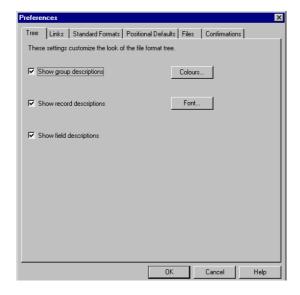
3.   Select the type of map component from the Item list (Group, Record/
      Segment, Composite, Field/Element).

4.   Further define the type of map 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 maps.
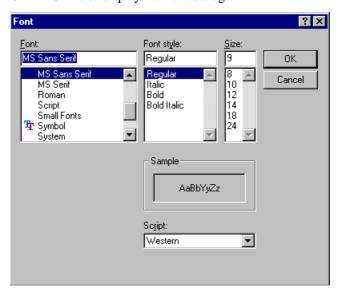
Customizing
Global Fonts

The Font 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 MS Sans Serif 9 point. The Font feature enables you to change the font type, style, and point size. This gives you the flexibility to shrink the font if you need to view more of the map on your monitor, enlarge the font, or change the type and style to be more easily readable by you.

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

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

2.   Click FONT... to display the Font dialog.
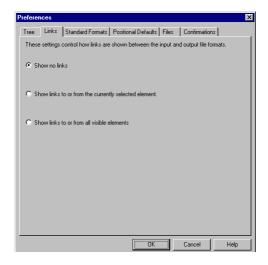
3.   Select the type of font from the Font box. The default is MS Sans Serif.

4.   Select the style from the Font Style box. The default is Regular.

5.   Select the point size from the Size box. 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 maps.

Customizing Global Display of Links

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

Complete the following steps to customize the global display of mapping links:

1. Select Preferences... from the Options menu to display the Preferences dialog. The Preferences dialog enables you to set global defaults for GENTRAN:Director.

2. Click the Links tab to display the linking options. The Links tab enables you to set the default display of mapping links.

2. The options on the Links tab enable you to select from the following:

■ **Show no links**
Do not visually display mapping links.

■ **Show links to or from the currently selected element**
Display only the mapping links for the currently selected field (this option enables you to concentrate on the selected field and removes the confusion of viewing many links at once).

■ **Show links to or from all visible elements**
Display all the mapping links.

Click the option for the link display option you want to be the default for all maps.

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

Making the Two Sides of a Map Equal

When you open a map, the Input and Output sides of a map are displayed in equal dimensions. After manipulating a map and moving the center bar that divides the display of the Input and Output sides, you may wish to again make the two sides equal. The Equalize function enables you to reinstate the two sides of the map with focus in equal dimensions. The use of the Equalize function is optional.

Complete the following step to make the two sides of a map equal:
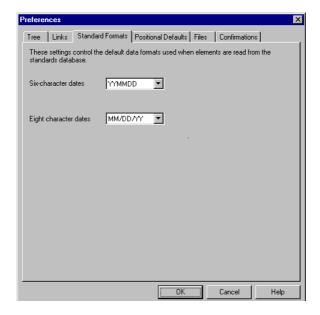
1. Select Equalize from the View menu. This moves the center dividing bar between the two sides of the map with focus, so that the two sides of the map are equally proportioned.

Setting Default Date Format

You typically establish the default date format for all date fields *one time only*. This default is used when EDI documents are initially loaded from the standard. This default can be overridden on the Field Properties dialog. This global option changes the default date format for all maps. However, the format of the existing date fields do not change; the default is only used for new maps.

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 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 the 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 Map Components to Use

The map components for an EDI File are the following:

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

The map components for a Positional File are the following:

- Positional File
- Record
- Field

The specific EDI map components that you use depends on the type of map 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 sides of the map, 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 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.

> **As an alternative to activation, you can use the Auto Trim feature if you have a sample EDI file defined. Please see *Setting up the Document Name and Reference* on page 96 for more information on the Auto Trim feature.**

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

1. Select Activation from the Functions menu or the Main Toolbar.

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

2. Open (double-click) each group that contains segments or groups that need to be activated.

3. Click each inactive group that you need to use. This activates the groups.

> **If you accidentally click a map component that you did not mean to activate, right-click the map component and select Deactivate from the shortcut menu.**

4. Click each inactive segment that you need to use. This activates the segments.

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. This activates the composites.

7. Open (double-click) each composite that contains elements that need to be activated.

8. Click each inactive element that you need to use. This activates the elements.

9. Once you have activated all the necessary groups, segments, and elements, click the Activation button on the Main Toolbar to turn activation mode off.

Setting up the Document Name and Reference

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

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 Application Integration subsystem, the Name box of the document browsers that contain this document in GENTRAN:Director contains the data from that mapped field. If you set up the Reference in the Application Integration subsystem, the RefData box of the document browsers that contain this document in GENTRAN:Director contains the data from that mapped field.

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

■   Change the field's data type to "String" on the Field Properties dialog.

     OR

■   Set up an extended rule to update the document record.
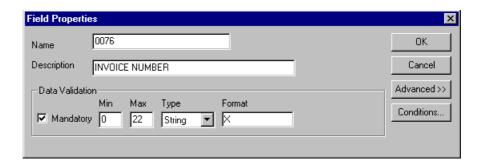
> **STOP**   **If you change a Dt/Tm or Number field's data type to "String," you lose the data type compliance checking for the original type.**

Complete the following steps to set up the Document Name:

1.   We recommend that you select a non-recurring field from the header or trailer record of the EDI side of the document. The field that you select varies for each type of document. The selected field should be something that identifies this document meaningfully in GENTRAN:Director. For example:

■   For an Invoice this field is typically the Invoice Number.

■   For a Purchase Order this field is typically the Purchase Order Number.

2.   Right-click the appropriate field and select Properties... from the
     shortcut menu. The Field Properties dialog is displayed.

Field Properties					
Name	0076				OK
Description	INVOICE NUMBER				Cancel
Data Validation					Advanced >>
	Min	Max	Type	Format	Conditions...
☑ Mandatory	0	22	String ▼	X	

3.   Click ADVANCED>> to display the Rules section of the dialog.

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 a field from the first record of the EDI side of the document. The field that you select varies for each type of document. The selected field should be something that identifies this document meaningfully in GENTRAN:Director.
    For example:

    ■ For an Invoice this field is typically the Purchase Order Number.

    ■ For a Purchase Order this field is typically the Purchase Order Date. However, to use Purchase Order Date, you must use an extended rule on that field, instead of following the steps below.

2. Right-click the appropriate field and select Properties... from the shortcut menu. The Field Properties dialog is displayed.

3. Click ADVANCED>> to display the Rules section of the dialog.

4. Click the Use Standard Rule check box.

5. Click the Update option.

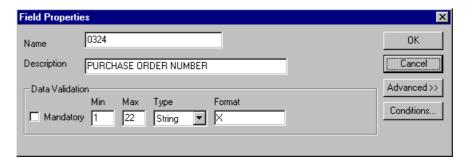6. From the Table list, select Document Record.

7. From the Field list, select Reference Data.

8. Click OK to set up the Reference.

3

Customizing the Properties of Map Components

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

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

■   Customizing the Properties of a Record, Segment, or Looping Structure

■   Using LS and LE Segments

■   Customizing the Properties of a Field or Element

■   Customizing the Properties of a Composite Element

> **Press F1 from any area of the system to access the online Help for that function. The online Help system contains dialog box element definitions and other detailed information.**

Customizing a Record, Segment, or Group

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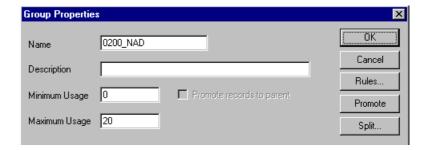
Customizing Group Properties

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.

STOP   **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 Description box. This box is used to provide a brief explanation of the loop.

4. In the Minimum 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 Maximum Usage box, type the maximum number of times the loop can be repeated.

6. Click OK to save the changes to the group.
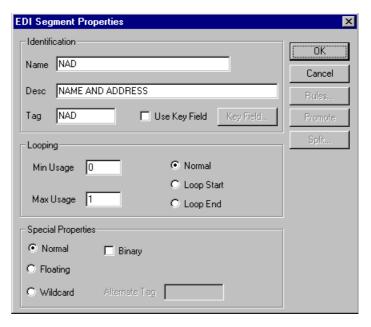
Customizing Segment Properties

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 105 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 EDI 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/record or element/field occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where "n" is the number of the occurrence in the map.**

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. If you want to modify the segment tag, type the segment identification code <TAG> in the Tag box. For example, a record is recognized by the system as:

    <TAG>[Delimeter]<DATA>[Delimeter]<DATA>.......<DATA> [Segment Terminator]

    The segment TAG for each segment enables the system to recognize that segment.

5. 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."**

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

7. 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 |

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

Customizing Record Properties

You can modify the properties of any record, including the minimum and maximum number of times the record can repeat, whether the record is mandatory or conditional.

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

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

     The Positional Record Properties dialog is displayed.

2.   In the Name box, type the new record name.

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

3.   In the Desc box, type a description of the record. This box is used to provide a brief explanation of the record that allows you to differentiate it from similar records.

4. In the Tag box, type the record identification code <TAG>. For example, a record is recognized by the system as:

<TAG>[Field_1][Field_2]

The record TAG for each record enables the system to recognize that record and then determine the mapping requirements.

> **We recommend that you define the record tag on the Positional Record Properties dialog, instead of defining fields with the purpose of explicitly containing the record tag.**

5. In the Position box, type the starting column position of the tag in the data record.

The Looping section of the Positional Record Properties dialog enables you to specify the minimum and maximum number of times this record can repeat, and whether a record is mandatory or conditional. Alter this section *only* if the record is a looping structure. Otherwise, accept the defaults.

6. In the Min Usage box, type the minimum amount of times the record must repeat.

> **If the Min Usage box contains a "0" (zero), the record is "conditional." If the Min Usage box contains a "1" or greater, the record is "mandatory."**

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

8. Click OK to save the changes to the record.

Using Loop Start and Loop End Segments

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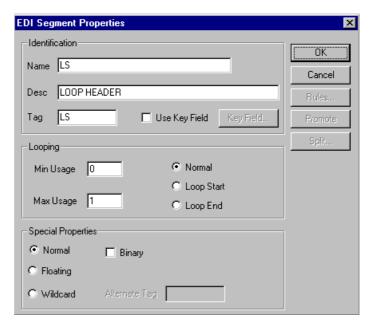
> **You need to communicate with your trading partner to determine which loop identification code your partner sends you (for an inbound map), or expects to receive (for an outbound map).**

Defining a LS Segment Inbound
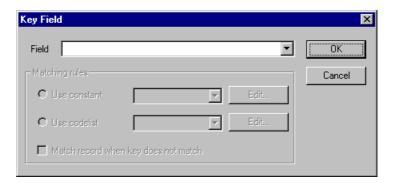
Complete the following steps to define an LS segment for an inbound map:

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

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 list 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.

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

    Click the Use constant option.

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

Translation Object Constants

ID	Type	Value
Sub Table 1		

Close

New...

Change...

Delete

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

Edit Constant

ID

Type   OK

Value   Cancel

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.
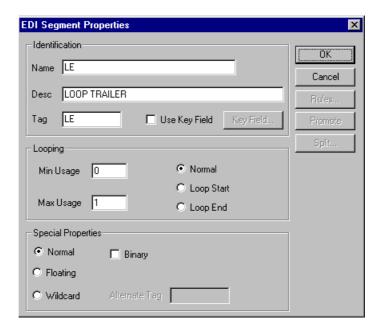
     For an inbound map, this is the loop identifier code that you expect to receive from your trading partner.

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 EDI Segment Properties dialog.

Defining a LE Segment Inbound
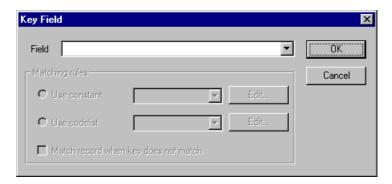
Complete the following steps to define an LE segment for an inbound map:

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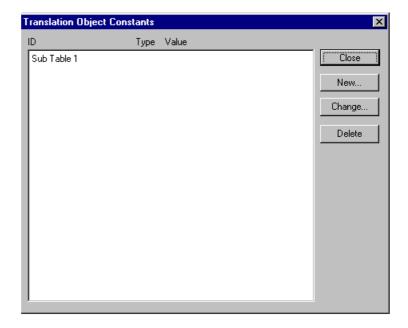

2. Verify that the Loop End option in the Looping section of the dialog is selected.

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 list 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 map.

   Select the constant you created for the LS segment from the list. The system matches the selected constant against the loop identifier code.
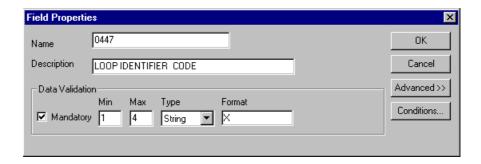
7. Click OK to exit the Key Field dialog.

8. Click CLOSE to exit the EDI Segment Properties dialog.

Defining a LS Segment Outbound
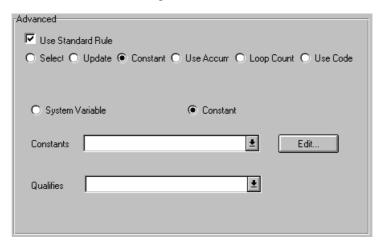
Complete the following steps to define an LS segment for an outbound map:
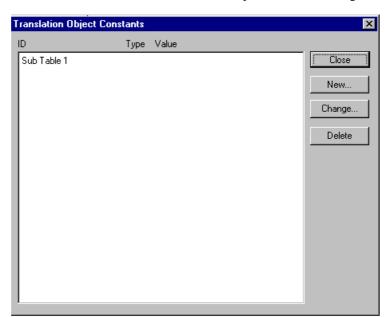
1. Double-click the Loop Identifier Code field in the LS segment.

   The Field Properties dialog is displayed.

2. Click ADVANCED>> to display the Rules section of the dialog.

3. Click the Use Standard Rule check box.

4. Click the Constant option.

5. Click the second Constant option.

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

7. Click NEW... to access the Edit Constant dialog.
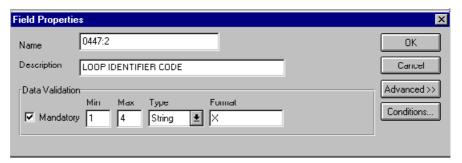
8. In the ID box, type the literal constant identifier.

9. From the Type list, select "String." This the category of this literal constant.

10. In the Value box, type the value of the literal constant. For an outbound map, this is the loop identifier code that your partner is expecting to receive from you.

11. Click OK to add the constant to the system.

12. From the Constants list, select the constant that you created. The system matches the selected constant against the loop identifier code.

13. Click CLOSE to exit the Field Properties dialog.

Defining a LE Segment Outbound
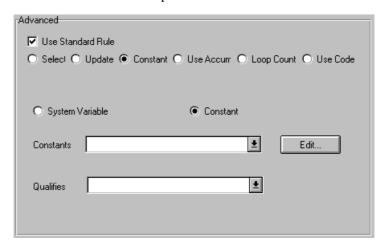
Complete the following steps to define an LE segment for an outbound map:

1.   Double-click the Loop Identifier Code field in the LE segment.

The Field Properties dialog is displayed.

2.   Click ADVANCED>> to display the Rules section of the dialog.
3.   Click the Use Standard Rule check box.
4.   Click the Constant option.
5.   Click the second Constant option.
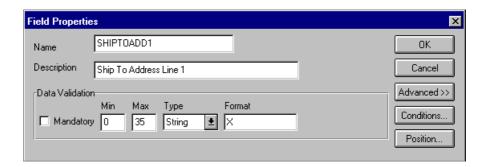
6.   From the Constants list, select the constant that you created. The system matches the selected constant against the loop identifier code.

7.   Click CLOSE to exit the Field Properties dialog.

Customizing a Field or Element

You can modify the properties of the fields and 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 a field or element:

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

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

STOP   **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 field/element, type the new description in the Description box. The description is used to provide a brief explanation of the field/element that allows you to differentiate it from similar fields/elements.

4.   If you want to designate the field/element as "mandatory" (must be present), click the Mandatory check box.

5.   To change the minimum number of characters that is acceptable for the field/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 field/element, type the maximum length in the Max box.

7.   From the Type list, select the type of the field/element. Valid values are:

String   =   alphanumeric field/element
Number   =   numeric or real field/element
Dt/Tm   =   date or time field/element

8.   From the Format list, select how the field/element is formatted. The choices for this field depend on the type of the field/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 field/element must be formatted as the specified syntax token dictates (the default syntax token is "X"). Please see *Using Syntax Tokens* on page 77 for more information on defining and using syntax tokens.

9.   Click OK to save the changes to the field/element.

Defining and Modifying Relational Conditions

You can use relational conditions to connect fields together for syntax or compliance reasons. For example, Field A is invalid unless Field B is present. Therefore, if you set up a condition that pairs Fields A and B, the system generates a compliance error if one of those fields is not present.
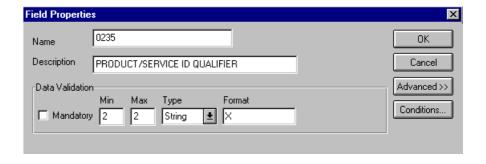
You can also view the conditional relationships between elements, as provided by the standard.
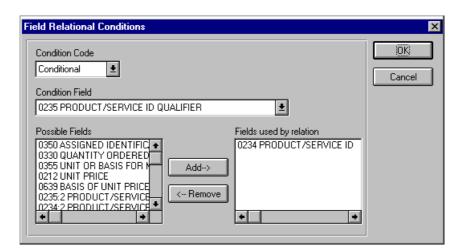
STOP **The system enables you to edit EDI conditional relationships, but if you do, you generates a compliance error. We recommend that you view edi conditions only.**

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

1. Double-click the field or 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 field 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 field connection condition from the Condition Code list. Valid values are:

■   **Paired/Multiple** – if any of the specified fields are present then all fields must be present

■   **Required** – at least one of the specified fields must be present

■   **Exclusion** – no more than one of the specified fields may be present

■   **Conditional** – if the first Condition field is present, the rest of the fields must also be present

■   **List Conditional** – if the first Condition field is present, at least one of the specified fields must also be present

4.   Select the first field from the Condition Field list. This is the field 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 fields in the translation object that are valid to be used in a condition at this point. Highlight a field or fields in the Possible Fields list and click the ADD--> button to move the fields to the Fields used by relation list, to include the fields 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 fields that you selected to be a part of the conditional relationship. To remove the fields from the conditional relationship, highlight a field or fields in the Fields used by relation list and click the <--REMOVE button to move the fields back to the Possible Fields list.

7. Click OK to add the conditional relationship to the field.

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 maps, 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/record, composite, or element/field occurs more than once in a map it is identified by its name <ID>. The second and subsequent occurrences are identified by <ID>:n, where "n" is the number of the occurrence in the map.**

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.

4

Using Standard Rules

The GENTRAN:Director Application Integration subsystem provides you with standard rules that you can apply to fields and elements. Standard rules give you access to functions that are necessary for mapping operations that are more complex than simple linking, but less involved than extended rules. Extended rules enable you to use a GENTRAN:Director proprietary programming language to perform virtually any mapping operation you require. Please see *Creating Simple Field-to-Field Links* on page 84 for more information on linking. Please see *Introducing Extended Rules* on page 244 for more information on extended rules.

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

■   Using Accumulators

■   Using Code List Tables
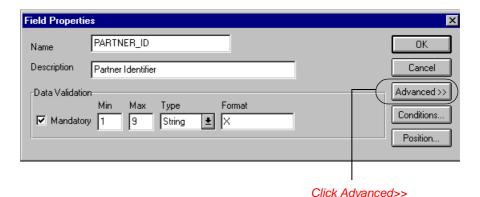
Press F1 from any area of the system to access the online Help for that function. The online Help system contains dialog box element definitions and other detailed information.

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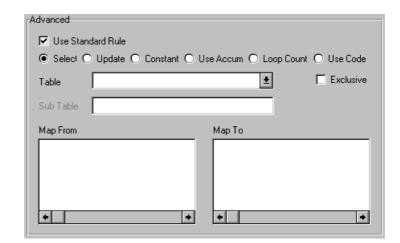
The Select function enables you to select entries from a location table, cross-reference table, partner table, or lookup table (all tables created in the GENTRAN:Director Partner Editor). You can then map the fields in those tables to one or more fields in the data. The Select function uses the value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

Complete the following steps to use the Select function:

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

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 field.

> **The Sub Table box is only active 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/field from the specified table entry from which you want to map the contents.

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

Map To   From this list, choose the element/field to which you want to map the contents of the Map From box. Each element/field 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 are displayed 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.**
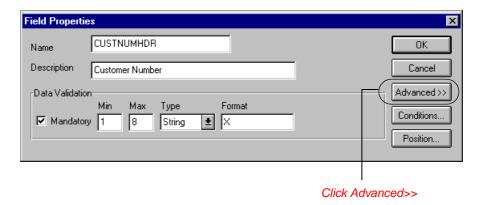
Example 1

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

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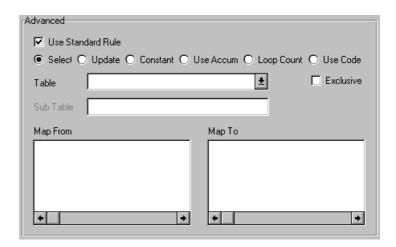
Complete the following steps to map the customer number from the partner definition:

1. Double-click the CUSTNUMHDR field to display the Field Properties dialog.
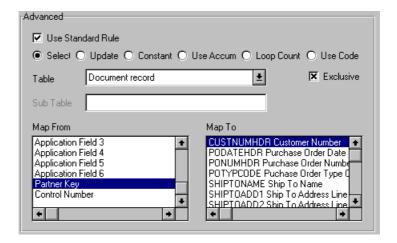
Click Advanced>>

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.

5. From the Table list, select "Document record." This indicates that the system updates this field with selected information from the document record.

6. Click the Exclusive check box to indicate that a compliance error should be generated if the select does not find a valid entry.

7. From the Map From list, choose "Partner Key." This is the field that contains the customer number for this partner.

8. From the Map To list, choose "CUSTNUMHDR." This is the field to which the information from the partner key of the document record is mapped.

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

9. Click OK to add the standard rule to the CUSTNUMHDR field.

Example 2

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

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

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

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

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

Complete the following steps to translate your partner's customer product code to your UPC Code for each item your partner ordered:

1. Map the customer product code from the Product/Service ID element to your CUSTPROCODE field. Please see *Creating Simple Field-to-Field Links* on page 84 for more information on linking.

2.   In the Partner Editor, create a cross-reference table named
     "PRODCODE." See the GENTRAN:Director *User's Guide* or online
     Help for more information on creating a cross-reference table.
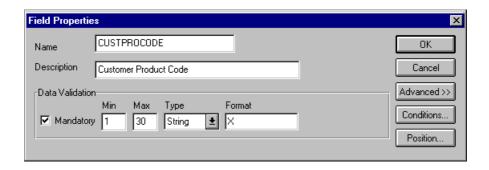
> **The cross-reference table you create only contains six
> entries. This is an example – an actual table would
> contain all of the items manufactured by your
> company that your partner purchases.**
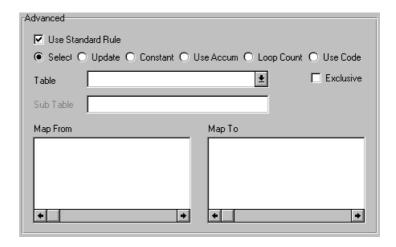
Use the information in the following table when you create the cross-
reference table. The "My Value" box contains your UPC code for the
item. The "Partner Value" box contains your partner's customer product
code for the item. The "Description" box contains a description (and
size, if applicable) of the item, and the Reference Data box contains
additional quantity or color information.

My Value	Partner Value	Description	Reference Data
12345678901234	555666411	Large Dog Collar	Red
23456789012345	555666817	Spring Breeze Cat Litter	10# Bag
34567890123456	555666908	Small Beef Basted Rawhide Bones	Bag of 12
45678901234567	555666321	X-Large Food Bowl	Black
56789012345678	555666287	FatKat Kitty Bed	Blue with White Trim
67890123456789	555666413	Catnip Mouse Toy	

3.   Double-click the CUSTPROCODE field. The Field Properties dialog is
     displayed.
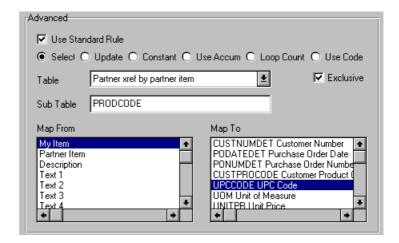
4. If the Advanced section of the dialog is not currently displayed, click ADVANCED>> to access this section, which enables you to define rules.

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

6. Verify that the Select option is selected.

7. From the Table box, select "Partner xref by partner item." This indicates that you are using a cross-reference table and translating by your partner's item.

8. In the Sub Table box, type "PRODCODE." This is the name of the cross-reference table you created.

9. Click the Exclusive check box to signal the system that if this translation fails, you want the system to generate an error.

10. From the Map From list, choose "My Item." This is the field from which the contents is mapped.

11. From the Map To list, choose "UPCCODE." This is the field to which the information from the lookup table is mapped.

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

12. Click OK to map from the cross-reference table.

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 (if the map side format type is EDI), with the contents of the element or field.

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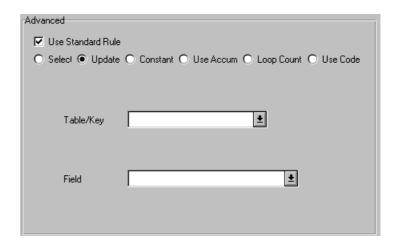
1. Double-click an existing element or field, or create a new element or field (please see *Using a Fixed Format File* and *Using an EDI File* for more information on creating fields and elements). The Field Properties dialog is displayed.

| Field Properties	✕
Name	1004	OK
Description	DOCUMENT/MESSAGE NUMBER	Cancel
Data Validation		Advanced >>
	Min Max Type Format	Conditions...
☐ Mandatory	1 22 String ▼ X	

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.   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 field/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 – if the format type of this side of the map is EDI)

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

Example

We recommend that you set up a Document Name for each map, to make the identification of a document created by this map easier in GENTRAN:Director. This allows you to differentiate between documents in the document browsers in GENTRAN:Director.
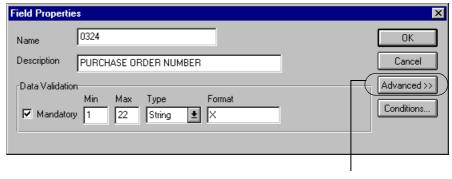
To set up a Document Name, you must select an element or field in the map that contains data that distinguishes the documents translated by this map. Then, the Name box of the document browsers that contain this document contains the data from the selected element or field.

> **You can choose any element or field as the Document Name, but the element or field 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:
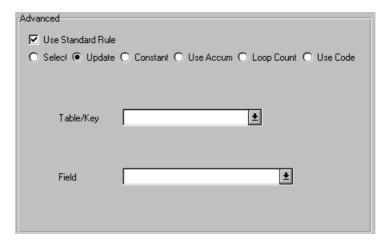
1. Double-click the BEG03 (0324) element to display the Field Properties dialog.
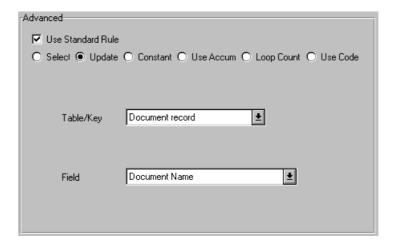
2. If the Advanced section of the dialog is not currently displayed, click ADVANCED>> to access this section, which enables you to define rules.

3. Click the Use Standard Rule check box in the Advanced section, to display the rule options.

4.   Click the Update option.

5.   From the Table/Key list, choose "Document record." This indicates that
     you are updating the document record in GENTRAN:Director.

6.   From the Field list, choose "Document Name." This indicates that you
     are updating the Name box in the document browsers with the contents
     of this element.

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

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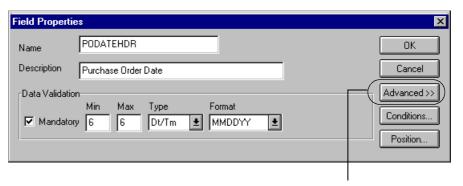
The Constant function enables you to move a literal constant value to the specified element or field and indicate a qualifying relationship with another element or field.

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 or field. The selected element or field must have a data type of "Dt/Tm."

Complete the following steps to use a system variable:
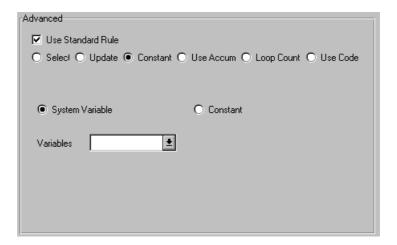
1. Double-click an existing element or field, or create a new element or field (please see *Using a Fixed Format File* and *Using an EDI File* for more information on creating fields and elements). The Field Properties dialog is displayed.

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

The dialog box element definition is as follows:
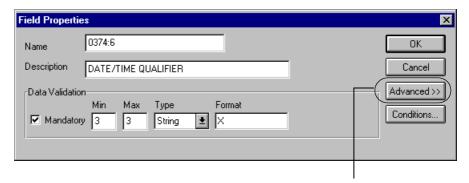
Variables   Select current date or current time from the Variables list to map that
                 variable to the element or field.

Using a Constant

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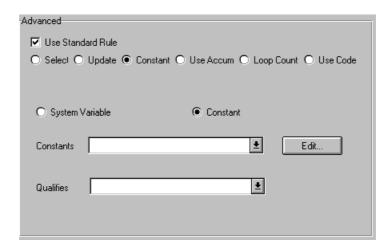
Complete the following steps to use a constant:

1. Double-click an existing element or field, or create a new element or field (please see *Using a Fixed Format File* and *Using an EDI File* for more information on creating fields and 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. Click the Constant option.

5.   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 or field. 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 or field from the Qualifies list if you want to indicate that the current element or field qualifies the selected element or field (establish a qualifying relationship between the two elements or fields). If the Qualifies element is not generated because of a lack of data, the constant is not moved to the current element or field.

Edit...   This button enables you to access the Translation Object Constants dialog, which allows you to create, edit, and delete literal constants.
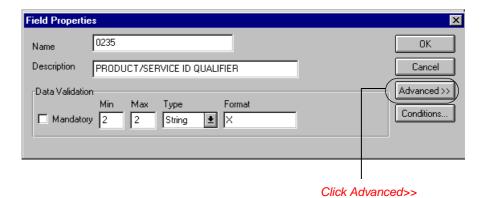
Example

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

In this example, you use a constant to define a qualifier for a Product/Service ID.
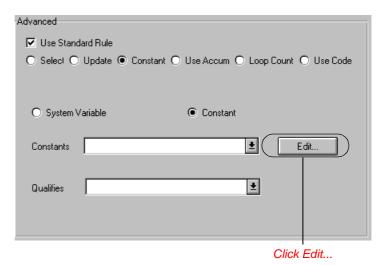
Complete the following steps to define a qualifying relationship:

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.
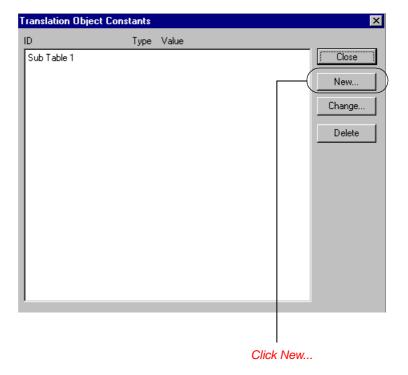
Click Advanced>>

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. Click the Use Standard Rule check box.

4. Click the Constant option.

5.   Click the second Constant option.

Click Edit...
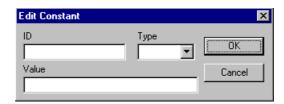
6.   Click EDIT... to access the Translation Object Constants dialog.

Click New...
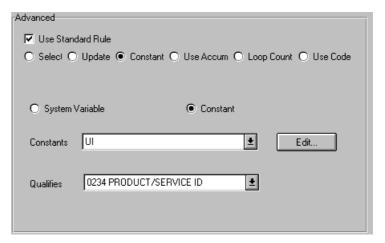
7.   Click NEW... to access the Edit Constant dialog.

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

9.   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.

10. In the Value box, type "UI" to indicate that the Product/Service ID field must contain the UPC Code. This is the value of the literal constant.

11. Click OK to add the constant to the system.

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

13. 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* fields or elements in the same record or segment as the qualifying field.

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

14. Click CLOSE on the Field Properties dialog and the qualifying relationship between the two elements is established.
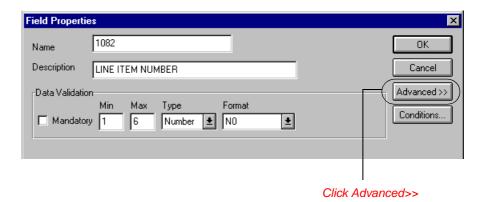
Using the Use Accum Standard Rule

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 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.

Please see *Using Accumulators* for more information about defining accumulators.
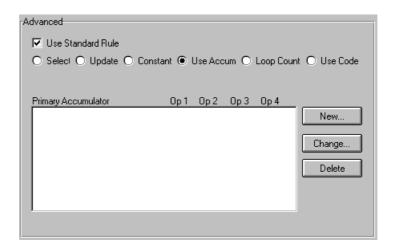
Complete the following steps to use the Use Accum function:

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

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.   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 or field.

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

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 map.

Delete   Highlight a calculation in the Primary Accumulator list and click this button to delete the selected calculation.
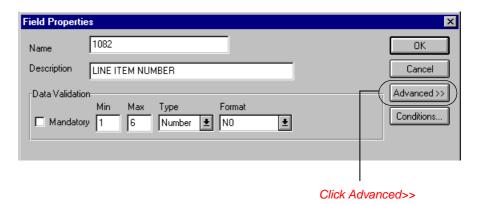
STOP   **The selected calculation is deleted without warning.**

Using the Loop Count Standard Rule

The Loop Count function 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).
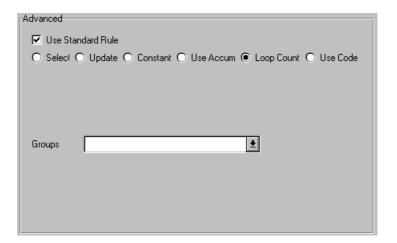
Complete the following steps to use the Loop Count function:

1. Double-click an existing element or field, or create a new element or field (please see *Using a Fixed Format File* and *Using an EDI File* for more information on creating fields and 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.   Click the Loop Count option.

The dialog box element definition is as follows:

Groups   Select the loop that you want to count. If the loop is a nested loop, you can
track the current loop <u>or</u> 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 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.
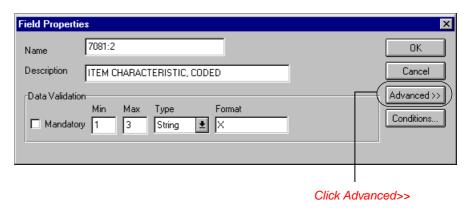
The GENTRAN:Director Application subsystem enables you to create code tables to be used with the current map. You can set up code tables in the Application Integration subsystem to function like the partner cross-reference and lookup tables in GENTRAN:Director. However, code tables that are set up in the Application subsystem can be used *only* for the current map. Code tables that you create in GENTRAN:Director can be used globally for all maps.

A field 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*.
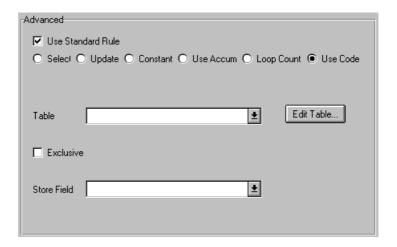
Please see *Using Code List Tables* on page 191 for more information on using code list tables.

Complete the following steps to use the Use Code function:

1. Double-click an existing element or field, or create a new element or field (please see *Using a Fixed Format File* and *Using an EDI File* for more information on creating fields and 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. Click the Use Code option.

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
              field or element *must* contain one of the codes from the specified table
              (nothing else is valid for that field). For example, if the field or 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 field or element, the system flags it as an error.

Store Field Select an element or field 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 map the customer product code from the Product/Service ID to the customer product code field (CUSTPROCODE) on the application side of the map. To ensure that the data that is mapped from the Product/Service ID element really is your partner's customer product code, you need to establish and use a code table with the Product Service ID *Qualifier*.
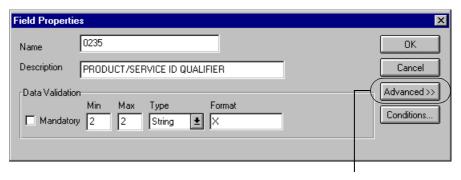
Complete the following steps to validate the data in the Product/Service ID field against a code list table:

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 204 for more information on loading a code table.
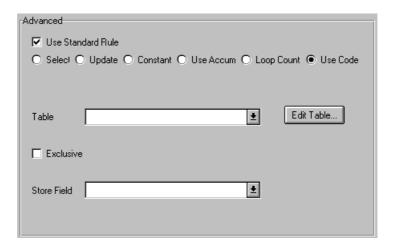
> **Please note that the standards provide code list tables only for elements that use them. For example, in the TD4 segment, the TD401 (Special Handling Code) and TD403 (Hazardous Material Class Code) have code tables provided by the standard.**

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 192 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. If the Advanced section of the dialog is not currently displayed, click
    ADVANCED>> to access this section, which enables you to define rules.

5. In the Advanced section, click the Use Standard Rule check box.

6. Click the Use Code option.
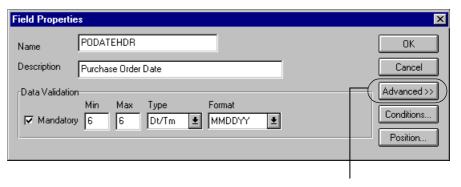
7. From the Table list, select the code list table that the data in this field is
    validated against (0235).

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

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

9. Click OK to add this standard rule to the field.

Using the System Date and Time

The GENTRAN:Director Application Integration subsystem enables you to update any field or element (that has a data type of "Dt/Tm") with the current system date or time.
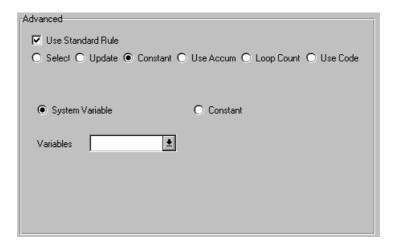
Complete the following steps to update a field or element with the current system date or time:

1. Use the right mouse button to click the desired field or element. Select Properties... from the shortcut menu. 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.

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

5. From the Variables list:

■   Select "Current Date" if you want the field to be updated with the current system date.

    OR

■   Select "Current Time" if you want the field 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 map. Typically, constants are used to load common data into each application record on a map. You create the literal constant and name it. You can "hard code" (enter) a value that is loaded into the field that uses the constant. Or, you can store information from a field 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.

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

A qualifying relationship establishes a relationship between an element and its qualifier. A qualifier contains a code that further defines the element. Qualifying relationships are typically defined in outbound maps. Qualifying relationships are defined using an standard rule, as explained below.
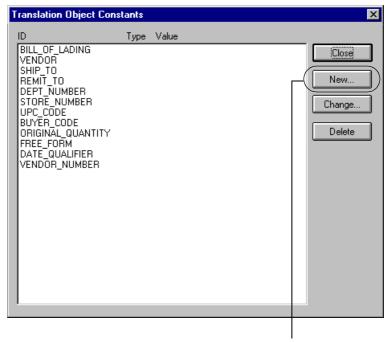
You can perform the following functions with literal constants and qualifying relationships:

■   Defining Literal Constants

■   Editing Literal Constants

■   Deleting Literal Constants

■   Mapping Literal Constants

■   Generating Qualifiers

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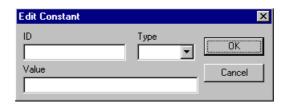
Complete the following steps to create a literal constant so you can use it to store information:

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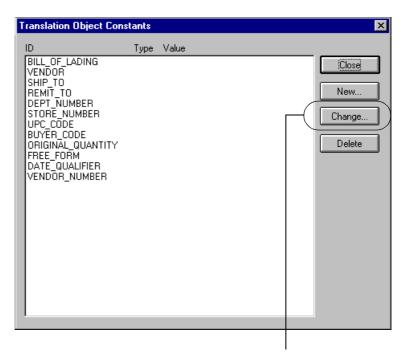
3.   In the ID field, type the literal constant identifier. This is typically a description of the field or element in which the constant is used. If you need to refer to the constant in an extended rule, you should use the data from this field.

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 field, type the actual constant expression. This is the value of the literal constant.

6.   Click OK to add the constant to the system.

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.

3. Click CHANGE... to access the Edit Constant dialog.
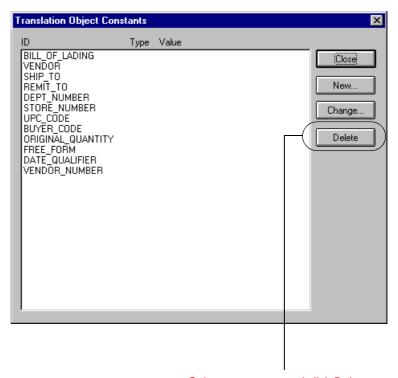
4. Make the changes to the constant.
5. Click OK to accept the changes.

Deleting Literal Constants

Complete the following steps to delete 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 Delete

2.   Highlight the constant you want to delete.

3.   Click DELETE to remove the constant.

STOP   **The selected entry is deleted without warning.**

Mapping Literal
Constants
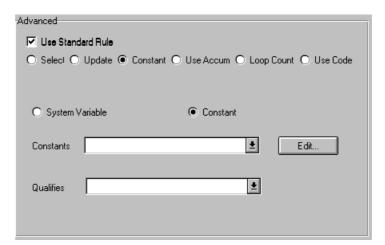
Complete the following steps to map a constant in which you previously stored information using an extended rule:

1. Double-click the field or element in which you want to use the constant. 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.

4. Click the Constant option.

5.   Click the second Constant option.

```
┌─ Advanced ──────────────────────────────────────────────┐
│  ☑ Use Standard Rule                                     │
│  ○ Select  ○ Update  ◉ Constant  ○ Use Accum  ○ Loop Count  ○ Use Code │
│                                                          │
│                                                          │
│     ○ System Variable            ◉ Constant              │
│                                                          │
│  Constants   [_____▼]   [  Edit...  ] │
│                                                          │
│                                                          │
│  Qualifies   [_____▼]                 │
│                                                          │
│                                                          │
└──────────────────────────────────────────────────────────┘
```

6.   From the Constants list, select the constant that you want to use.

7.   Click OK and the data that was stored in the selected constant is loaded in the field.

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:
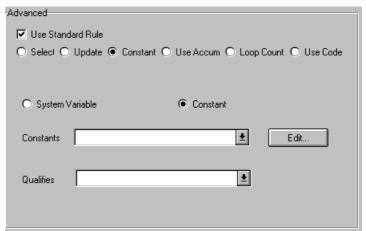
1. Double-click the field or element that you want to use to further define (qualify) another field or element. The Field Properties dialog is displayed.

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. Click the Use Standard Rule check box.

4. Click the Constant option.

5. Click the second Constant option.

```
┌─ Advanced ──────────────────────────────────────────────────┐
│  ☑ Use Standard Rule                                         │
│  ○ Select  ○ Update  ◉ Constant  ○ Use Accum  ○ Loop Count  ○ Use Code  │
│                                                              │
│                                                              │
│     ○ System Variable              ◉ Constant                │
│                                                              │
│   Constants  [                          ▼]   [  Edit...  ]   │
│                                                              │
│                                                              │
│   Qualifies  [                          ▼]                   │
│                                                              │
│                                                              │
└──────────────────────────────────────────────────────────────┘
```

6. From the Qualifies list, select the field or element that this field qualifies.

> 🛈 **The Qualifies list contains only the other *active* fields or elements in the same record or segment as the qualifying field.**

7. Click OK and the qualifying relationship between the two fields is established.

Using Partner-Related Information in Maps
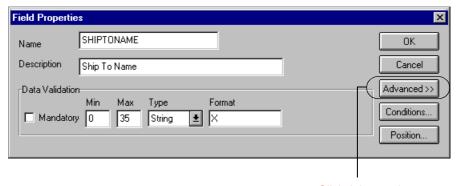
The GENTRAN:Director Application Integration subsystem allows you to use information from the GENTRAN:Director Partner Editor in your maps via standard rules. You can map information from your partner's profile in the Partner Editor to a selected element or field in the map. You can also map information 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

Complete the following steps to map information from the Partner or Internal System Partner Definition:

1. Double-click the field or element to which the standard rule is applied. 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.

4. Click the Select option. The Standard Rules section is displayed.

5. 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.

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 (Name, EDI Code, Application Code) from which you want the contents mapped.

8. From the Map To list, select the field or element to which the information from the Partner Editor is mapped.

9. Click OK to add the standard rule.
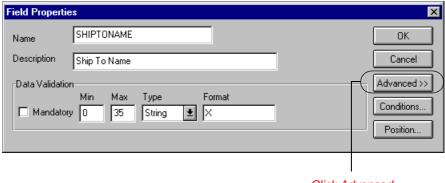
Using Information from Location Tables

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.

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 map information from a Location Table:

1. Double-click the field or element to which the standard rule is applied. 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.

4. Click the Select option. The Standard Rules section is displayed.

5. From the Table box, select the key that is used to look up the Partner Location entry. You is allowed to map from any field 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.

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 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.

8. From the Map To list, select the field or element to which the information from the Partner Editor is mapped.

9. 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.

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 map information from a Lookup Table:

1. Double-click the field or element to which the standard rule is applied. 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.

4. Click the Select option. The Standard Rules section is displayed.

5. From the Table box, select the key from which the system looks up 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 box 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 box on the Lookup Entry dialog of the Partner Editor.

6. In the Sub Table box, type the name of the lookup table.

7. If you want the system to generate an error if the lookup fails, click the Exclusive check box.

8. From the Map From list, select the field (Item, Description, Text 1, Text 2, Text 3, Text 4) from which the contents are mapped.

9. From the Map To list, select the field or element to which the information from the Partner Editor is mapped.

10. Click OK to add the standard rule.
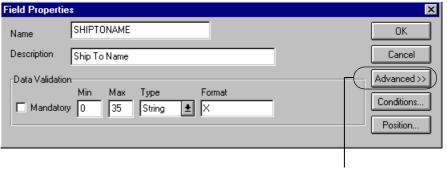
Using Information from Cross-Reference Tables

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.

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 map information from a Cross-Reference (XREF) Table:

1. Double-click the field or element to which the standard rule is applied. 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.

4. Click the Select option. The Standard Rules section is displayed.

5. 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 box 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 box 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 box 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 box on the CrossRef Entry dialog of the Partner Editor.

6. In the Sub Table box, type the name of the cross-reference table.

7. If you want the system to generate an error if the look up fails, click the Exclusive check box.

8. 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.

9. From the Map To list, select the field or element to which the information from the Partner Editor is mapped.

10. 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 fields. This function enables you to calculate sums, hash totals, value totals, counts, and generate HL level and parent codes, and other mathematical operations.
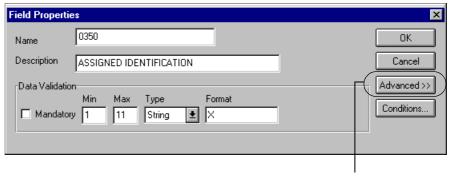
This section addresses three specific mathematical operations that you can perform with accumulators by illustrating an example of when you would use each.

Counting Line Items

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

Complete the following steps to count line items and generate a control total for an outbound ANSI X12 purchase order:
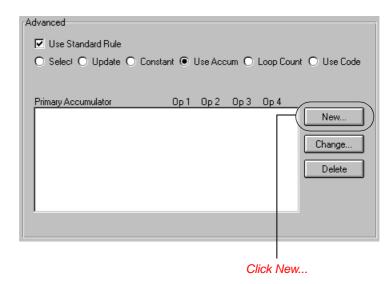
1. Double-click the P0101 element (in the P01 segment in the P01 group). This is the field that you typically use to count the line items. The Field Properties dialog is displayed.

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.
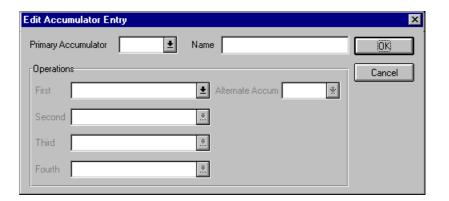
4.   Click the Use Accum option.

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

6. 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 map. This means that accumulator "0," whether it is used in the Primary Accumulator or Alternate Accum box is the same accumulator with the same contents. If you assign calculations to accumulator "0" at the beginning of the map and then use accumulator "0" again later in the map, the content of that accumulator is the result of the earlier calculation. Any additional calculations you assign to that accumulator are performed on the contents resulting from an earlier calculation.**

7. 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.

8. 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 this box are defined as follows:

   - **Increment primary** adds "1" (one) to the contents of the Primary Accumulator (i.e., Primary = Primary + 1).

   - **Decrement primary** subtracts "1" (one) from the contents of the Primary Accumulator (i.e., Primary = Primary - 1).

   - **Sum in primary** adds the numeric value (takes the positive or negative sign of the numbers into account) of the field to the contents of the Primary Accumulator (i.e., Primary = (+/-)Primary + (+/-)Field).

   - **Hash sum in primary** adds the absolute value (does not take the positive or negative sign of the numbers into account) of the field to the contents of the Primary Accumulator (i.e., Primary + Field).

■ **Load primary** loads the contents of the field into the Primary Accumulator (i.e., Primary = Field).

■ **Use primary** loads the contents of the Primary Accumulator into the field (i.e., Field = Primary).

■ **Zero primary** sets the value of the Primary Accumulator to zero (i.e., Primary = 0).

■ **Multiply with primary** multiplies the field with the contents of the Primary Accumulator, and stores the result in the Primary Accumulator (i.e., Primary = Primary * Field).

■ **Divide by primary** divides the field with the contents of the Primary Accumulator, and stores the result in the Primary Accumulator (i.e., Primary = Field / Primary).

■ **Divide primary by field** divides the contents of the Primary Accumulator with the field, and stores the result in the Primary Accumulator (i.e., Primary = Primary / Field).

■ **Modulo with primary** divides the contents of the field with the contents of the Primary Accumulator, and stores the *remainder* of that operation in the Primary Accumulator (i.e., Primary = Primary % Field).

■ **Modulo with field** divides the contents of the Primary Accumulator with the contents of the field, and stores the *remainder* of that operation in the Primary Accumulator (i.e., Primary = Field % Primary).

■ **Negate primary** makes the contents of the Primary Accumulator negative (i.e., Primary = Primary * -1).

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

■ **Move primary to alternate** copies the contents of the Primary Accumulator to the Alternate Accum. This overwrites the current contents of the Alternate Accum field (i.e., Alternate = Primary).

■ **Add primary to alternate** adds the contents of the Primary Accumulator to the contents of the Alternate Accum, and stores the result in the Primary Accumulator (i.e., Primary = Primary + Alternate).
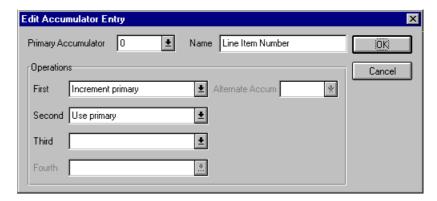
- ■   **Multiply primary by alternate** multiplies the contents of the Primary Accumulator with the contents of the Alternate Accum, and stores the result in the Primary Accumulator (i.e., Primary = Primary * Alternate).

- ■   **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., Primary = Primary / 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., Primary = Primary % Alternate).

9.   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.
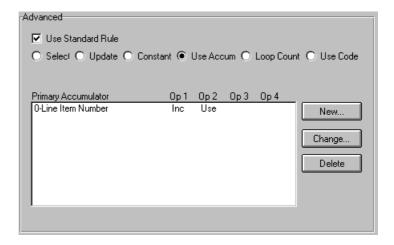
> **The Second box is active only after you select a First operation that does not involve the Alternate Accum. The Third box is active only after you select a Second operation. The Fourth box is active only after you select a Third operation.**

The Edit Accumulator Entry dialog should now look like this:
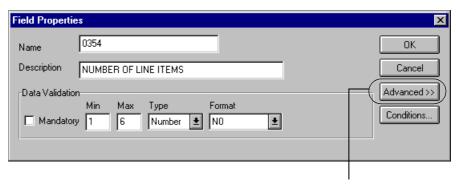
10.   Click OK to add the accumulator.

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

11. Click OK on the Field Properties dialog to add the standard rule to the
    P0101 element.

12. Double-click the CTT01 element (in the CTT segment). This is the field that typically contains the total number of line items. The Field Properties dialog is displayed.

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

14. Click the Use Accum option.

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

16. From the Primary Accumulator list, select primary accumulator "0." This accumulator currently contains the total number of line items.

17. From the First list, select "Use primary." This specifies 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:

18. Click OK to add the accumulator.

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

19. Click OK on the Field Properties dialog to add the standard rule to the CTT01 element.
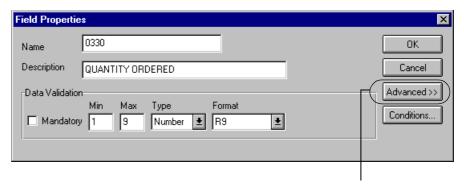
    The CTT01 element now contains the total number of line items in the purchase order.

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) field.

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

1. Double-click the P0102 element (in the P01 segment in the P01 group). This is the field 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.

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 a Primary Accumulator from the list (for this example, select primary accumulator "1"). 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.

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." This operation specifies that the system 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 for each iteration of the P01 group.

    The Edit Accumulator Entry dialog should now look like this:

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 field that typically contains the total quantity of the purchase order. The Field Properties dialog is displayed.
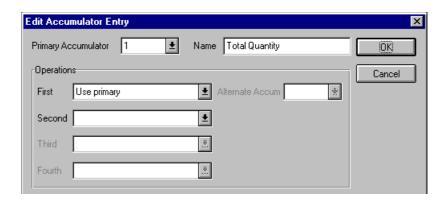
Click Advanced>>

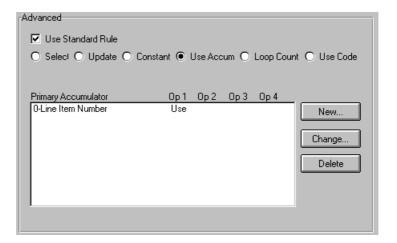11. In the Advanced section, click the Use Standard Rule check box to display the rule options.

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.
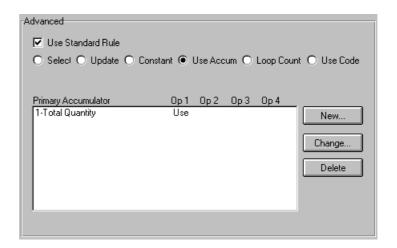
15. From the First list, select "Use primary." This operation specifies that
    the system loads the current value of the accumulator into the CTT02
    (Hash Total) element.

    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.
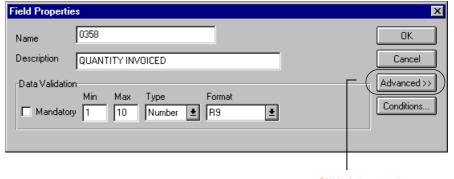
Resetting and Calculating a Value Total

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

Multiplying quantity invoiced by unit price

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

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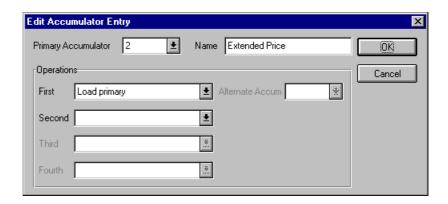
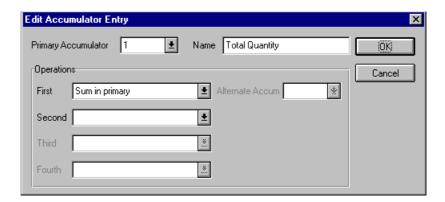*Click Advanced>>*

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

3. Click the Use Accum option.

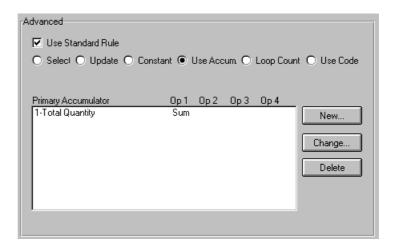4. Click NEW... to access the Edit Accumulator Entry dialog to create a new calculation for this element.

5. Select a Primary Accumulator from the list (for this example, select primary accumulator "2"). This number is a name by which the accumulator is referred. 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.

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." This operation specifies that the system loads the contents of the element into the Primary Accumulator for each iteration of the IT1 group.

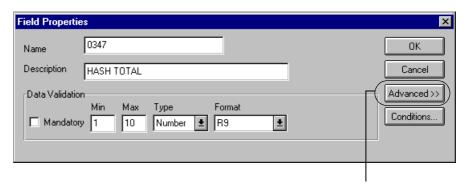   The Edit Accumulator Entry dialog should now look like this:

8. Click OK to add the accumulator.

   The Advanced section of the Field Properties dialog should now look like this:
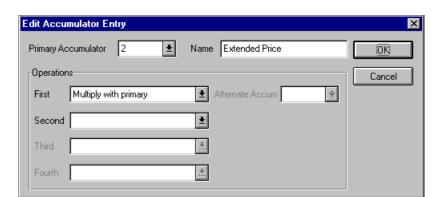
9. Click OK on the Field Properties dialog to add the standard rule to the IT102 element.

10. 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.

Field Properties

Name	0212
Description	UNIT PRICE

Data Validation

	Min	Max	Type	Format
☐ Mandatory	1	14	Number ▾	R9 ▾

OK
Cancel
Advanced >>
Conditions...

Click Advanced>>
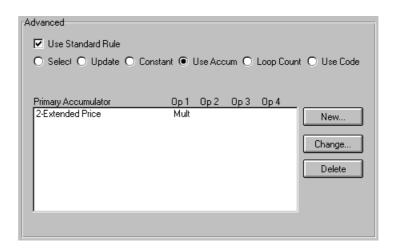
11. If the Advanced section of the dialog is not currently displayed, click ADVANCED>> to access this section, which enables you to define rules.

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 list, select primary accumulator "2."

16. From the First list, select "Multiply with primary." This operation specifies that the system multiplies the value of the IT104 (Unit Price) element with the contents of the primary accumulator, and store the result in the primary accumulator for each iteration of the IT1 group.

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:

18. Click OK on the Field Properties dialog to add the standard rule to the IT104 element.
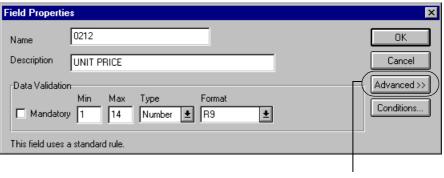
> **If there is an extended price element in your EDI file, you could load the total from the extended price calculation into that element. To do this, you need to use an accumulator on that extended price element that specifies "Use primary" for accumulator 2.**

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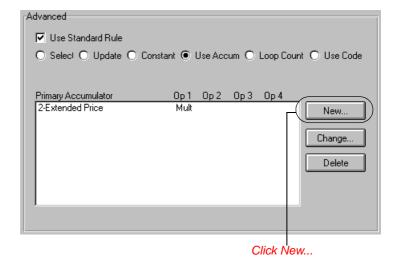
Complete the following steps to generate a running total of the extended price:

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

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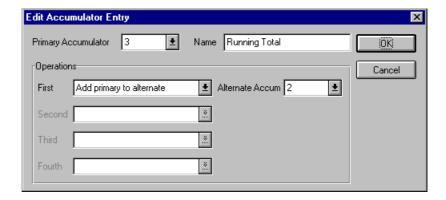
2. In the Advanced section, you already established one accumulator that is displayed in the list.

Click New...
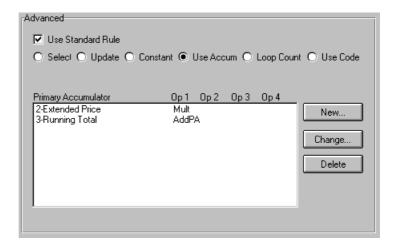
3.   Click NEW... to access the Edit Accumulator Entry dialog to create another calculation for this element.

4.   From the Primary Accumulator list, select primary accumulator "3."

5.   In the Name box, type "Running Total."

6.   From the First list, select "Add primary to alternate." This operation specifies that the system adds the contents of the primary accumulator to the contents of the alternate accumulator, and stores the result in the primary accumulator for each iteration of the IT1 group.

7.   From the Alternate Accum list, select "2." This operation specifies that the system add the value of accumulator "2" (which contains the extended price for a line item) to the value of accumulator "3." The system stores the sum in accumulator "3," which therefore contains a running total of the extended price with each iteration of the IT1 group.

The Edit Accumulator Entry dialog should now look like this:

8.   Click OK to add the accumulator.

The Advanced section of the Field Properties dialog should now look like this:
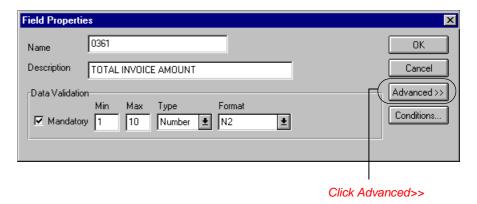
9. Click OK on the Field Properties dialog to add the standard rule to the IT104 element.

Loading a running total of extended price

Complete the following steps to load the running total of the extended price into the TDS01 (Total Invoice Amount) element:
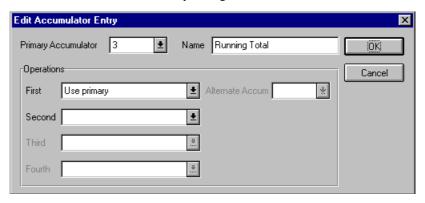
1. Double-click the TDS01 element (in the TDS segment in the IT1 group). This is the field that contains the unit price for each line item. The Field Properties dialog is displayed.

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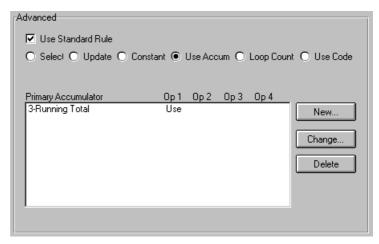
4.   Click NEW... to access the Edit Accumulator Entry dialog to create a
     new calculation for this element.

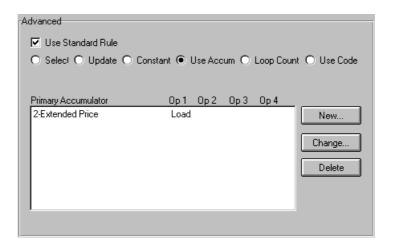5.   From the Primary Accumulator list, select primary accumulator "3."

6.   From the First list, select "Use primary." This operation specifies that
     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:
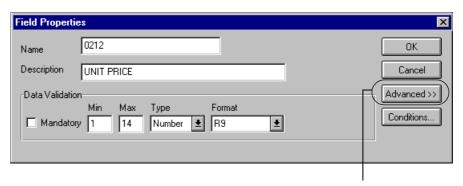

7.   Click OK to add the accumulator.

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

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

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 a code table, you can use a "Use Code" standard rule to either look up a value from a code table or validate the contents of a field against the values in the code table.

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 generates an error if a validation against the code table fails. You can import and export code lists and copy and paste code lists between maps.

A field 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*.
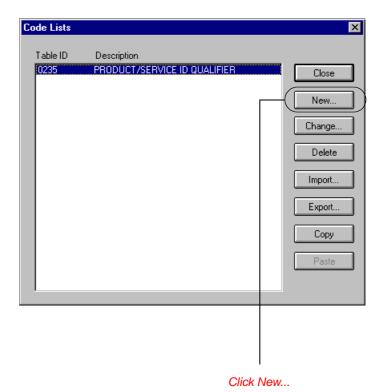
You can perform the following functions on Code List Tables:
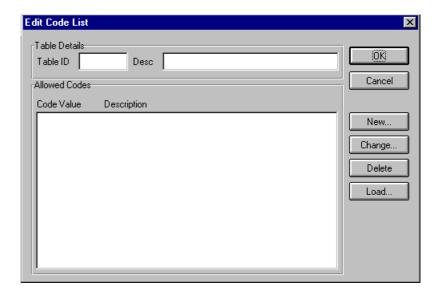
- ■ Defining a Code List
- ■ Modifying a Code List or Code List Entry
- ■ Deleting a Code List or Code List Entry
- ■ Importing a Code List
- ■ Exporting a Code List
- ■ Loading Code Tables from the Standard
- ■ Copying and Pasting Code Lists
- ■ Validating Data Against Code List Tables
- ■ Mapping Code 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...

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

3. In the Table ID box, type the name of the field or element for which this code list table is used.

4. In the Description box, type the description of the field or element for which this code list table is used.

5. Click NEW... to access the Edit Code List Entry dialog.
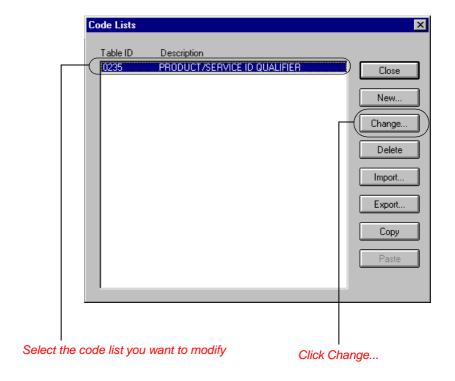
6. In the Value box, type the actual value of the code.

7. In the Description box, type a description of the code value. The code value description is used if you specify an element or field (in the Store Fields list on the Field Properties dialog) to which you want the code description mapped.

8. Click OK to save the code list entry.

9. Repeat steps 5 through 8 to add more code list entries to the code list table.

10. Click CLOSE to save and exit the Edit Code List dialog.

11. Click CLOSE to exit the Code Lists dialog.
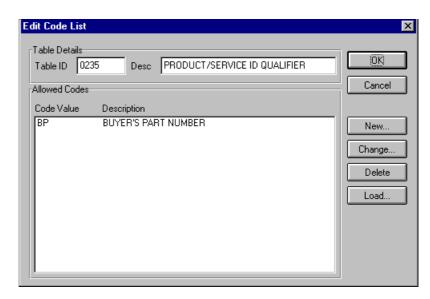
Modifying a
Code List

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.

Select the code list you want to modify   *Click Change...*

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

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

4.   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.
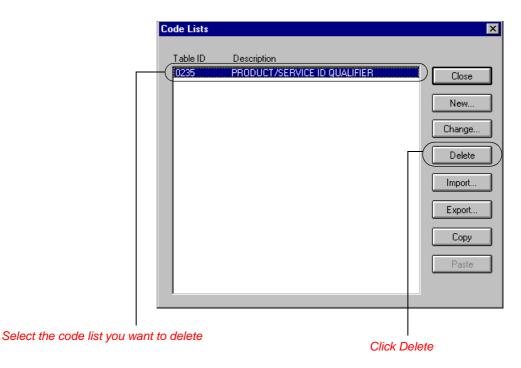
5.   Make the necessary changes.

6.   Click OK to save the code list entry.

7.   Repeat steps 2 through 6 to modify other code list entries in the code list table.

8.   Click CLOSE to save and exit the Edit Code List dialog.

9.   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
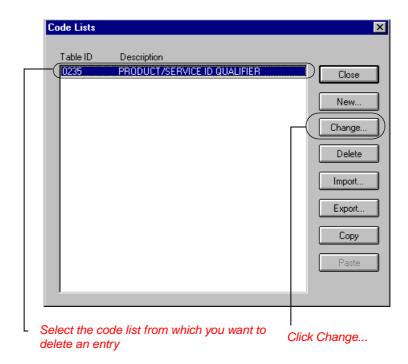
2. If you want to delete a code list, highlight that list.

3. Click DELETE to delete the code list table.

STOP   **The selected table is 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.

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

STOP   **The selected entry is deleted without warning.**
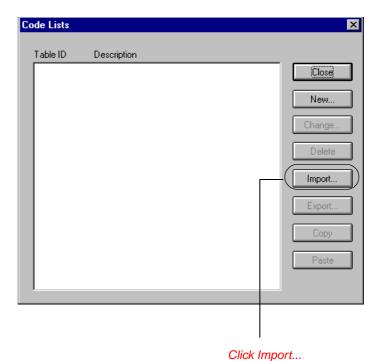
5.   Click OK to save the code list table.

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).

     The default file extension for code lists is ".CDE."

4.   Select the code list file from the list and click OPEN to import the code
     list and return to the Code Lists dialog. The imported code list is
     available for your use.

5.   Click OK to exit the Code Lists dialog.
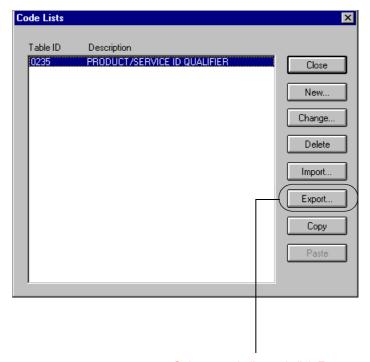
Exporting a Code List

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 folder.
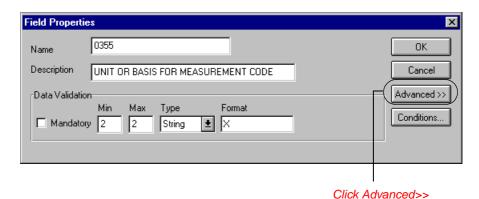
> **The default file extension for code lists is ".CDE."**

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

Complete the following steps to load a code list table from the standard:
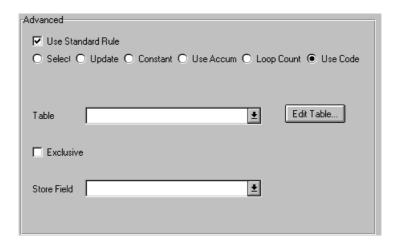
1. Double-click the element for which you need to use a code table. The Field Properties dialog is displayed.
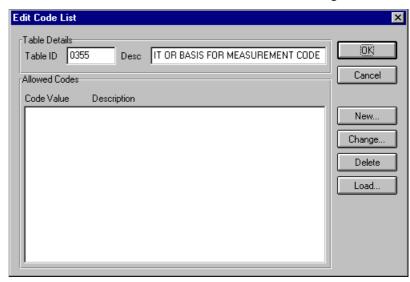
Click Advanced>>

> **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.**

3. Click the Use Standard Rule check box.

4. Click the Use Code option.
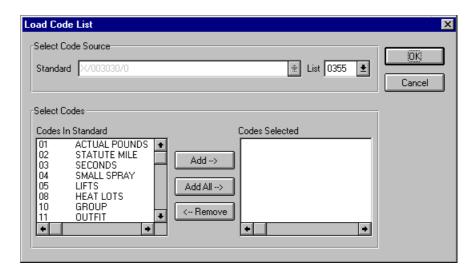
5. Click EDIT TABLE... to access the Edit Code List dialog.

The Table ID box contains the name of the field or element for which this code list table is used. The Description box contains the description of the field or element for which this code list table is used.

6. Click LOAD... and you are prompted to select a data source name (DSN) from which to access the EDI standard. Select the appropriate DSN and click OK to access the Load Code List dialog.

7. 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.

STOP   **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.**
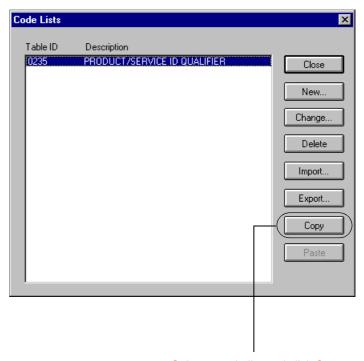
8. Click OK to load the code list.

Copying and Pasting Code Lists

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

Complete the following steps to copy and paste a code list table:
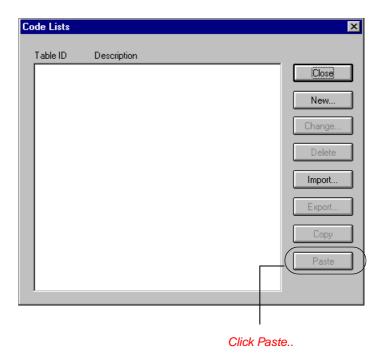
1. Select Code Lists from the Edit menu or the Main Toolbar. The Code Lists dialog is displayed.

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..
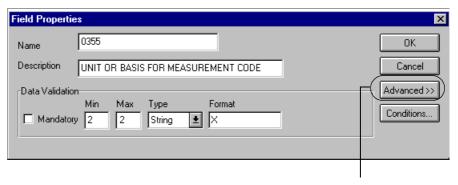
6.   Click PASTE... to add the copied code list to this map.

7.   Click CLOSE to exit the Code Lists dialog.

Validating Data Against Code List Tables

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.
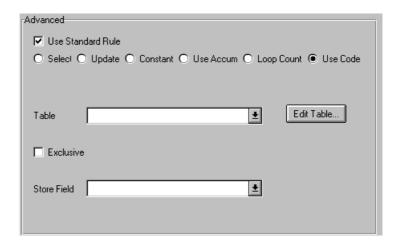
> 🛈 **Please note that the standards provide code list tables only for elements that use them. For example, in the TD4 segment, the TD401 (Special Handling Code) and TD403 (Hazardous Material Class Code) have code tables provided by the standard.**

3. In the Advanced section, click the Use Standard Rule check box.

4. Click the Use Code option.

5.   From the Table list, select the code list table that the data in this field 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 204 for more information.
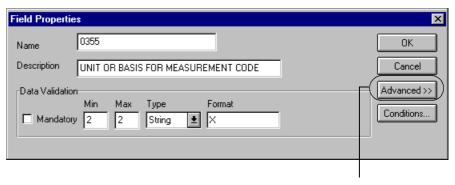
6.   If you need to specify (for compliance reasons) that the field/element *must* contain one of the codes from the specified table (nothing else is valid for that field), click the Exclusive check box.

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

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

Mapping Code Item Descriptions

Complete the following steps to map code item descriptions:

1. Double-click the element for which you need to map a code item description. The Field Properties dialog is displayed.
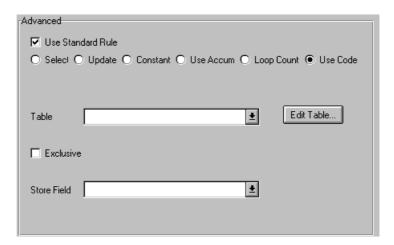
> **i** **Please note that the standards provide code list tables only for elements that use them. For example, in the TD4 segment, the TD401 (Special Handling Code) and TD403 (Hazardous Material Class Code) have code tables provided by the standard.**

3. In the Advanced section, click the Use Standard Rule check box.

4. Click the Use Code option.

5.   From the Table list, select the code list table that the data in this field 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 204 for more information.

6.   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.

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

7.   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."

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

5

Setting Up the Import Process

The import process is part of the outbound translation process. In the outbound translation process, the system translates your application file format to EDI standard formats, so you can send documents to your partners. To translate outbound data, you need to create an import translation object and a system import translation object in the GENTRAN:Director Application Integration subsystem. The import translation object defines how to move data from your application file (flat file definition), which may contain multiple documents, to the EDI standard-formatted documents that your partners expect to receive from you. You need to create a system import translation object to determine which trading relationship (established in Partner Editor) corresponds to each document in the application file, so the system knows which import translation object to use to process the document.

This chapter explains how to set up the import process, including the following topics:

■   Importing data
■   Modifying system configuration
■   Creating a system import map

> **Press F1 from any area of the system to access the online Help for that function. The online Help system contains dialog box element definitions and other detailed information.**

Importing Data

During the import process, GENTRAN:Director imports a specified application file into the Workspace. An import file can contain multiple flat file definitions (documents) that need to be sent to multiple trading partners.
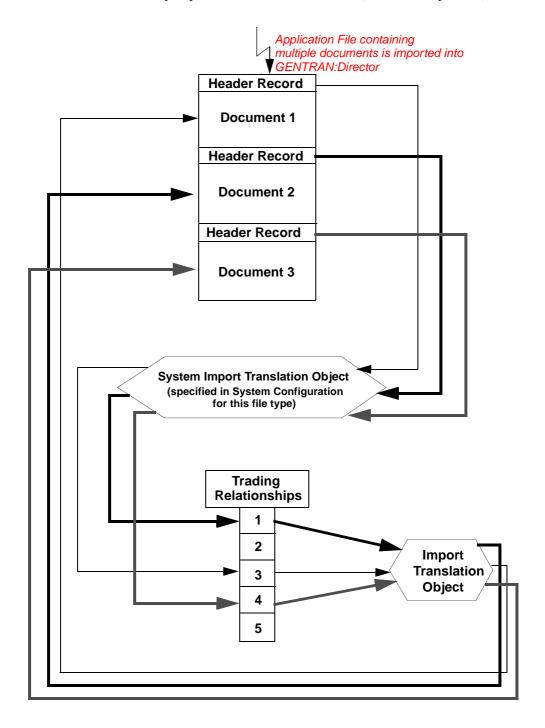
You can import any type of flat file, including TDF (Transaction Data File). TDF is a flat file representation of the EDI standard document. The flat file characteristics can include multiple record types (more than one record definition) and records delimited by carriage returns/line feeds.

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

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

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

The import process is illustrated as follows (translator not pictured):

*Application File containing
multiple documents is imported into
GENTRAN:Director*

Header Record

Document 1

Header Record

Document 2

Header Record

Document 3

System Import Translation Object
(specified in System Configuration
for this file type)

**Trading
Relationships**

1

2

3

4

5

**Import
Translation
Object**

Modifying System Configuration

When you import a file, the system checks the Imports tab of the System Configuration program (shipped with GENTRAN:Director) to verify that the file is matched with a registered system import translation object. Your system administrator may need to add the match to the System Configuration program. Please see *Modifying Import Parameters* in the *Administration Guide* for more information on defining or modifying import specifications.

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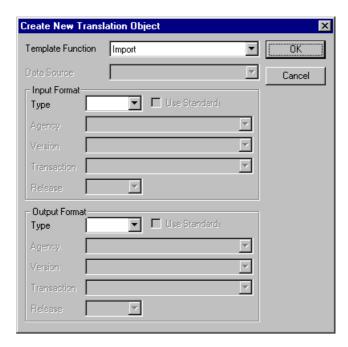
A system import map is used by the system to find the partner relationship for a document (flat file definition), to determine which import map is used to translate the data. The system import map builds the key that the translator uses to find the partner relationship. The sole function of the system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

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

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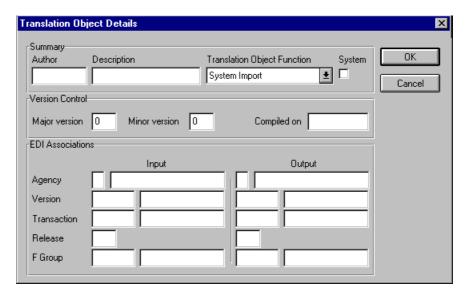
Complete the following steps to create a system import map:

1. Select New from the File menu to create a new map. The Create New Translation Object dialog is displayed.
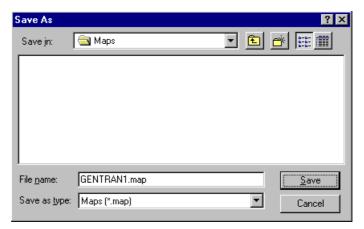
2. From the Translation Object Function list, select "System Import."

3. From the Input Format Type list, select "Positional."

4. From the Output Format Type list, select "Positional." The system actually ignores the Output side of the map and only process the Input side.

5. Click OK to create the new system import map (this may take a few seconds). The new map is displayed in the GENTRAN:Director - Application Integration Window.

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

7. Complete the Author box. This box typically contains the name of the person who created the map.

8. Complete the Description box. For system import maps, we recommend that you type "System Import *," where "*" is typically the unique type of document for which this system import map is used. The Import tab of the System Configuration program describes which system import translation object is used to begin translation during the import process. Please see the *Administration Guide* for more information about how your system administrator needs to modify the system configuration program.

9. The Translation Object Function list contains "System Import." The only reason you change this is if you determine at this point that you really want to create another type of map.

10. 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.

11. Click OK to save the translation object details and access the Save As dialog.

12. We recommend that you store maps in the map folder where GENTRAN:Director is installed.

13. Type the name of the map (1-8 characters) in the "File name" box. The default file extension is ".MAP."

14. Click SAVE to save the map.

    After creating the system import map, you need to define the header record for the system import map. You can use either a six-field key (this is the preferred method) or the alternate key method.

Defining the Six-Field Key

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

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

■ Please see *Creating the First Record* on page - 25 for more information on how to define the header record.

■ Please see *Creating Fields* on page - 32 for more information on how to define the fields in the record.

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

Defining the Partner Key

Complete the following steps to define the partner key:

1. Double-click the partner key field to display the Field Properties dialog.

Click Advanced>>

2. Click ADVANCED>> to access the Advanced 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.

5.   From the Table list, select "Partner by EDI Code," "Partner by Alternate code," or "Partner by Partner Key." This indicates that the system updates this field with the indicated partner key.

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

6.   Click OK to add the standard rule to the partner key field.

Defining the Standard Field
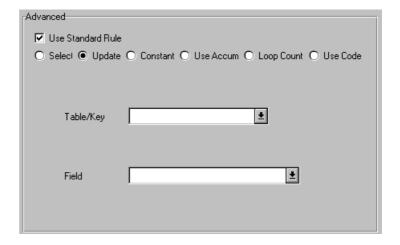
Complete the following steps to define the EDI standard field:

1. Double-click the standard field to display the Field Properties dialog.
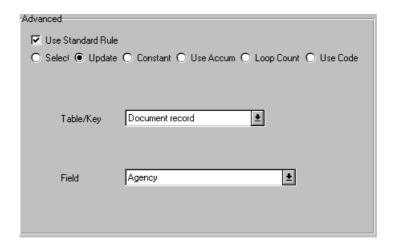
2. In the Advanced section, click the Use Standard Rule check box to display the rule options.

3. Click the Update option.

4. From the Table/Key list, choose "Document record." This indicates that you are updating the document record in the GENTRAN:Director internal system buffer (because the translator is not mapping data at this point in the process).

5. From the Field list, choose "Agency." This indicates that you are updating the agency (standard) field in the system buffer.

The Advanced section of the Field Properties dialog should now look like this:
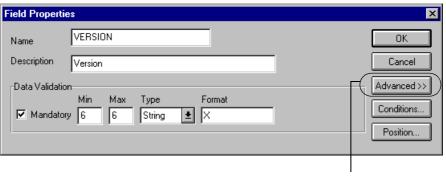
6.   Click OK to add the standard rule to the standard field.

Defining the Version Field
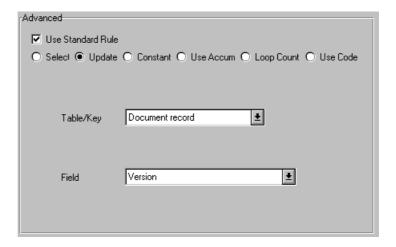
Complete the following steps to define the version:

1.   Double-click the version field to display the Field Properties dialog.

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 the GENTRAN:Director internal system buffer (because the translator is not mapping data at this point in the process).

5.   From the Field list, choose "Version." This indicates that you are updating the standard version field in the system buffer.

The Advanced section of the Field Properties dialog should now look like this:
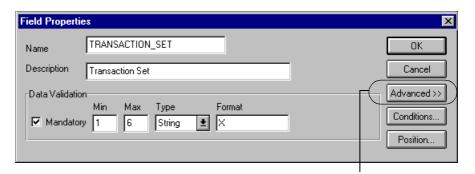
6. Click OK to add the standard rule to the version field.

Defining the Transaction Field

Complete the following steps to define the transaction set (document):
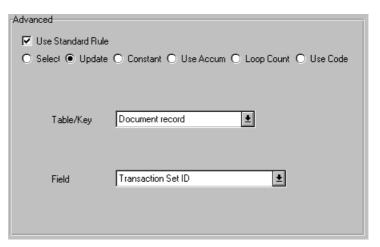
1. Double-click the transaction set field to display the Field Properties dialog.

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 the GENTRAN:Director
    internal system buffer (because the translator is not mapping data at this
    point in the process).

5. From the Field list, choose "Transaction Set ID." This indicates that you
    are updating the transaction set field in the system buffer.

    The Advanced section of the Field Properties dialog should now look
    like this:
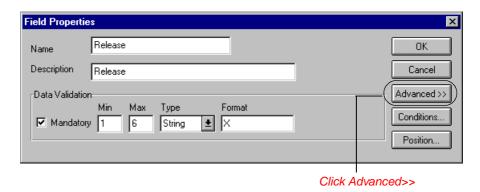
6. Click OK to add the standard rule to the transaction set field.

Defining the   Complete the following steps to define the release of the selected standard
Release Field   version (TRADACOMS *only*):
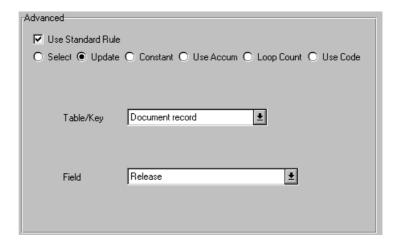
1. Double-click the transaction set field to display the Field Properties dia-
    log.

    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 the GENTRAN:Director internal system buffer (because the translator is not mapping data at this point in the process).

5. From the Field list, choose "Release." This indicates that you are updating the release field in the system buffer.

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

6. Click OK to add the standard rule to the release field.

Defining the Test/Production Field
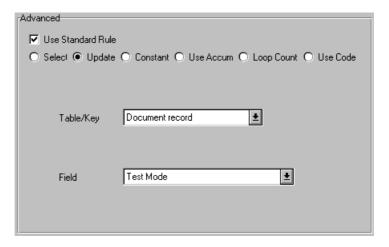
Complete the following steps to define the test or production status field:

1. Double-click the test or production field to display the Field Properties dialog.

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 the GENTRAN:Director internal system buffer (because the translator is not mapping data at this point in the process).

5. From the Field list, choose "Test Mode." This indicates that you are updating the release field in the system buffer.

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

6. Click OK to add the standard rule to the test or production status field.

Defining the Alternate Key

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

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

If you want to also define the other fields that the corresponding import map header record contains, you can do so at this time.
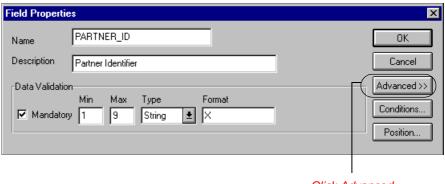
- Please see *Creating the First Record* on page 25 for more information on how to define the header record.

- Please see *Creating Fields* on page 32 for more information on how to define the fields in the record.

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

Defining the Partner Key

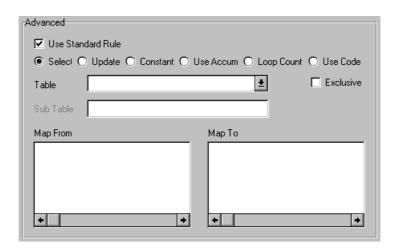Complete the following steps to define the partner key:

1. Double-click the partner key field to display the Field Properties dialog.
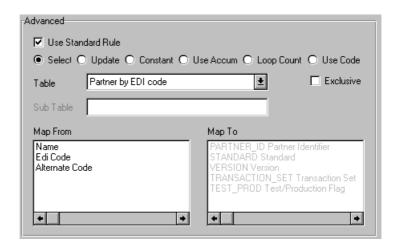

Click Advanced>>

2. Click ADVANCED>> to access the Advanced 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.

5.   From the Table list, select "Partner by EDI Code," "Partner by Alternate
     code," or "Partner by Partner Key." This indicates that the system
     updates this field with the indicated partner key.

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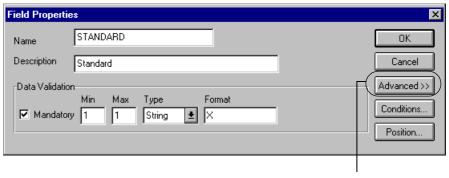   like this:

6.   Click OK to add the standard rule to the partner key field.

Defining the Application ID Field

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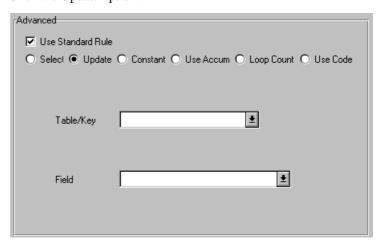
ℹ️ **You must define either the application ID field or the Application Alias Value field (three-field key). You may define both of these fields if you wish to use the alternate key method.**

1. Double-click the standard field to display the Field Properties dialog.

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 the GENTRAN:Director internal system buffer (because the translator is not mapping data at this point in the process).

5. From the Field list, choose "Application Field 1." This indicates that you are updating an optional key field in the system buffer.

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

6. Click OK to add the standard rule to the application ID field.
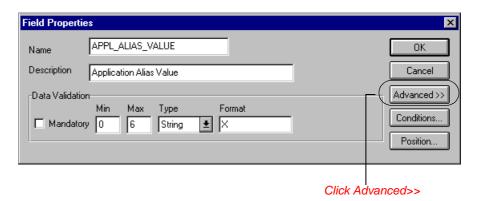
Defining the Application Alias Value Field

Complete the following steps to define the application alias value field:

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

1. Double-click the standard field to display the Field Properties dialog.

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 the GENTRAN:Director internal system buffer (because the translator is not mapping data at this point in the process).

5. From the Field list, choose "Application Field 2." This indicates that you are updating an optional key field in the system buffer.

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

6. Click OK to add the standard rule to the application alias value field.

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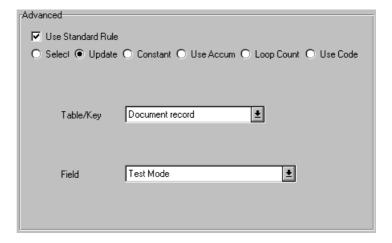
If you do not define the test/production status field, the document record is set to "production" by default. In this situation, the system translation object can only locate the correct relationship if the test mode of the partner relationship is actually set to "production." We recommend that you define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

Complete the following steps to define the test/production status field:

1. Double-click the transaction set field to display the Field Properties dialog.

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 the GENTRAN:Director internal system buffer (because the translator is not mapping data at this point in the process).

5. From the Field list, choose "Test Mode." This indicates that you are updating the release field in the system buffer.

   The Advanced section of the Field Properties dialog should now look like this:
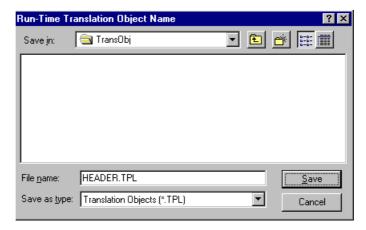
6. Click OK to add the standard rule to the test or production status field.

Compiling the System Import Translation Object

After you create, define, and save the system import map, you need to compile the map and create a system import translation object. The map that you create using GENTRAN:Director is a *source map*. When that source map is compiled, the result is a *compiled translation object*.

Complete the following steps to generate the system import translation object:

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

2. Type the name of the translation object, if necessary. The "File name" box automatically contains the 1-8 characters you used to name the source map. Use the default .TPL file extension.

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

3. From the "Save in" list, change the Drive and Folder where the compiled translation object is stored, if necessary.

4. Click SAVE and the system compiles the map and create a the system import translation object.

5. The translation object must 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.

6. If your system administrator has not already added this system import translation object to the System Configuration program, do so now. Please see *Modifying Import Parameters* in the *Administration Guide* for more information on how to modify import parameters.

Setting Up the Export Process

To translate inbound data, you need to create an export map in
GENTRAN:Director (because the system is *exporting to* your application
file). The export map defines *how* to move data from the EDI standard-
formatted documents that your partners send you to your application file
(flat file definition). This map also enables you to populate your application
file fields with supplementary information from the interchange, group, and
transaction set envelopes.

This chapter explains how to set up the export process, including the
following topics:

■   Inbound process prior to exporting data

■   Using supplementary envelope information

> **Press F1 from any area of the system to access the
> online Help for that function. The online Help system
> contains dialog box element definitions and other
> detailed information.**

Inbound Process Prior to Exporting Data

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

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

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

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

To set up the export process, complete the following steps:

1. **Create an export map**
    Please refer to *Creating a New Map* on page 18 for more information on how to create a new map.

2. **Register the export map with GENTRAN:Director**
    Please refer to the GENTRAN:Director *User's Guide* or online Help for more information on registering a translation object.

3. **Create the appropriate inbound trading relationship**
    Please refer to the GENTRAN:Director *User's Guide* or online Help for more information on creating an inbound relationship.
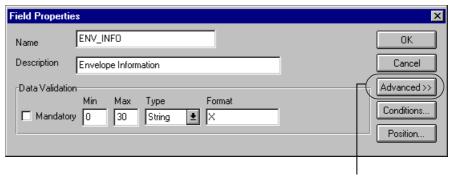
Using Supplementary Envelope Information

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

Complete the following steps to use supplementary envelope information in your application file:

1. Double-click an existing application field, or create a new field (please see *Using a Fixed Format File* for more information on creating fields).

    The Field Properties dialog is displayed.

2. Click ADVANCED**>>** to access the Advanced 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.

5. From the Table list, select "Generic envelope segment." This enables you to map from any of the fields within the current envelope information, regardless of the EDI standard used.

6. Click the Exclusive check box to indicate that a compliance error should be generated if the select does not find a valid entry.

7. From the Map From list, choose the envelope information that you want to map to your application field, based on the following table:

Map From Selection	Information You Want Mapped
Field 1	Transaction Set ID
Field 2	Document Name
Field 3	Group ID
Field 5	Partner Name
Field 6	Interchange Control Number
Field 7	Group Control Number
Field 8	Document Control Number
Field 9	Received Version
Field 15	Received Agency
Field 16	Used Agency
Field 17	Used Version
Field 22	ISA Test Mode
Field 23	Document Test Mode

STOP **Please be certain that you choose the correct Map From selection. Any selection that is not listed in the above table is currently reserved and *should not be selected under any circumstances*.**

8. From the Map To list, choose the field to which you want to map the contents of the Map From field. Each field from the output side of the map is displayed in this list.

9. If you want to map additional envelope information, repeated steps 7 and 8 as many times as necessary. A total of eight fields can be mapped using one Select rule.

10. Click OK to add the standard rule to the application field.

7

Using Extended Rules

Extended rules enable you to use a GENTRAN:Director proprietary programming language to perform virtually any mapping operation you require. You can use these rules to define more complex translations than are available through the link function (simple mapping) and 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

> **Press F1 from any area of the system to access the online Help for that function. The online Help system contains dialog box element definitions and other detailed information.**

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 input or output specification of the map, that you use in a rule must be declared before you use those variables. Therefore, an extended rule consists of two sections, a *declarations* section followed by a *statements* section. The declarations section is only required if you use additional variables. This is where you declare the names and types of any variables you use either in this rule or in any other rule that is within the scope of this rule. The statements section is where you define the actions that you want the rule to execute.

Declarations 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 colon and underscore characters. The first character in a variables 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 link 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 253*.

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.

You can attach extended rules to three different levels of map components:
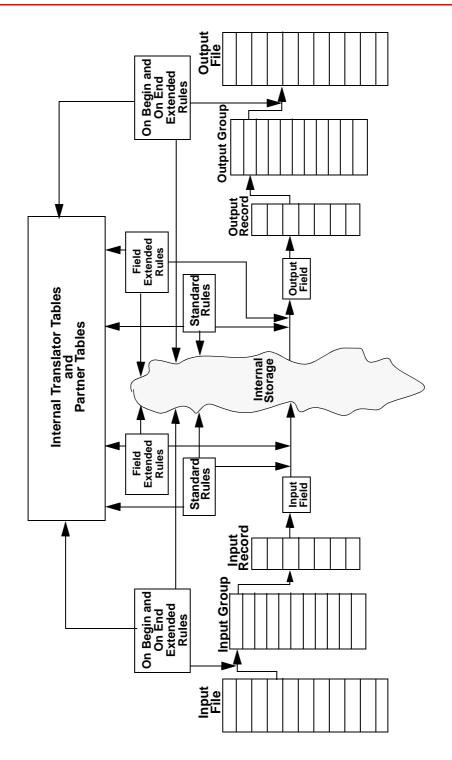
■ EDI file and application file groups, using the Loop Level Extended Rules dialog.

■ Groups, sub-groups, repeating records, and repeating segments, using the Loop Level Extended Rule dialog.

■ Fields and elements, using the Field Level Extended Rule dialog.

Scope The *scope* of an extended rule determines which variables are accessible from within a given extended rule. The scope varies depending on the current state of the map. 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 Begin extended rules (defined on the Loop Level Extended Rules dialog) are in scope until the conclusion of its companion On End rule (also defined on the Loop Level Extended Rules dialog).

■ Field level extended rules are only in scope for the duration of the field or element.

An extended rule attached to the current map component depends on the type and state of the map component. 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 255. An extended rule that is attached to a field or element is only in scope for the duration of the field or element. Field level extended rules are always processed after standard rules.

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

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

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

Input rule processing   The translator processes the Input side of the map first and then the Output side. The following is the sequence of how the translator processes rules on the Input side of a map:

1. Load the Input definition.
2. Read the Input file.
3. Determine if data is present for the first/next group and then run the group On_Begin rule, if present.
4. Load each field in the group and execute field level rules (for each field) in the following sequence:
    - ■   Run standard rules
    - ■   Run extended rules
5. At the end of the group, execute the On_End rule, if present.
6. Repeat steps 2 - 5 for each group in the Input file.

Output rule processing   The following is the sequence of how the translator processes rules on the Output side of a map:

1. Verify whether or not data exists for the first/next record.
2. If the record is the first record of a group, run On_Begin rule, if present.
3. For each field in the record, execute field level rules (for each field) in the following sequence:
    - ■   Run standard rules
    - ■   Run extended rules
4. Format data according to specified field properties (on Field Properties dialog).
5. Write the record to the Output file.
6. At the end of the group, execute On_End rule, if present.
7. Repeat steps 1 - 6 for each record in the Output file.
8. Create or update the document entry in the database.

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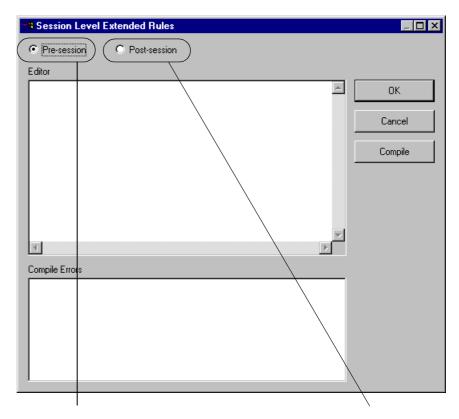
The map component that an extended rule accesses depends on what you want the scope of the rule to be. You also need to determine when you want the rule to be execute (e.g., before or after the map component is processed).

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

Defining a Session Rule

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.
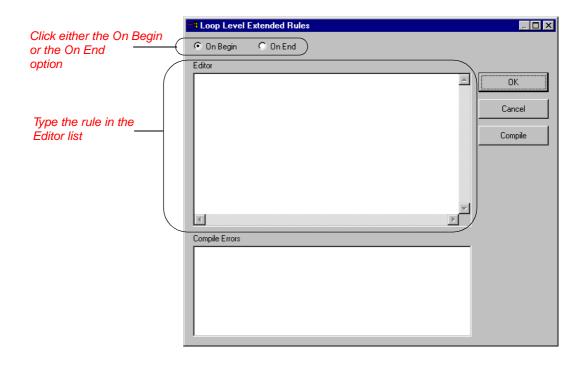
    This function gives you immediate feedback about the accuracy of your rule. Regardless, the rule is compiled when you compile the entire translation object.
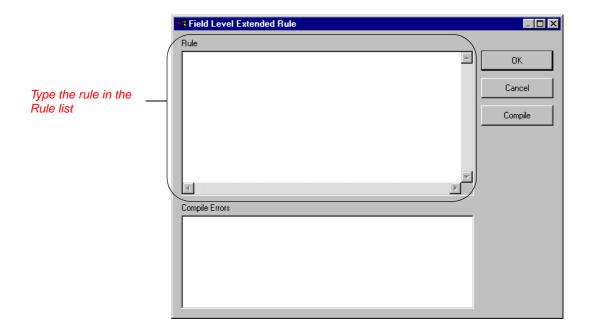
5. Correct any errors that the system flagged and click COMPILE again. Repeat this process until there are no errors generated.

6. Click OK to add the extended rule.

Defining a Map Component Rule

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

1. Click the map component with the right mouse button to access the shortcut menu. Select Rules... (or Extended Rule... if the map component is a field or element) from the shortcut menu.

    ■   If the map component is an EDI file, application file, group, sub-group, repeating record, or repeating segment, the Loop Level Extended Rules dialog is displayed.

    ■   If the map component is a field or element, the Field Level Extended Rule dialog is displayed.

Click either the On Begin or the On End option

Type the rule in the Editor list

Type the rule in the Rule list

2.   If you want the extended rule to be executed before the system
     processes the map component, click the On Begin option. If you want
     the rule to be executed when the system concludes its processing of that
     map component, click the On End option. You can define both an On
     Begin and an On End rule for the a single map component.

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. 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

■   CERROR

■   EMPTY

■   EXIST

■   INDEX

■   NTOA

■   COUNT

■   DELETE

■   FSEEK

■   FTELL

■   READBLOCK

■   LEFT

■   MID

■   RIGHT

■   SELECT

■   UPDATE

■   WRITEBLOCK

i   **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 GENTRAN:Director extended rules are variables, constants, map components/internal storage, arrays, and accumulators. You can address existing map components and you also have the ability to create additional instances of map components, as long as the map component is originally defined in internal storage. 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 mapping action, the syntax is #FIELD_NAME
#field_1 = 2;
where 2 is a numeric constant value
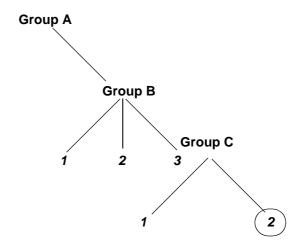
■ To address a field within a group or create a field within a group in internal storage within the scope of the current hierarchy, the syntax is $GROUP.#FIELD_NAME
$N1.#0234

■   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]
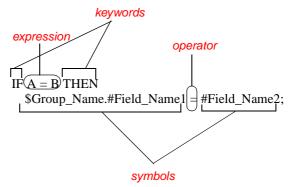
■ An accumulator can be accessed in the same manner as variables or internal storage. To address an accumulator, use the syntax accum(n) where "n" is the number (not the name) of the accumulator: **accum(2) = 5;**

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 numbers, 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 variable or field and returns the result as a string. The right function extracts a specified number of character from the right of the string variable and returns the result as a string. The mid 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* 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:

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. If this function does not find the specified substring inside the string, it returns a value of "-1." 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 a follows:

integer a;
real b;
string[8] s;

a = atoi("5");
b = aton("5.5");
ntoa(5.5, s);

Raise Compliance Error Function (CERROR)

The *cerror* function raises a compliance error. The syntax is as follows:

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

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

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 uses 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]);

File Pointer Functions (FSEEK, FTELL)

The *fseek* function moves the file pointer to a new location which is a specified number of bytes (offset) from the beginning of the file (origin). To invoke the fseek function against the input file, the value for current_file is "0." To invoke the fseek function against the output file, the value for current_file is "1." The fseek function is typically only used in conjunction with the ftell, readblock, and writeblock functions. The syntax is as follows:

fseek(current_file,offset,origin);

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

numeric_variable = ftell(current_file);

Block of Data Functions (READBLOCK, WRITEBLOCK)

The *readblock* function reads a block of data (segment or record) from the input file and places it into the argument of a string variable. The *writeblock* function writes the data contains in the argument of a string variable to the output file. The readblock and writeblock functions are used in conjunction with each other to pass a block of data from the input file to the output file without compliance checking or testing for proper EDI syntax. Together these functions provide a more efficient alternative of using "wildcard" segments, which are typically implemented in build and break maps. The syntax is as follows:

readblock(string_variable);

writeblock(string_variable);

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.
condition	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.

■ Map components are shown in all upper case letters.

■ A double slash (//) introduces comments.

■ Lines that are too long to fit on one line in this section may be continue on the next line using a line-continuation character (➡).

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.

cerror

The *cerror* function raises a compliance error. You typically specify this function as an action to be performed if a condition is false.

The syntax is as follows:

cerror(error_number,$GROUP_NAME[index][index][index].
➥#FIELD_NAME);

An example is as follows:

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

//This raises compliance error 100 on the VATC element of the
//specified instance of the ILD group

The compliance error codes are as follows:

Error Number	Error Generated
100	Mandatory Element
105	Conditional Element
110	Incorrect Element Format
115	Incorrect Element Length
120	Too Many Components
125	Element Delimiter
130	Conditional Relation
135	Element Code
140	Implicit Rule Failure
200	Mandatory Component
205	Conditional Component
210	Incorrect Component Format
215	Incorrect Component Length
220	Component Delimiter
225	Component Code
300	Mandatory Segment
305	Segment Sequence
310	Invalid Start End
400	Session Error
405	Unknown Partner
410	Invalid Ctl Number
415	Invalid Sub Count
420	Unknown Relationship
425	Unknown Standard

Error Number	Error Generated
430	Unknown Version
435	Unknown Transaction

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 = " "

fseek

The **fseek** function moves the file pointer to a new location which is a specified number of bytes (offset) from the beginning of the file (origin). To invoke the fseek function against the input file, the value for current_file is "0." To invoke the fseek function against the output file, the value for current_file is "1." The fseek function is typically only used in conjunction with the ftell, readblock, and writeblock functions. The syntax is as follows:

fseek(current_file,offset,origin);

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

numeric_variable = ftell(current_file);

An example is as follows:

string[1024]temp_buffer;

Integer Position;

Position = ftell(0)

```
while readblock(temp_buffer) do
begin
  if left(tem_buffer,3) = "IEA" then
    begin
       fseek(0,Position,begin);
        break;
     end
  writeblock(tem_butffer);
   Position = ftell(0);
end
//Read segment from input file and place in temp_buffer. Look for
//"IEA" segment tag. If found, reset file pointer to where it was before
//the "IEA" segment was read. Write contents of temp_buffer to output
//file. Set "Position" = the current file pointer position.
```

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. 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. 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.

i **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.**

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".

readblock

The **readblock** function reads a block of data (segment or record) from the input file and places it into the argument of a string variable. The readblock and writeblock functions are used in conjunction with each other to pass a block of data from the input file to the output file without compliance checking or testing for proper EDI syntax. Together these functions provide a more efficient alternative of using "wildcard" segments, which are typically implemented in build and break maps.

The syntax is as follows:

readblock(string_variable);

An example is as follows:

```
while readblock(temp_buffer) do
begin
  if left(tem_buffer,3) = "IEA" then
    begin
        fseek(0,Position,begin);
         break;
     end
  writeblock(tem_butffer);
   Position = ftell(0);
end
```

//Read segment from input file and place in temp_buffer. Look for
//"IEA" segment tag. If found, reset file pointer to where it was before
//the "IEA" segment was read. Write contents of temp_buffer to output
//file. Set "Position" = the current file pointer position.

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 ➥Partnerlookup 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)

Format Specifier	Description
%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.

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 execute another program while running the translator. This program is executed asynchronously. You specify the program and determine how you want the program window displayed. You can also return an error code, if desired. If the error code is greater than 32, the program ran without errors. If the error code is less than 32, the program did not run because of an error. If the error code is "0," the system is out of memory. If the error code is "2," you didn't specify a file name. The error code is *not* the return value from the program you executed.

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 (you must use the *number* to indicate how you want the program window displayed, not the window_display value):

Number	Window_Display	Definition
0	SW_HIDE	Hides the window and activates another window.
1	SW_SHOWNORMAL	Activates and displays a window. If the window is minimized or maximized, it is restored to the original size and position. This flag should be specified when displaying a window for the first time.
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).

Number	Window_Display	Definition
8	SW_SHOWNA	Displays the window in its most recent size and position without activating it (the current active window remains active).
9	SW_RESTORE	Activates and displays the window. If the window was minimized or maximized, it is restored to its original size and position.
10	SW_SHOWDEFAULT	Activates the window and allows Windows to determine the size and position.

writeblock

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

The syntax is as follows:

readblock(string_variable);

An example is as follows:

```
while readblock(temp_buffer) do
begin
  if left(tem_buffer,3) = "IEA" then
    begin
        fseek(0,Position,begin);
         break;
      end
  writeblock(tem_butffer);
   Position = ftell(0);
end
```

//Read segment from input file and place in temp_buffer. Look for //"IEA" segment tag. If found, reset file pointer to where it was before //the "IEA" segment was read. Write contents of temp_buffer to output //file. Set "Position" = the current file pointer position.

Select and

<div style="text-align: right">

Update Available Options

</div>

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   The field names that are available when using the select or update extended
Table   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 extended rule 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

8

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 record).

A group must have a unique name

Description: You tried to give a group a name that is already used by another group or record.

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 record. 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 record a name that is already used by another group or record.

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 record 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 field that
                      is already used in an existing constant.

User Action: Type a unique literal constant identifier in the ID field.

**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 field.

**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 character or character code in Record Delimiter 1

Description: You entered an invalid Record Delimiter 1 on the Positional File Format Properties dialog.

You have entered an invalid character or character code in Record Delimiter 2

Description: You entered an invalid Record Delimiter 2 on the Positional File Format Properties dialog.

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 translation object description on the Translation Object Details dialog.
User Action:	Type the translation object description in the Description field on the Translation Object Details dialog. This field must be unique because it is used by the system to identify the map. We recommend that you use the following identifying characteristics: the partner that this map is used for, the standard, the version, the type of transaction this map uses, and the direction of the map (i.e., Import, Export, etc.). For example, "MWT X 3030 850 Export" is the description of a map used with partner MWT, for an ANSI X.12 version 003030 Purchase Order (850), that is sent Inbound (Export).

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 field.

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 field 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 field 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 field 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.

G L O S S A R Y

?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 GENTRAN:Director. 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 map and creates a translation object. The map that you create using GENTRAN:Director is a *source map*. When that source map 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.

EANA   International Article Numbering Association, Brussels, Belgium.

EDI   Electronic Data Interchange (EDI) is the process by which companies can exchange business documents directly from application to application by computer without paper documents being produced.

EDI standard   This is the rules for turning a business document into an EDI document.

EDIA   The Electronic Data Interchange Association, formerly known as the Transportation Data Coordinating Committee (TDCC).

EDIFACT   This is the standards organization of the United Nations Economic and Social Council. The acronym is short for the Electronic Data Interchange for Administration, Commerce, and Transport.

element   The smallest piece of usable information defined by the standards. Examples might include a quantity, unit price, or description. An individual element can have somewhat different meanings depending on context. Therefore, elements are normally not considered to have useful meaning until they are combined into segments. An element is the EDI map component that is mapped (linked) to a corresponding application field to move data to and from the EDI file.

There are three types of data elements, as illustrated in the table below:

Data Element	Definition
Simple data element	A single piece of information defined by the standards.
Composite data element	A data element that is made up of multiple component elements.
Component data element	A simple data element that belongs to a composite data element. Component data elements are also called sub-elements.

envelope 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.

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 is 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.

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 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.

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.

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

literal constant   These are used by the system as a repository to store information that is used at a later point in the map. Typically, constants are used in an outbound map to generate a qualifier.

location table   This is a table created in Partner Editor that is used to contain address-related information about the partner.

lookup table   This is a table created in Partner Editor that is used to select information related to a value in inbound or outbound data.

loop count   This standard rule enables you to count the number of times a loop is repeated, if the element or field is part of a loop. If the loop is a nested loop, you can track the current loop *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 GENTRAN:Director 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.

outbound mapping   The system translates your application file format to EDI standard formats, so you can send documents to your partners. To translate outbound data, you need to create an import map and a system import map in GENTRAN:Director.

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.

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.

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 your trading partner.

record Contains a group of related fields. A record can occur once or can repeat multiple times.

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

respond This GENTRAN: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.

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 your trading partner. Only selected documents are enveloped and sent. If no documents are selected, ALL documents are sent. Successfully sent documents are moved to the Out Drawer.

set See Transaction Set.

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).

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.

<u>**Inbound Translation Objects:**</u>

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.

<u>**Outbound Translation Objects:**</u>

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 GENTRAN:Director, the Application Integration subsystem, and the 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.

STOP	**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.

I N D E X

Symbols

?In Documents *309*
?Out Documents *309*

A

accumulators *143*
    calculating a value total *183*
    calculating hash totals *179*
    calculating sums *171*
    using *171*
acknowledgements *309*
activation *44*, *94*, *309*
AIAG *309*
alphabetic language reference
    atoi *267*
    aton *268*
    begin *268*
    break *269*
    cerror *269*
    concat *271*
    continue *272*
    count *272*
    delete *272*
    do *288*
    else *277*
    empty *273*
    end *274*
    exist *274*
    fseek *275*
    get *276*
    if *277*
    index *278*
    left *278*
    len *279*
    messagebox *279*
    mid *281*
    ntoa *281*
    readblock *282*, *290*
    right *282*
    select *284*
    set *284*
    strdate *285*
    strstr *286*
    then *277*
    update *287*
    while *288*
    winexec *288*
ANA *309*
ANSI *309*
application file *309*
application file format *8*
application system *310*
atoi *267*
aton *268*

B

begin *268*
break *269*
browser *310*
button *310*

C

cerror *269*
check box *310*
click *310*
code list entry
    deleting *197*
code list table *147*
code list tables *191*, *310*
    copying and pasting *207*
    defining *192*
    deleting *197*
    exporting *202*
    importing *200*
    loading *204*
    mapping code item descriptions *211*
    modifying *195*
    validating data against *209*
code table *147*
code tables
    copying and pasting *207*

translator *320*
transmission *320*
    chain *320*
turnaround document *321*
turnaround map *321*
type
    date/time *81*
    number *78*
    string *67*

U
UCS *321*
update *287, 321*
    using *131*
use accum *321*
    using *143*
use code *321*
    using *147*
user translation object *6, 322*
User's Guide
    conventions *4*
using *191*

V
value total
    calculating *183*
version *322*
VICS *322*

W
while *288*
winexec *288*
Workspace *322*

X
X12 *322*