Gentran:Server® for Windows®

XML User's Guide

Version 5.1

Sterling Commerce
An IBM Company

4232-510-USER07-0000

Copyright Notice

Gentran:Server for Windows

Sterling Commerce Software
Trade Secret Notice

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

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

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

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

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

Third Party Software:
Portions of the Sterling Commerce Software may include products, or may be distributed on the same storage media with products, ("Third Party Software") offered by third parties ("Third Party Licensors").

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

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

Sterling Commerce, Inc.
4600 Lakehurst Court Dublin, OH 43016-2000 * 614/793-7000

Table of Contents

Appendix A XML Build and Break Maps

Appendix B XML Tutorial

About This Guide

Contents

Overview

Introduction

Gentran:Server® for Windows® enables you to manage XML documents along with traditional electronic commerce messages in the same manner. This allows you to maintain a homogenous processing and data management environment, regardless of how the incoming and outgoing data is formatted (XML, EDI, proprietary message formats, etc.).

This manual is intended to explain how to use XML (eXtensible Markup Language) with Gentran:Server for Windows and assist you in performing various tasks in Gentran:Server. This manual uses a task-oriented approach, which is intended to answer any questions you may have about Gentran:Server with step-by-step instructions.

Note
This manual is *not* intended to explain or define XML.

Reference
See the most recent XML specification (available from the World Wide Web Consortium—W3C) for more information about the XML language.

`www.w3c.org`

Before you Begin

Assumptions

This list contains the items with which this manual assumes you are familiar.

- Gentran:Server for Windows
- Windows
- XML
- How and when extended rules are invoked and their scope
- How to create extended rules

Prerequisite

This list describes the software prerequisites to use XML with Gentran:Server.

- You must have Gentran:Server for Windows version 5.0 currently installed.
- You must have installed the XML option from the Gentran:Server 5.0 Options Pack CD.
- To use the Data Definition Format (DDF), you must have Internet Explorer 5.0 installed on your machine.

What's in This Manual

Description of contents

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

- *About This Guide* explains the content, organization, and conventions in this guide. This chapter also describes how to get technical support,
- *Chapter 1, Introducing XML with Gentran:Server* provides you with an introduction to how XML is implemented with Gentran:Server for Windows, and an introduction to Data Definition Format (DDF).
- *Chapter 2, Using XML with Gentran:Server* describes tasks you need to complete to begin using XML in Gentran:Server, and how to work with XML map components.
- *Appendix A, XML Build and Break Maps* provides general guidelines on how to create XML build and break maps, which is an advanced Gentran:Server feature.
- *Appendix B, XML Tutorial* contains a tutorial for the Application Integration subsystem, using XML. This tutorial includes an inbound mapping example (invoice) and an outbound mapping example (invoice). The purpose of this tutorial is to present the general mapping process, using examples that teach you a logical approach and methods that should be utilized when you create your own maps.

Getting Support

Introduction

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

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

Phone number

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

Before calling support

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

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

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

Accessing the Sterling Commerce Support Web Site

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

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

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

Reference
Refer to the *Getting Started Guide* for information on how to access the Customer Support Web Site.

Documentation

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

CHAPTER

1

Gentran:Server XML Overview

Contents

Overview

Chapter Overview

Introduction

This chapter provides you with a introduction of how Gentran:Server uses XML, including the new features and functions of Gentran:Server.

Why use XML?

XML (eXtensible Markup Language) provides you with an flexible language to define document content. This enables you to exchange information with your business partners independent of platform or system compatibility.

XML with Gentran:Server Overview

Overview

Gentran:Server enables you to manage and translate documents, regardless of the document format (XML, traditional electronic commerce, your proprietary format, etc.).

Gentran:Server's XML implementation conforms to the rules of the XML language 1.0 specification, as published by the World Wide Web Consortium (except as specified in the *Specifications* section below). In the interest of flexibility, Gentran:Server diverges from the World Wide Web Consortium's base 1.0 XML specification in the following ways:

- You can specify the number of times that a group can repeat.
- You can specify the number of times an element in a mixed group can repeat.
- You can repeat an element (with a different structure than the original element) in a different part of the document (e.g., you can define an "address" element twice—once under "Ship To" and once under "Bill To").

Specifications

Gentran:Server supports XML with the following specifications:

- The XML document must be *well-formed* (meets the well-formed document criteria that is specified for XML).

  Note
  If the document is not well-formed, Gentran:Server generates an error message.

- Gentran:Server does not validate against the Document Type Definition (DTD) during translation but it does use the DTD when creating maps.
- In this release, Gentran:Server supports external parameter entities but does *not* support mapping of external entities, notations, elements of type "ANY," comments, conditional sections, internal DTDs, conditional sections, unparsed entities (non-XML data), or processing instructions. In most cases, Gentran:Server ignores the items listed as not supported.
- Gentran:Server can read UTF-8 and UTF-16 encoded files if a compatible code page is loaded on your computer.

Menu options

If the XML side of a map is currently selected, the following menu options can be selected from the Edit\Create Sub and Edit\Insert menus (unavailable items are dimmed):

- Element
- Content particle
- Pcdata
- Attribute

Note
The Create Sub and Insert functions are now available when you right-click a map object.

XML Map Objects

Introduction

Gentran:Server now uses a set of icons to represent XML map objects.

Map object icons

This table describes the map object icons that Gentran:Server uses to visually represent the XML file:

Icon	Description
	The **XML File icon** represents the XML document that Gentran:Server is mapping, including the root element. It is a looping structure that contains elements and/or content particles 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.
	An **XML element** contains related elements and/or content particles. In addition, an element can contain one pcdata and/or one attribute container. These objects repeat in sequence until either the element data ends or the maximum number of times that the loop is allowed to repeat is exhausted. A repeating element that contains another repeating element corresponds to a nested looping structure.
	A **content particle** contains related elements and/or content particles that define either a choice or a sequence. A content particle can also contain one pcdata. If specified, these objects can repeat in sequence until either the content particle data ends or the maximum number of times that the loop is allowed to repeat is exhausted. If you create a content particle that is subordinate to another content particle, this corresponds to a nested looping structure (a loop within a loop).
	A **pcdata** object contains character data. Only one pcdata object can be defined per element or content particle. Gentran:Server automatically names the pcdata object with the name of the parent element or content particle. When a pcdata has an operation performed against it (link, standard rule, or as an extended rule storage field), the system displays a red checkmark over the pcdata icon. (Continued on next page)

(Contd) Icon	Description
	An **attribute container object** does not correspond to an XML feature. Gentran:Server uses attribute container objects to contain the attributes of an XML element. This object has no properties. An attribute container object is automatically created when the user creates the first attribute of an XML element. Subsequent attribute objects are created in the existing attribute container object.
	The **attribute object** specifies information associated with an element that further defines the element. An attribute container object is automatically created when the user creates the first attribute of an XML element. Subsequent attribute objects are created in the existing attribute container object. **Note** Attributes do not have to occur in sequence in the input data. When an attribute has an operation performed against it (link, standard rule, or as an extended rule storage field), the system displays a red checkmark over the attribute icon.

Overview of XML Translation Encoding Support

Introduction

This section describes the manner in which Gentran:Server supports XML encoding for inbound and outbound translation.

Inbound

For inbound translation, Gentran:Server processes XML files that were created using one of the encoding in this table.

Encoding	Windows Code Page	Notes
UTF-7	65000	N/A
UTF-8	N/A	Unicode Transformation Format, 8-bit encoding form
UTF-16	N/A	Unicode Transformation Format, 16-bit encoding form
Shift-Jis	932	Japanese
Big5	950	Traditional Chinese
GB2312	936	Simplified Chinese
dos-862	862	N/A
dos-720	720	N/A
EUC-JP	51932	N/A
HZ-GB-2312	52936	N/A
ISO-8859-2	28592	N/A
ISO-8859-4	28594	N/A
ISO-8859-5	28585	N/A
ISO-8859-7	28597	N/A
ISO-8859-9	28599	N/A
ISO-2022-JP	50220	N/A
ISO-2022-KR	50225	N/A
KOI8-R	20866	N/A
KSC_5601	949	N/A

(Continued on next page)

(Contd) Encoding	Windows Code Page	Notes
Windows-874	874	N/A
Windows-1250	1250	N/A
Windows-1251	1251	N/A
Windows-1252	1252	N/A
Windows-1253	1253	N/A
Windows-1254	1254	N/A
Windows-1255	1255	N/A
Windows-1256	1256	N/A
Windows-1257	1257	N/A
Windows-1258	1258	N/A

Note
You should always include an encoding attribute in the XML declaration of the input XML file unless the file is encoded using UTF-8, which is the default per the XML specification. If you do not include an encoding attribute in the XML declaration, the translator assumes the file is UTF-8 encoded.

Example
```
<?xml version="1.0" encoding="Big5"?>
```

Outbound

For outbound translation, Gentran:Server creates XML files with the encoding listed in this table.

Note
You configure the encoding in the properties of the root tag of the XML map.

Reference
See *Working with the XML File* on page 2 - 17 for more information.

Encoding	Notes
UTF-8	Unicode Transformation Format, 8-bit encoding form
UTF-16	Unicode Transformation Format, 16-bit encoding form

(Continued on next page)

<table>
<tr><th>Encoding</th><th colspan="2">Notes</th></tr>
<tr><td>Default</td><td colspan="2">If you select this encoding when you define the properties of the XML map, the encoding that is actually written to the output file by the system depends on the code page that is currently installed in the operating system. This table relates the encoded string to its associated Windows code page.</td></tr>
<tr><td></td><td>Default Encoding</td><td>Windows Code Page</td></tr>
<tr><td></td><td>UTF-7</td><td>65000</td></tr>
<tr><td></td><td>UTF-8</td><td>N/A</td></tr>
<tr><td></td><td>UTF-16</td><td>N/A</td></tr>
<tr><td></td><td>Shift-Jis</td><td>932</td></tr>
<tr><td></td><td>Big5</td><td>950</td></tr>
<tr><td></td><td>GB2312</td><td>936</td></tr>
<tr><td></td><td>dos-862</td><td>862</td></tr>
<tr><td></td><td>dos-720</td><td>720</td></tr>
<tr><td></td><td>EUC-JP</td><td>51932</td></tr>
<tr><td></td><td>HZ-GB-2312</td><td>52936</td></tr>
<tr><td></td><td>ISO-8859-1</td><td>1252</td></tr>
<tr><td></td><td>ISO-8859-2</td><td>28592</td></tr>
<tr><td></td><td>ISO-8859-4</td><td>28594</td></tr>
<tr><td></td><td>ISO-8859-5</td><td>28585</td></tr>
<tr><td></td><td>ISO-8859-7</td><td>28597</td></tr>
<tr><td></td><td>ISO-8859-9</td><td>28599</td></tr>
<tr><td></td><td>ISO-2022-JP</td><td>50220</td></tr>
<tr><td></td><td>ISO-2022-KR</td><td>50225</td></tr>
<tr><td></td><td>KOI8-R</td><td>20866</td></tr>
<tr><td></td><td>KSC_5601</td><td>949</td></tr>
<tr><td></td><td>Windows-874</td><td>874</td></tr>
<tr><td></td><td>Windows-1250</td><td>1250</td></tr>
</table>

(Continued on next page)

Encoding	Notes	
Default (contd)	**Default Encoding**	**Windows Code Page**
	Windows-1251	1251
	Windows-1252	1252
	Windows-1253	1253
	Windows-1254	1254
	Windows-1255	1255
	Windows-1256	1256
	Windows-1257	1257
	Windows-1258	1258

Example

```
<?xml version="1.0" encoding="Windows-1257"?>
```

Note

If the operating system has any other code page installed, Gentran:Server generates the output file using the UTF-8 encoding, which is the default per the XML specification.

Data Definition Format

Overview

Introduction

The Data Definition Format (DDF) is a file format that describes file formats that can be imported or exported into Gentran:Server Application Integration. This enables you to easily define your proprietary files to Gentran:Server.

DDF import

Gentran:Server supports the import of DDF files in two ways:

- When you create a new map, the New Map Wizard allows you to select DDF files to use as the basis for creating the INPUT and/or OUTPUT file formats.

  Reference
  See *Creating an XML Map* on page 2 - 8 for more information on creating a new map using DDF.

- You can use the Open File Definition function to replace the file format on the select side of the map with a previously-defined file format (either .IFD or .DDF).

  Reference
  See *How to Import a DDF* on page 1 - 12 for more information on loading a DDF file definition.

DDF export

You can use the Save File Definition function to save the file format of the selected side of a map as a DDF file.

What does the DDF contain?

The DDF contains a definition of a file format used in a map, including the hierarchical and looping structure of the data, and the map objects (i.e., groups, records, field, etc.) and their attributes (i.e., names, descriptions, data types, etc.). The DDF is expressed in XML, as defined in the Gentran_DDF DTD.

Note
The Gentran_DDF DTD (which describes how Gentran:Server stores the side of a map) is included on the XML installation disc.

DDF restrictions

Gentran:Server DDFs have the following restrictions:

- The DDF does not contain standard or extended rules, links, or any other map information that does not specifically relate to the file format. The exceptions to this rule are the use of code lists and constants. If the map references a "Use code" standard rule and instructs the system to raise a compliance error if the code is not found in the code list, Gentran:Server saves the standard rule. Gentran:Server saves every code list and constant, regardless of whether it is used in a rule.

  Reference
  See the *Application Integration User's Guide* for more information on code lists and using the "Use code" standard rule.

- To use DDF with Gentran:Server, you must have the Microsoft XML Parser installed. You can obtain the XML Parser by installing the latest release of Microsoft's Internet Explorer. If you do not have the XML Parser installed, Gentran:Server cannot support DDF and will not display it as an available option.
- Gentran:Server assigns acceptable defaults when attributes are not included.

  Example
  If a value is not specified for Minimum Length, the system assigns zero.

How to Import a DDF

Introduction

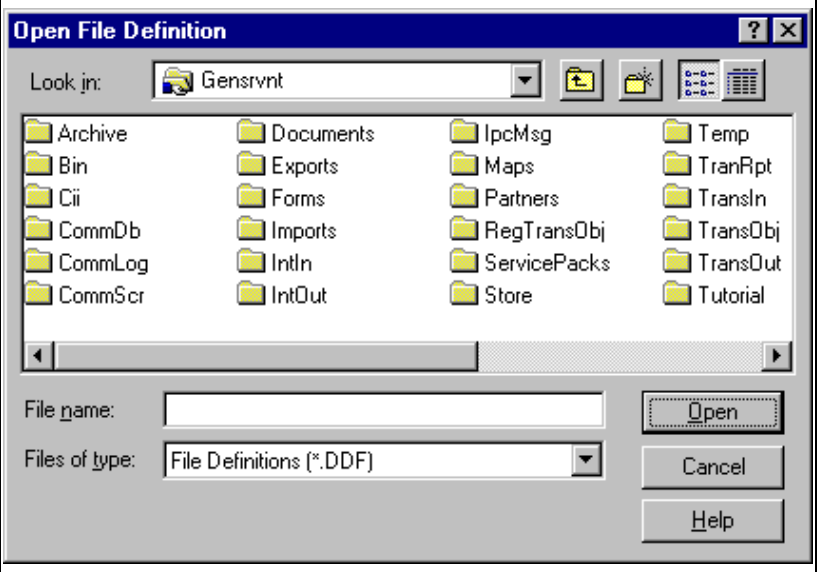
Gentran:Server enables you to load an individual file format definition that you previously saved. This feature provides you with a quick way to build either side of your map.

Reference
See *How to Export a DDF* on page 1 - 14 for more information on saving a file definition.

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

Procedure

Use this procedure to import a DDF.

Step	Action
1	Right-click the **File Format icon** (either the INPUT or OUTPUT side of the map) and select **Open File Definition** from the shortcut menu. **Note** If you already used Gentran:Server to create that side of the map, you are prompted with a message that warns you that the existing file format will be replaced. Click **Yes** to continue. **System response** The system displays the Open File Definition dialog box.  (Continued on next page)
2	From the Files of type list, select DDF as the default file extension that you want to display.

(Contd) Step	Action
3	Does the Look in list reflect the drive\folder where Gentran:Server is installed (the default is GENSRVNT)? ◗ If *yes*, continue with step 3. ◗ If *no*, select the appropriate folder and continue with step 3.
4	From the list, select the file definition that you want to load into the system or type the file name in the Filename box.
5	Click **Open**. **System response** The system loads the selected file format definition. **Note** If the DDF is invalid, the system displays a message box explaining the problem and terminates the import. You can find and fix many basic errors by running your DDF and the Gentran_DDF DTD through a validating XML parser.

How to Export a DDF

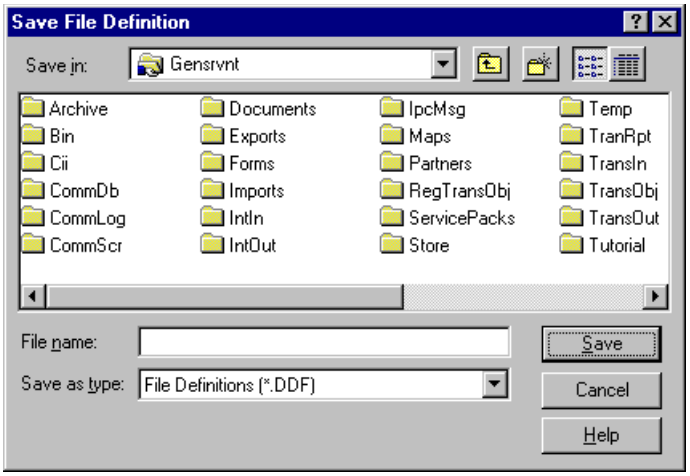
Introduction

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

Reference
See *How to Import a DDF* on page 1 - 12 for more information on loading a file definition.

Procedure

Use this procedure to export a DDF.

Step	Action
1	Right-click the **File Format icon** (either the INPUT or OUTPUT side of the map) and select **Save File Definition** from the shortcut menu. **System response** The system displays the Save File Definition dialog box. 
2	Does the Save in list reflect the drive\folder where Gentran:Server is installed (the default is GENSRVNT)? ◗ If *yes*, continue with step 3. ◗ If *no*, select the appropriate folder and continue with step 3.
3	Type the file name in the Filename box or select the file definition that you want to save from the list.
4	Click **Save**. **System response** The system saves the file format definition.

Gentran:Server DDF Document Type Definition

Introduction

The Gentran:Server Data Definition Format DTD (Gentran_DDF) describes how Gentran:Server stores one side of a map, including which elements are required, how they are arranged, and which elements they can contain.

The Gentran_DDF is included on the Installation CD.

CHAPTER

2

Using XML with Gentran:Server

Contents

Getting Started

Overview

XML Process This table contains the process that you follow to use XML with Gentran:Server.

Stage	Description
1	Install the Gentran:Server XML components on every machine in your system. **Reference** See the Options Pack Installation card for more information.
2	Install the XML standards on every machine in your system. **Reference** See *Installing the XML Standards from ESD Portal* on page 2 - 7 for more information.
3	Create a splitter entry to identify and extract XML documents. **Reference** See *Modifying the Splitter Configuration* on page 2 - 6 for more information.
4	Create a map to translate XML documents. **Reference** See *Creating an XML Map* on page 2 - 8 for more information.
5	If necessary, build the INPUT and OUTPUT sides of the map using one of these three methods. ◗ Load the data format from a saved definition. **Reference** See *How to Import a DDF* on page 1 - 12 for more information. ◗ Create map objects manually. **Reference** See *Working with XML Map Objects* on page 2 - 15 for more information. ◗ Load a map side with the New Map Wizard Customize option. **Reference** See *Creating an XML Map* on page 2 - 8 for more information. (Continued on next page)

(Contd) Stage	Description
6	Determine if the Gentran:Server for Windows version 5.0 translator changes require you to modify your maps. **Reference** See *Translator Functionality Changes for XML Data* on page 2 - 4 for more information.

Translator Functionality Changes for XML Data

Introduction

This section describes the translator changes implemented with Gentran:Server for Windows version 5.0. In this version, the behavior of the inbound XML translator is slightly different in regard to handling unknown elements. If the input file contains an XML element that was not defined in the map then the XML parser/translator simply skips it and continues processing the rest of the data. This differs in behavior from previous Gentran:Server releases in which if an unknown element failed to match anything in the map then translation of the Input side of the map would finish without reading beyond the unknown element. Essentially this means the XML translator will now read the full input file rather than stopping if it encounters an unknown element.

Note
This functionality does not affect any other data formats (delimited, positional, NCPDP, ODBC, CII).

Types of maps affected

The new behavior may affect the following types of maps:

- XML build maps that use ReadBlock/WriteBlock extended rules on the Output side of the map to read data from the input file.
- XML break maps that use ReadBlock/WriteBlock extended rules on the Output side of the map to read data from the input file.
- Any document level XML map that has a ReadBlock/WriteBlock extended rule on the output side to read XML data from the input file.

How to modify these maps

You need to modify any maps that utilize ReadBlock/WriteBlock on the Output side to receive the desired output. Follow these steps to modify the maps.

Step	Action

1	On the Pre-Session On-Begin rules create and initialize a global integer variable.
	
	Note
	This variable is used to store the file position from the point in the Input file where the ReadBlock/WriteBlock should be performed.
	
	Example
	`Integer Input_pos;`
	`Input_pos = 0;`
	
	(Continued on next page)

(Contd) Step	Action
2	On the Input side of the map on the last valid element defined that receives data, create an extended rule that performs the FTELL function. **Note** This stores the file position immediately following the end tag of that element. **Example** `Input_pos = Ftell(0);`
3	Add the FSEEK function to the extended rule on the Output side of the map on the line before the ReadBlock/WriteBlock is performed. **Note** This will place the file pointer back to the correct position in the Input file so the ReadBlock/WriteBlock can be performed. **Example** `Fseek(0,Input_pos,current);` **Note** For more information on the Ftell/Fseek extended rule functions, please see Alphabetical Language Reference in the *Application Integration User's Guide*.

Modifying the Splitter Configuration

Introduction

Gentran:Server uses splitters in the communications process to extract enveloped data from a transmission file and determine which transaction break translation objects should be used to process the data.

Splitter entries are defined on the System Configuration program Splitter tab. Each splitter entry contains the parameters that are necessary for the system to identify and split interchanges for received data.

Prior to receiving inbound XML data, you need to create a Gentran:Server splitter entry to identify and extract XML documents.

Note
You only need to create a new splitter entry if you are receiving inbound XML data. You need to create a splitter entry for *each* XML document that has a different start tag.

Procedure

Use this procedure to create a new splitter entry for XML documents.

Step	Action
1	From any machine on which the System Configuration program is installed, start the **System Configuration** program. **System response** The system displays the System Configuration dialog box (Controllers tab).
2	Select the **Splitter** tab. **System response** The splitter parameters are displayed.
3	Click **New**. **System response** The system allows you to define a new splitter entry.
4	From the Type list, select XML.
5	In the Start box, select the DOCTYPE or root element of your XML document.
6	From the Translation Object section, select the appropriate break translation objects.
7	◗ Click **OK** to exit the System Configuration program, or, ◗ click **Apply** to save any changes without exiting System Configuration.

Installing the XML Standards from ESD Portal

Introduction

These instructions contain procedures for downloading, unpacking, and installing the XML standards. The file you download from the Electronic Software Distribution (ESD) Portal is named **gsw50_xml_standards_cd.zip**.

Downloading the standards file

To download the gsw50_xml_standards_cd.zip file:

Step	Action
1	In the E-mail message you received from Sterling Commerce, click the link to the Electronic Software Distribution Download Area.
2	In the Security Alert dialog box, click **Yes**.
3	On the Login page, type the following information: ◗ Sterling order number ◗ Company name ◗ Customer ID
4	Click **Login**, and the system displays the Download Area.
5	Locate **Gentran:Server for Windows, version 5.0** and click **Download**.
6	In the File Download dialog box, click **Save**.
7	When the Save As dialog box opens, save the gsw50_xml_standards_cd.zip file to the C:\ drive. **Note** If Internet Explorer adds a number in brackets to the name of the downloaded file (for example, CD[1].platformoruniqueproductname.version), rename the file on the Windows system before you transfer it to the system where it will be installed.

Installing the standards

To install the XML standards:

Step	Action
1	Right-click the **gsw50_xml_standards_cd.zip** icon, and extract (unzip) the file to an empty directory. **Note** A folder named xml_standards_cd is created.
2	To install the core product, from the xml_standards_cd folder, run **setup.exe**.

Creating an XML Map

Introduction

The New Map Wizard enables you to quickly and easily create a map. As part of the map creation process, the Delimited EDI wizard enables you to create the map side format from the standards database. The New XML Wizard enables you to create your format from a selected predefined document source type (i.e., a DTD).

New XML Wizard restrictions

The New XML Wizard exhibits the following behavior:

- Raises a warning if it encounters attributes that use entities or notations.
- Changes attributes of type ENTITY or ENTITIES to type CDATA.
- Changes attributes of type NOTATION to type ENUMERATED.
- Ignores comments and processing instructions.
- Discards external general entities and notations.
- Does *not* support XML namespaces or conditional sections.
- Supports external parameter entities that reference a URL *only if* Internet Explorer 3.0 or higher is installed on the machine.

  Note
  To use the Data Definition Format (DDF), you must have Internet Explorer 5.0 or greater installed on your machine.

Procedure

Use this procedure to create an XML map.

Step	Action
1	From the File menu, select **New**. **System response** The system displays the New Map Wizard. (Continued on next page)

<table>
<tr><th>(Contd) Step</th><th>Action</th></tr>
<tr><td>2</td><td>
Answer the following questions and then click Next.

What kind of map are you creating?
The following table defines the selections:

<table>
<tr><th>Part</th><th>Function</th></tr>
<tr><td>Import</td><td>Used for outbound maps.</td></tr>
<tr><td>Export</td><td>Used for inbound maps.</td></tr>
<tr><td>Turnaround</td><td>Used for EDI to EDI maps.</td></tr>
<tr><td>Transaction build</td><td>Used in advanced mapping to build transaction envelopes.</td></tr>
<tr><td>Transaction break</td><td>Used in advanced mapping to separate documents.</td></tr>
<tr><td>Functional group build</td><td>Used in advanced mapping to build functional group envelopes.</td></tr>
<tr><td>Functional group break</td><td>Used in advanced mapping to separate functional groups.</td></tr>
<tr><td>Interchange build</td><td>Used in advanced mapping to build interchange envelopes.</td></tr>
<tr><td>Interchange break</td><td>Used in advanced mapping to separate interchanges.</td></tr>
<tr><td>F/A Inbound</td><td>Used in advanced mapping to reconcile functional acknowledgements.</td></tr>
<tr><td>F/A Outbound</td><td>Used in advanced mapping to generate functional acknowledgements.</td></tr>
</table>

What is the name of the map?
Type the unique name of the map. The system adds the .MAP extension.
What is your name?
Type your name if it differs from the user name prompted by the system.

System response
The system displays the New Map Wizard - Input Format dialog box.

Note
You need to complete the format of the Input side of the map. This is the format of the data that is translated by the Gentran:Server system.

(Continued on next page)
</td></tr>
</table>

(Contd) Step	Action
3	For the input side of the map: do you want to create a new data format using a syntax that you define? ◗ If *yes*, select one of the following input format options and continue with step 4: **Delimited EDI** (Electronic Data Interchange file) **ODBC** (Open Database Connectivity) **Positional** (VDA, GENCOD, application files, etc.) **XML** (Extensible Markup Language) **Note** For Inbound maps (Export), the Input Format Type is usually XML or EDI. For Outbound maps (System Import or Import), the Input Format Type is usually XML, ODBC, or Positional. For Turnaround maps, the Input Format Type is usually EDI or XML. ◗ If *no* (you want to load the data format from a saved definition), select the Load the data format from a saved definition option, and type the path and file name of the saved definition (or click **Browse** to display the Open File Definition dialog box). Continue with step 6.
4	Did you select Delimited EDI or XML and want to customize the format? ◗ If *yes*, click **Customize** and continue with the next step. **System response** The system displays the New Delimited EDI Wizard or New XML Wizard dialog box. **Note** The Delimited EDI wizard enables you to create your format from the standards database. The New XML Wizard enables you to create your format from a selected predefined document source type (i.e., a DTD). ◗ If *no*, click **Next** and continue with step 7. **System response** The system displays the New Map Wizard - Output Format dialog box. (Continued on next page)

<table>
<tr><th>(Contd) Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">5</td><td colspan="2">Follow the steps for the appropriate dialog box and then continue with step 7.</td></tr>
<tr><td>IF the dialog box displayed is...</td><td>THEN follow this procedure...</td></tr>
<tr><td>New XML Wizard</td><td>a. Select the document source type and click Next.

b. Type the name of your DTD file or a URL pointing to the DTD and click Next.

c. Select the doctype, set the maximum length of data elements, and click Next.

Notes
— The DTD does not explicitly define the root element, so you can choose from all the elements defined in the DTD. By default, the wizard selects the first element encountered in the DTD.

— You can specify the maximum length of data elements because this is not defined in the DTD.

d. Click Finish.

Note
If the system needed to make changes to the DTD to make it compliant with Gentran:Server, the system informs you of the changes. Click OK.

(Continued on next page)</td></tr>
</table>

(Contd) Step	Action	
5 (contd)	New Delimited EDI Wizard	a. Click **Next**. b. Select the ODBC data source that contains the standards database. c. Select the standards agency, version, transaction set, and release (for TRADACOMS only) and click **Next**. d. Click **Finish**.
	System response The system displays the New Map Wizard - Output Format dialog box.	
6	Did you choose to load the data format from a saved definition and click **Browse** to display the Open File Definition dialog box? ◗ If *yes*, type the file name and click **Open** to load the selected file format definition, and then continue with step 7. **Note** You can now select either a .DDF or .IFD file. ◗ If *no*, continue with step 7. **Note** If the DDF is invalid, the system displays a message box explaining the problem and terminates the import.	
7	For the output side of the map: do you want to create a new data format using a syntax that you define? ◗ If *yes*, select one of the following input format options and continue with step 4: **Delimited EDI** (Electronic Data Interchange file) **ODBC** (Open Database Connectivity) **Positional** (VDA, GENCOD, application files, etc.) **XML** (Extensible Markup Language) **Note** For Inbound maps (Export), the Input Format Type is usually XML or EDI. For Outbound maps (System Import or Import), the Input Format Type is usually XML, ODBC, or Positional. For Turnaround maps, the Input Format Type is usually EDI or XML. ◗ If *no* (you want to load the data format from a saved definition), select the Load the data format from a saved definition option and type the path and file name of the saved definition (or click **Browse** to display the Open File Definition dialog box). Continue with step 9. (Continued on next page)	

<table>
<tr><th>(Contd) Step</th><th colspan="2">Action</th></tr>
<tr><td>8</td><td colspan="2">Did you select Delimited EDI or XML and want to customize the format?
◗ If yes, click Customize and continue with the next step.
System response
The system displays the New Delimited EDI Wizard or New XML Wizard dialog box.
Note
The Delimited EDI wizard enables you to create your format from the standards database. The New XML Wizard enables you to create your format from a selected predefined document source type (i.e., a DTD).
◗ If no, click Finish to load the standards information you selected and create the new map (this may take a few seconds).
System response
The system displays the new map in the Application Integration Window.</td></tr>
<tr><td rowspan="2">9</td><td colspan="2">Follow the steps for the appropriate dialog box and then continue with step 11.</td></tr>
<tr><td>IF the dialog box displayed is...</td><td>THEN follow this procedure...</td></tr>
<tr><td>New XML Wizard</td><td>a. Select the document source type and click Next.
b. Type the name of your DTD file or a URL pointing to the DTD and click Next.
c. Select the doctype, set the maximum length of data elements, and click Next.
Notes
— The DTD does not explicitly define the root element, so you can choose from all the elements defined in the DTD.
— You can specify the maximum length of data elements because this is not defined in the DTD. By default, the wizard selects the first element encountered in the DTD.
d. Click Finish.
Note
If the system needed to make changes to the DTD to make it compliant with Gentran:Server, the system informs you of the changes. Click OK.
(Continued on next page)</td></tr>
</table>

(Contd) Step	Action	
9 (contd)	New Delimited EDI Wizard	a. Click **Next**. b. Select the ODBC data source that contains the standards database. c. Select the standards agency, version, transaction set, and release (for TRADACOMS only) and click **Next**. d. Click **Finish**.
	System response The system displays the New Map Wizard dialog box.	
10	Did you choose to load the data format from a saved definition and click **Browse** to display the Open File Definition dialog box? ◗ If *yes*, type the file name and click **Open** to load the selected file format definition, and then continue with step 11. **Note** You can now select either a .DDF or .IFD file. ◗ If *no*, continue with step 11.	
11	Click **Finish** to load the standards information you selected and create the new map (this may take a few seconds). **System response** The system displays the new map in the Application Integration Window. **Note** After you finish creating and saving a new map, you need to define the Input and Output sides of the map. The steps you take are different, depending on whether the map is an Import, System Import, Export, or Turnaround map.	

Working with XML Map Objects

Overview: Creating Map Objects

Introduction

The map objects that you can create depends on which map object is currently selected (has focus in the map). This table describes the available options (N/A indicates that no map object can be created when the specified object is selected).

IF the currently-selected object is a...	THEN you can create...
XML File 	◗ Element ◗ Content Particle ◗ Pcdata ◗ Attribute
Element 	◗ Element ◗ Content Particle ◗ Pcdata ◗ Attribute
Content Particle 	◗ Element ◗ Content Particle ◗ Pcdata
Pcdata	N/A
Attribute Container 	Attribute
Attribute 	N/A

Create Sub vs. Insert functions

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

IF you want to create a map object...	**THEN right-click the map object to access the shortcut menu and select...**
at the same level (equal) as the selected map object,	◗ Insert and then ◗ select the appropriate option.
that is subordinate to the selected map object,	◗ Create Sub and then ◗ select the appropriate option.

Working with the XML File

Introduction

The XML File object represents the XML document that Gentran:Server is mapping, including the root element. This object is created automatically by Gentran:Server. This table describes the two dialog boxes unique to the XML File object.

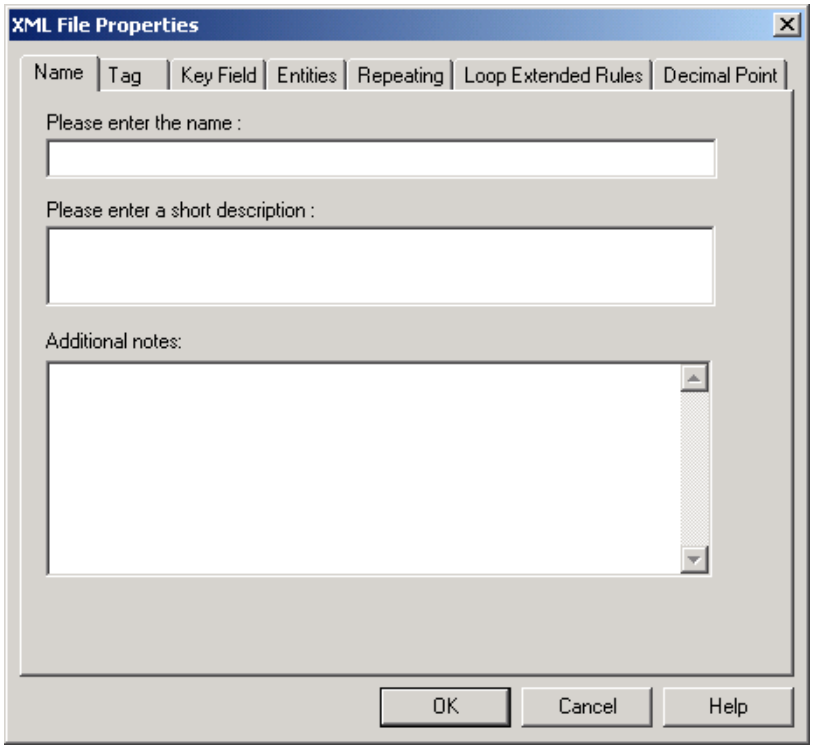
Part	Function
XML File Properties dialog box	Enables you to define entities, output format specifications, and file-level extended rules.
Entity Properties dialog box	Enables you to define entities. **Note** Accessible through the XML File Properties dialog box.

Note
The XML File object cannot be referenced by standard rules or links.

Using entities
Gentran:Server allows you to define internal general parsed entities, according to the XML definition.

XML File Properties dialog box

This diagram illustrates the XML File Properties dialog box (Name tab) on the OUTPUT side of the map.



XML File Properties parts and functions

This table lists the parts of the XML File Properties dialog box and their functions.

Part	Function
Name tab	
Name	Identifies the XML file.
Description	Describes the XML file. This box is used to differentiate the XML file from similar files.
Tag tab	
Tag	Identifies the XML tag for the root element of the document. **Default** INPUT or OUTPUT
Entities tab	
Name	Specifies the name of the entity.
Description	Specifies a brief description of the entity.
New	Accesses the Entity Properties dialog box, which enables you to create an entity.
Change	Accesses the Entity Properties dialog box, which enables you to edit the selected entity.
Delete	Deletes the selected entity.
Copy	Copies the selected entity. This enables you to copy an existing entity for use in another map.
Paste	Pastes a previously-copied entity. This enables you to copy entities from one map to another.
Repeating tab	
Conditional	Indicates that the XML file object is not mandatory.
Mandatory	Indicates that the XML file object must appear in the map.
Output tab (OUTPUT side of the map *only*)	
No prolog or document type declaration	Indicates that the system does not generate any header information for the XML document (neither a prolog nor a document type declaration).
Prolog specified	Indicates that the system generates a prolog at the start of the XML document.

(Continued on next page)

(Contd) Part	Function
Prolog and document type declaration specified	Indicates that the system generates both a prolog and a document type declaration at the start of the XML document.
Public ID	Specifies the public identifier that the system uses to create the document type declaration. **Note** This box is only available if you select the Prolog and document type declaration option.
System ID	Specifies the system identifier that the system uses to create the document type declaration. **Note** This box is only available if you select the Prolog and document type declaration option.
No Newlines	Indicates that the output data will be wraparound (streamed).
One element per line, indented	Indicates that the output data is formatted hierarchically and indented.
One element per line, no indentation	Indicates that the output data is formatted hierarchically but not indented.
Loop Extended Rules tab	
On Begin	Specifies that the extended rule is executed before the system processes the map object.
On End	Specifies that the extended rule is executed after the system concludes processing the map object.
Full Screen	Maximizes the dialog box.
Compile	Compiles the extended rule. Any warnings or errors are displayed in the Errors list. **Note** This function gives you immediate feedback about the accuracy of your rule. The rule is compiled when you compile the entire translation object.
Extended rule	Defines the extended rule.
Errors	Displays any errors generated when you clicked **Compile** to compile the extended rule.

(Continued on next page)

(Contd) Part	Function
Decimal Point	
Define Decimal Point	Whether the decimal point character is user-defined.
Decimal Point Character	The character that the translator will identify as a decimal point. The default is a period (.).

Modifying XML file properties

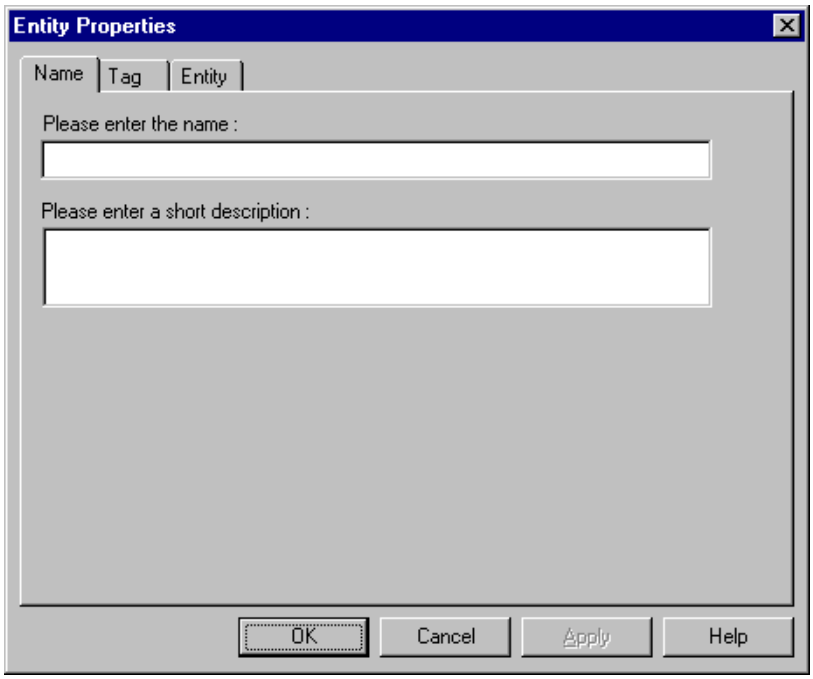
Use this procedure to modify the properties of an XML file.

Step	Action
1	Right-click the **XML File icon** and select **Properties** from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default).
2	Do you want to create an entity? ◗ If *yes*, select the **Entities** tab to access entity options. **Reference** See *Creating an entity* on page 2 - 23 for more information. ◗ If *no*, continue with step 3.
3	Do you want to modify the output options for the XML file? ◗ If *yes*, select the **Output** tab and continue with step 4. ◗ If *no*, continue with step 5.
4	On the Output tab, specify the following: ◗ whether the system generates a prolog and/or document type declaration ◗ public ID (if applicable) ◗ system ID (if applicable) ◗ how the XML elements are output to the file (Continued on next page)

(Contd) Step	Action
5	Do you want to specify an extended rule for the XML file? ◗ If *yes*, select the **Loop Extended Rules** tab, define the rule, and continue with step 6. **Reference** See the *Application Integration User's Guide* for more information on extended rules. ◗ If *no*, continue with step 6.
6	Do you want to specify that each decimal point in your data is defined and generated as a something other than the default period (.)? ◗ If *yes*, follow these steps: — Select the **Decimal Point** tab. — In the **Decimal Point** tab, select the **Define Decimal Point** check box. — In the **Decimal Point Character** box, type the value you want Sterling Integrator to use as a decimal point—for example, a comma (,). This resets the decimal point default to use what you specified instead of a period. — Click **OK**. You must click OK, even if the delimiter fields are empty, to reset the delimiters. ◗ If no, continue with the next step.
7	Click **OK**. **System response** The system saves your changes and closes the XML File Properties dialog box.

Entity Properties dialog box

This diagram illustrates the Entity Properties dialog box.



Entity Properties parts and functions

This table lists the parts of the Entity Properties dialog box and their functions.

Part	Function
Name tab	
Name	Identifies the entity. **Note** This is a descriptive name.
Description	Describes the entity. This box is used to differentiate the entity from similar entities.
Tag tab	
Tag	Defines the entity identification tag, as it appears in the XML document. **Default** The system uses the entity name by default.

(Continued on next page)

Part	Function
Entity tab	
Entity Value	Specifies the entity data. **Note** This is the text that the system inserts when it encounters the entity.

Creating an entity

Use this procedure to create an entity.

Step	Action
1	Right-click the **XML File icon** and select **Properties** from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default).
2	Select the **Entities** tab to access the entity options.
3	Click **New**. **System response** The system displays the Entity Properties dialog box.
4	On the **Name** tab, specify the following: ◗ unique entity name ◗ description (if applicable)
5	If necessary, select the **Tag** tab and change the value in the Tag box. **Note** The tag must match the entity tag in the XML document.
6	Select the **Entity** tab to access the entity options.
7	In the Entity Value box, type the entity data.
8	Click **OK**. **System response** The system saves the entity and closes the Entity Properties dialog box.
9	Click **OK**. **System response** The system closes the XML File Properties dialog box.

Working with XML Elements

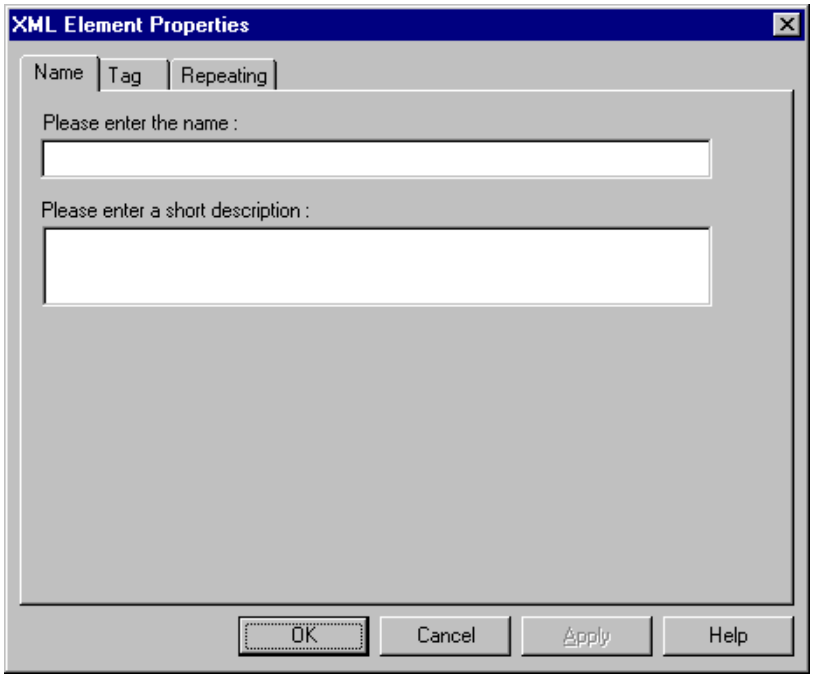
Introduction

An XML element contains related elements and/or content particles. In addition, an element can contain one pcdata and/or one attribute container. These objects repeat in sequence until either the element data ends or the maximum number of times that the loop is allowed to repeat is exhausted.

Note
The XML Element object cannot be referenced by standard rules or links.

XML Element Properties dialog box

This diagram illustrates the XML Element Properties dialog box (Name tab).



Parts and functions

This table lists the parts of the XML Element Properties dialog box and their functions.

Part	Function
Name tab	
Name	Defines the element name. **Notes** ◗ Each element must have a unique name. ◗ Do not use spaces or dashes (-) for the element name. ◗ You can use the underscore (_) to separate words.
Description	Describes the element. This box is used to provide a brief explanation of the element that allows you to differentiate it from similar elements.
Tag tab	
Tag	Defines the element identification tag, as it appears in the XML document. **Note** Gentran:Server validates the tag against the characters that XML allows for element names. **Default** The system uses the element name by default.
Key Field tab (only if element contains one or more attributes)	
Field	Specifies all the attributes that are defined for this element. **Note** The key field function enables you to specify a second qualification in selecting an element (the element name is the first qualification).
Use constant/ **Edit**	Indicates that the system must match the element if the contents of the selected attribute match the literal constant selected from the list. **Note** Click **Edit** (at the right of the Use constant list) to access the Translation Object Constants dialog box. (Continued on next page)

(Contd) Part	Function
Use codelist/ **Edit**	Indicates that the system must match the element if the contents of the selected attribute match the selected code list. **Note** Click **Edit** (at the right of the Use constant list) to access the Code Lists dialog box.
Match record when key does not match	Indicates that the system will match the element if the contents of the selected attribute does not contain the value specified in the Matching rules section. **Note** If the specified condition is not met, the element does not conform to the definition, and processing continues.
Repeating tab	
Conditional	Indicates that the element is not mandatory.
Mandatory	Indicates that the element must appear in the map.
Can not repeat	Indicates that the element does not repeat (is a single instance).
Can repeat	Indicates that the element can repeat (loop) as many times as necessary.
Can repeat, with a maximum usage	Indicates that the element can repeat (loop) as many times as is designated in the Maximum usage box.
Maximum usage	Defines how many times the element can repeat (loop).
Loop Extended Rules tab (only if element repeats)	
On Begin	Specifies that the extended rule is executed before the system processes the element.
On End	Specifies that the extended rule is executed after the system concludes processing the element.
Full Screen	Maximizes the dialog box.
Compile	Compiles the extended rule. Any warnings or errors are displayed in the Errors list. **Note** This function gives you immediate feedback about the accuracy of your rule. The rule is compiled when you compile the entire translation object.

(Continued on next page)

(Contd) Part	Function
Extended rule	Defines the extended rule.
Errors	Displays any errors generated when you clicked **Compile** to compile the extended rule.

Creating an element

Use this procedure to create an element.

Step	Action
1	Right-click a **map object** and select either **Create Sub** or **Insert** from the shortcut menu. **Reference** See *Overview: Creating Map Objects* on page 2 - 15 for more information on the Create Sub and Insert functions.
2	From the shortcut menu, select **Element**. **System response** The system displays the XML Element Properties dialog box.
3	On the Name tab, specify the following: ◗ unique element name and ◗ description (if applicable).
4	If necessary, select the **Tag** tab and change the value in the Tag box. **Note** This value must match the element in the XML document.
5	Select the **Repeating** tab to access the occurrence options.
6	Select either the Conditional or Mandatory option to specify whether the element is required in the map.
7	Select the appropriate repeating option for the element.
8	If you need to specify the number of times the element can repeat (loop), type that number in the Maximum usage box.
9	Did you specify that the element repeats (loops)? ◗ If *yes*, continue with step 10. ◗ If *no*, continue with step 11. (Continued on next page)

(Contd) Step	Action
10	Do you want to specify an extended rule for this element? ◗ If *yes*, select the **Loop Extended Rules** tab, define the rule, and continue with step 11. **Reference** See the *Application Integration User's Guide* for more information on extended rules. ◗ If *no*, continue with step 11.
11	Click **OK**. **System response** The system saves the element and closes the XML Element Properties dialog box.

Working with Content Particles

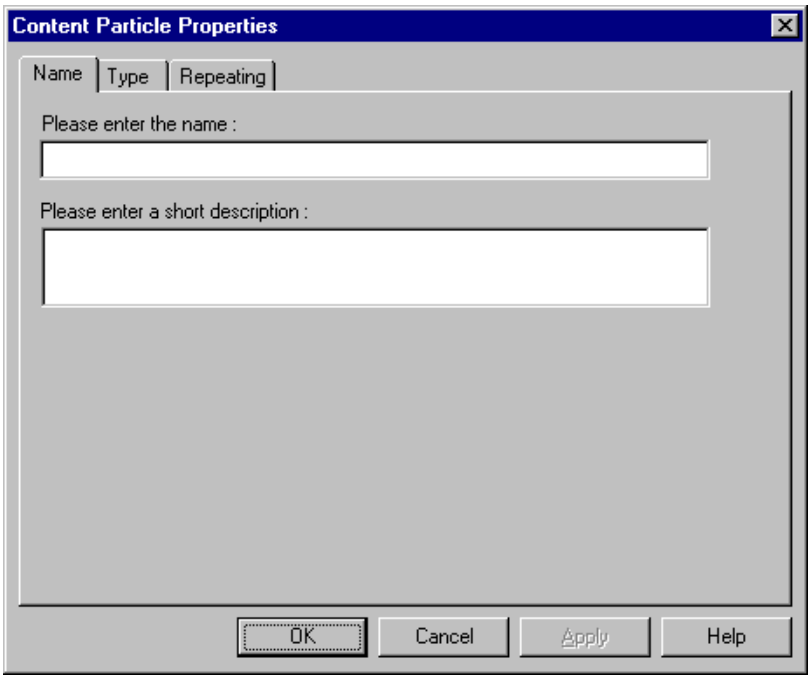
Introduction

In Gentran:Server, a content particle contains child objects that define either a choice or a sequence. A content particle can contain related elements and/or content particles. In addition, a content particle can contain one pcdata. If specified, these objects repeat in sequence until either the content particle data ends or the maximum number of times that the loop is allowed to repeat is exhausted.

Note
The Content Particle object cannot be referenced by standard rules or links.

Content Particle Properties dialog box

This diagram illustrates the Content Particle Properties dialog box (Name tab).



Parts and functions

This table lists the parts of the Content Particle Properties dialog box and their functions.

Part	Function
Name tab	
Name	Defines the name of the content particle. **Note** Do not use spaces or dashes (-) for the content particle name. You can use the underscore (_) to separate words.
Description	Describes the content particle. This box is used to provide a brief explanation of the content particle that allows you to differentiate it from similar content particles.
Type tab	
Choice (A | B)	Indicates that the child objects of the content particle represent a choice (a disjunction) of the child objects.
Sequence (A , B)	Indicates that the child objects of the content particle represent a sequence of the child objects.
All (A , B) | (B, A)	Not currently supported.
Repeating tab	
Conditional	Indicates that the content particle is not mandatory.
Mandatory	Indicates that the content particle must appear in the map.
Can not repeat	Indicates that the content particle does not repeat (is a single instance).
Can repeat	Indicates that the content particle can repeat (loop) as many times as necessary.
Can repeat, with a maximum usage	Indicates that the content particle can repeat (loop) as many times as is designated in the Maximum usage box.
Maximum usage	Defines how many times the content particle can repeat (loop).
Loop Extended Rules tab (only if element repeats)	
On Begin	Specifies that the extended rule is executed before the system processes the content particle.
On End	Specifies that the extended rule is executed after the system concludes processing the content particle.

(Continued on next page)

(Contd) Part	Function
Full Screen	Maximizes the dialog box.
Compile	Compiles the extended rule. Any warnings or errors are displayed in the Errors list. **Note** This function gives you immediate feedback about the accuracy of your rule. The rule is compiled when you compile the entire translation object.
Extended rule	Defines the extended rule.
Errors	Displays any errors generated when you clicked **Compile** to compile the extended rule.

Creating a content particle

Use this procedure to create a content particle.

Step	Action
1	Right-click a **map object** and select either **Create Sub** or **Insert** from the shortcut menu. **Reference** See *Overview: Creating Map Objects* on page 2 - 15 for more information on the Create Sub and Insert functions.
2	From the shortcut menu, select **Content Particle**. **System response** The system displays the Content Particle Properties dialog box.
3	On the Name tab, specify the following: ◗ unique content particle name and ◗ description (if applicable).
4	Select the **Type** tab to access the content particle type options.
5	Select the appropriate option to define what the child objects of the content particle represent.
6	Select the **Repeating** tab to access the occurrence options.
7	Select either the Conditional or Mandatory option to specify whether the content particle is required in the map.
8	Select the appropriate repeating option for the content particle. (Continued on next page)

(Contd) Step	Action
9	In the Maximum usage box, type the number of times the content particle can repeat (loop).
10	Did you specify that the content particle repeats (loops)? ◗ If *yes*, continue with step 11. ◗ If *no*, continue with step 12.
11	Do you want to specify an extended rule for this content particle? ◗ If *yes*, select the **Loop Extended Rules** tab, define the rule, and continue with step 12. **Reference** See the *Application Integration User's Guide* for more information on extended rules. ◗ If *no*, continue with step 12.
12	Click **OK**. **System response** The system saves the content particle and closes the Content Particle Properties dialog box.

Working with Pcdata

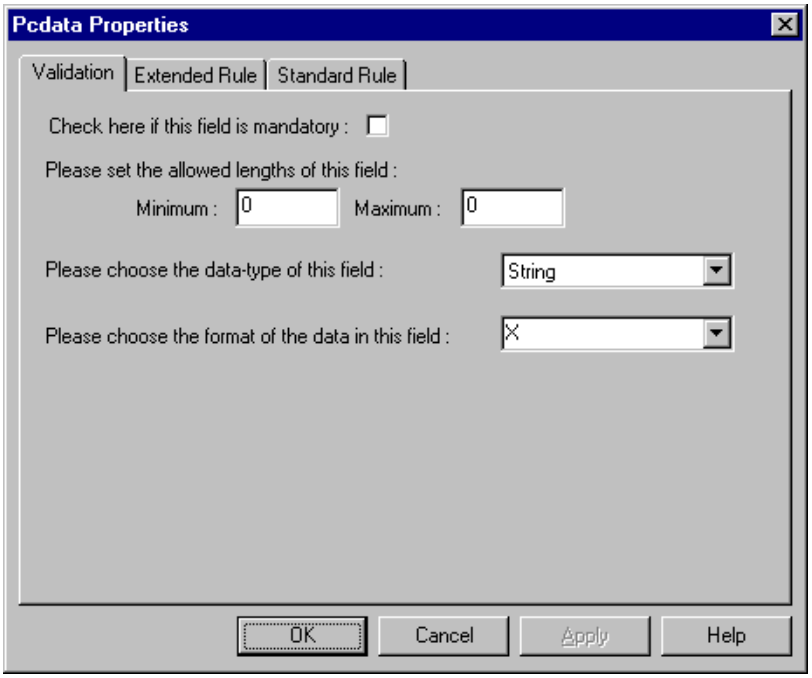
Introduction

A pcdata object contains character data in an XML document. Only one pcdata object can be defined per element or content particle. Gentran:Server automatically names the pcdata object with the name of the parent element or content particle.

Note
When a pcdata has an operation performed against it (link, standard rule, or as an extended rule storage field), the system displays a red checkmark over the pcdata icon.

Pcdata Properties dialog box

This diagram illustrates the Pcdata Properties dialog box (Validation tab).



Parts and functions

This table lists the parts of the Pcdata Properties dialog box and their functions.

Part	Function
Validation tab	
Mandatory	Indicates whether the pcdata is mandatory (must appear).
Minimum	Specifies the minimum length of the pcdata.
Maximum	Specifies the maximum length of the pcdata.

(Continued on next page)

(Contd) Part	Function
Data-type	Specifies the type of data. Valid values are: ◗ String (alphanumeric element) ◗ Number (numeric or real element) ◗ Date/Time (date or time element)
Format	Specifies how the pcdata is formatted. **Note** Depending on which Data-type you selected, you can either select the data format from a list (if you choose Number or Date/Time in the Type field), or enter a Syntax Token to denote that this field must be formatted as the specified Syntax Token dictates. When you installed Gentran:Server, you assigned a default format to the string fields. This format serves as the basis for character validation. Most U.S. users use a default format that corresponds to ASCII characters (for example, the X syntax token). Most users of Asian or European languages and encoded character sets should use the Free Format (0x01-0xFF).
Extended Rule tab	
Full Screen	Maximizes the dialog box.
Compile	Compiles the extended rule. Any warnings or errors are displayed in the Errors list. **Note** This function gives you immediate feedback about the accuracy of your rule. The rule is compiled when you compile the entire translation object.
Extended rule	Defines the extended rule.
Errors	Displays any errors generated when you clicked **Compile** to compile the extended rule.
Standard Rule tab	
Standard rule	Specifies a standard rule that will affect this field or element during processing. Each of the rules are mutually exclusive. **Reference** See the *Application Integration User's Guide* for more information on standard rules.

Creating a pcdata

Use this procedure to create a pcdata.

Step	Action
1	Right-click a **map object** and select either **Create Sub** or **Insert** from the shortcut menu. **Reference** See *Overview: Creating Map Objects* on page 2 - 15 for more information on the Create Sub and Insert functions.
2	From the shortcut menu, select **Pcdata**. **System response** The system displays the Pcdata Properties dialog box.
3	On the **Validation** tab, specify the following: ◗ whether the pcdata is required or not ◗ minimum length ◗ maximum length ◗ type of data ◗ how the data is formatted
4	Do you want to specify an extended rule for this pcdata? ◗ If *yes*, select the **Extended Rule** tab, define the rule, and continue with step 5. **Reference** See the *Application Integration User's Guide* for more information on extended rules. ◗ If *no*, continue with step 6.
5	Do you want to specify a standard rule for this pcdata? ◗ If *yes*, select the **Standard Rule** tab, define the rule, and continue with step 6. **Reference** See the *Application Integration User's Guide* for more information on standard rules. ◗ If *no*, continue with step 6.
6	Click **OK**. **System response** The system saves the pcdata and closes the Pcdata Properties dialog box.

Working with Attributes

Introduction

In Gentran:Server, each attribute is contained in an attribute container. An element can only have one attribute container object, but the attribute container object can enclose many attribute objects.

Note
When an attribute has an operation performed against it (link, standard rule, or as an extended rule storage field), the system displays a red checkmark over the attribute icon.

Attribute container object

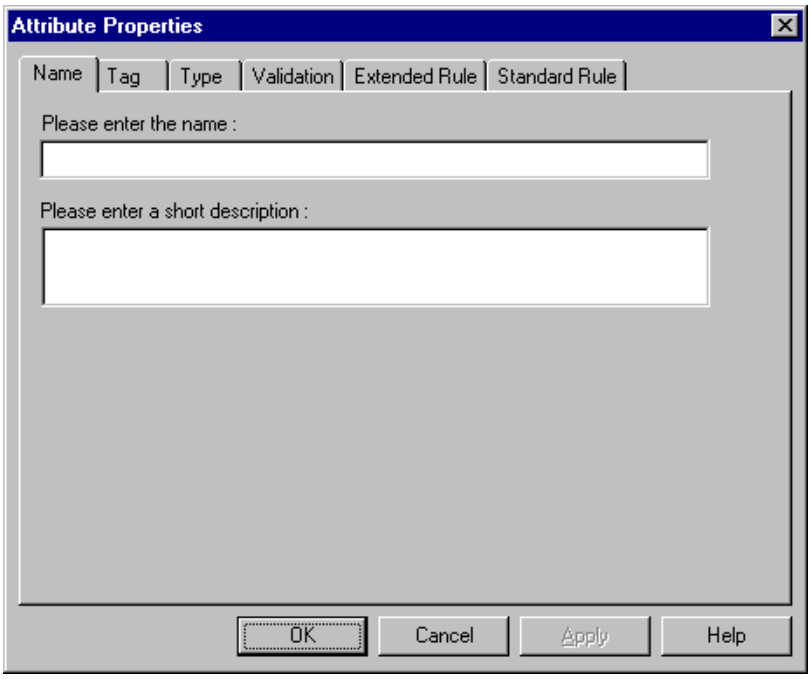
The attribute container object does not correspond to an XML feature. Gentran:Server uses attribute container objects to contain the attributes of an XML element, so attribute containers do not have properties. An attribute container object is automatically created when the user creates the first attribute of an XML element. Subsequent attribute objects are created in the existing attribute container object.

Attribute object

The attribute object specifies information associated with an element that further defines the element.

Attribute Properties dialog box

This diagram illustrates the Attribute Properties dialog box (Name tab).



Parts and functions

This table lists the parts of the Pcdata Properties dialog box and their functions.

<table>
<tr><th>Part</th><th colspan="2">Function</th></tr>
<tr><td colspan="3">Name tab</td></tr>
<tr><td>Name</td><td colspan="2">Defines the name of the attribute.

Note
Do not use spaces or dashes (-) for the record name. You can use the underscore (_) to separate words.</td></tr>
<tr><td>Description</td><td colspan="2">Describes the attribute. This box is used to provide a brief explanation of the attribute that allows you to differentiate it from similar attributes.</td></tr>
<tr><td colspan="3">Tag tab</td></tr>
<tr><td>Tag</td><td colspan="2">Defines the attribute identification tag, as it appears in the XML document.

Note
Gentran:Server validates the tag against the characters that XML allows for element names. An XML tag must start with a letter, an underscore, or a colon, followed by valid XML name characters. The system uses the attribute name by default.</td></tr>
<tr><td colspan="3">Type tab</td></tr>
<tr><td rowspan="5">Attribute Type</td><td colspan="2">Specifies the type of data that can be used in this attribute. This table describes the attribute types.</td></tr>
<tr><td>Attribute</td><td>Description</td></tr>
<tr><td>CDATA</td><td>Character data (a string of characters).</td></tr>
<tr><td>ENUMERATED</td><td>The value must match a value in the associated code list and all values in the code list must match the NMTOKEN production, as defined by the XML specification.

Note
To use an enumerated attribute, you must also create a code list and use a code list standard rule for the attribute with it set to raise a compliance error if the code is no found in the list.</td></tr>
<tr><td>ID</td><td>A valid and unique identifier.</td></tr>
<tr><td>IDREF</td><td>A reference to a unique identifier.</td></tr>
</table>

(Continued on next page)

(Contd) Part	Function	
Attribute Type (contd)	IDREFS	A list of references to unique identifiers.
	NMTOKEN	The value follows the rules specified in XML for name tokens.
	NMTOKENS	A list of name tokens.
Implied	Indicates that this attribute is optional. If no value is set, the document is still considered valid.	
Required	Indicates that this attribute is mandatory. If no value is set, the document is not valid.	
Default Exists	Indicates that a default value exists for this attribute. **Note** You must define the default value. If the incoming data does not contain a value for this attribute, Gentran:Server creates it with the default value.	
Fixed	Indicates that the default value of this attribute is fixed (cannot be changed). **Note** You must define the default value. If the incoming data does not match this value, the document is not valid.	
Default value	Specifies the default value for the attribute.	
Validation tab		
Minimum	Specifies the minimum length of the attribute.	
Maximum	Specifies the maximum length of the attribute.	
Data-type	Specifies the type of data. Valid values are: ◗ String (alphanumeric element) ◗ Number (numeric or real element) ◗ Date/Time (date or time element) (Continued on next page)	

(Contd) Part	Function
Format	Specifies how the attribute is formatted. **Note** Depending on which Data-type you selected, you can either select the data format from a list (if you choose Number or Date/Time in the Type field), or enter a Syntax Token to denote that this field must be formatted as the specified Syntax Token dictates. When you installed Gentran:Server, you assigned a default format to the string fields. This format serves as the basis for character validation. Most U.S. users use a default format that corresponds to ASCII characters (for example, the X syntax token). Most users of Asian or European languages and encoded character sets should use the Free Format (0x01-0xFF).
Extended Rule tab	
Full Screen	Maximizes the dialog box.
Compile	Compiles the extended rule. Any warnings or errors are displayed in the Errors list. **Note** This function gives you immediate feedback about the accuracy of your rule. The rule is compiled when you compile the entire translation object.
Extended rule	Defines the extended rule.
Errors	Displays any errors generated when you clicked **Compile** to compile the extended rule.
Standard Rule tab	
Standard rule	Specifies a standard rule that will affect this field or element during processing. Each of the rules are mutually exclusive. **Reference** See the *Application Integration User's Guide* for more information on standard rules.

Creating an attribute

Use this procedure to create an attribute.

Step	Action
1	Right-click a **map object** and select either **Create Sub** or **Insert** from the shortcut menu. **Reference** See *Overview: Creating Map Objects* on page 2 - 15 for more information on the Create Sub and Insert functions.
2	From the shortcut menu, select **Attribute**. **System response** The system displays the Attribute Properties dialog box.
3	On the Name tab, specify the following: ◗ unique attribute name ◗ description (if applicable)
4	If necessary, select the **Tag** tab and change the value in the Tag box. **Note** This must match the attribute tag in the XML document.
5	Select the **Type** tab to access the attribute type options.
6	On the Type tab, specify the following: ◗ attribute type, ◗ default usage of the attribute, and ◗ default value (only if you selected Default Exists or Fixed). **Note** To use an enumerated attribute, you must also create a code list and use a code list standard rule with the attribute.
7	Select the **Validation** tab to access the validation options.
8	On the Validation tab, specify the following: ◗ minimum length ◗ maximum length ◗ type of data ◗ how the data is formatted (Continued on next page)

(Contd) Step	Action
9	Do you want to specify an extended rule for this attribute? ◗ If *yes*, select the **Extended Rule** tab, define the rule, and continue with step 10. **Reference** See the *Application Integration User's Guide* for more information on extended rules. ◗ If *no*, continue with step 11.
10	Do you want to specify a standard rule for this attribute? ◗ If *yes*, select the **Standard Rule** tab, define the rule, and continue with step 11. **Reference** See the *Application Integration User's Guide* for more information on standard rules. ◗ If *no*, continue with step 11.
11	Click **OK**. **System response** The system saves the attribute and closes the Attribute Properties dialog box.

APPENDIX

A

XML Build and Break Maps

Contents

Introduction

Overview

Using the Gentran:Server for Windows XML feature may require you to create customized build and break maps that are specific to your XML document. Depending on your XML layout (i.e., if you use many doctypes), you may need to create a set of build and break maps for each doctype.

Note
We provide you with tutorial XML build and break maps and translation objects in the GENSRVNT\Tutorial\XML folder. You will not be able to use these translation objects without significant modification to accommodate your configuration. If you do modify these maps, to use the translation objects you must register them with Gentran:Server and then select the build and break translation objects for the splitter entry you configured for XML (and change the Start tag to PETTEST_INVOICE).

Reference

- See How to Register a New Translation Object in the *Gentran:Server User's Guide* for more information on registering translation objects.
- See *Modifying the Splitter Configuration* on page 2 - 6 for more information on setting up an XML Splitter entry.

Use of wildcard segments in build and break maps

In Gentran:Server, build and break maps generally use wildcard segments to pass data through the build and break maps anonymously—one segment at a time without compliance checking or parsing data into fields. However, wildcard processing does not work with XML and therefore you need to use readblock, writeblock, fseek, and ftell extended rules to pass the data through the maps.

Reference
See step 12 on page 10 for an example of using these extended rules.

All XML data is converted to Unicode before it is processed by the system, which affects the manner in which the ReadBlock and WriteBlock extended rule functions transfer the data. A character in Unicode consists of 2-bytes, so when defining your string variable to contain the block of data read and written using ReadBlock/WriteBlock, you must define it as double the maximum character length you could receive for an XML Block of Data.

A Block of data for XML data consists of the following:

1. The XML Start Tag and any attributes for that Start tag
2. PCDATA
3. The XML End Tag

Use of wildcard segments in build and break maps (contd.)

Example

```
<Special Instructions> XML data is converted into Unicode before
processing
</Special Instructions>
```

The translator reads the above XML as three separate blocks of data:

1. The Start tag: <Special Instructions>
2. The PCDATA: XML data is converted into Unicode before processing
3. The End tag: </Special Instructions>

In the scenario, the PCDATA is 52 characters in length. Thus, when translator processes this section of data it actually occupies 104 bytes. Therefore, the string variable into which this value is read (using the ReadBlock and WriteBlock functions) must be at least 104 characters to receive the entire string.

Reference

See Using Extended Rules in the *Application Integration User's Guide* for more information on extended rules.

In this appendix

This appendix provides general guidelines on how to create build and break maps for translating XML documents.

Note

This appendix is not intended to detail step-by-step instructions on how to complete XML build and break maps. Each set of XML build and break maps can be different because the structure of every XML document is unique.

XML Build Maps

Overview

In this section

This section describes general guidelines for creating XML build maps. It is not intended to detail step-by-step instructions because each XML build map can be different because the structure of every XML document is unique.

Reference
See *Creating an XML Map* on page 2 - 8 for more information on creating maps. See *Create two maps* on page A - 6 for more information on the specific information necessary to create build maps.

What is a build map?

Typically, a build map is used to generate envelope segments around outbound data. There are three different levels of build maps: interchange build, functional group, build, and transaction build. You will only use functional group build maps if your XML structure contains data that you want to be mapped to the Gentran:Server group table.

Notes

- Build maps and the build process are required even if you do not require data enveloping.
- Depending on the structure of your XML document, you may not require an XML group build map.
- Update standard rules can only be used for build maps (Interchange, Group, or Transaction Set), and they are only valid on the Input side of the map. The Interchange_tb database table is accessed in an Interchange build map, the Group_tb is accessed in a Group build map, and the Document_tb is accessed in a Transaction Set build map.

Pass-through vs. enveloping build maps

This table lists the difference between pass-through build maps (also referred to as no-build maps) and enveloping build maps.

Pass-Through Build Maps	Enveloping Build Maps
The XML data created by your import map does *not* require enveloping. **Example** You use pass-through build maps if the XML structure created by the import map is complete and does not require the addition of any further XML elements.	The XML data created by your import map requires enveloping. **Example** You use enveloping build maps if the XML structure created by the import map is not complete.

Outbound processing stages

This table describes the stages of outbound processing.

Stage	Description
1	The system determines which system import map to use. **References** ◗ See Setting Up the Import Process in the *Application Integration User's Guide* for more information on creating a system import map. ◗ See How to Define a New Import Specification in the *Administration Guide* for more information on setting up an XML import specification.
2	The system locates the partner and relationship, checks the data for compliance, and translates the import data file. **Reference** See How to Create a New Outbound Relationship in the *Gentran:Server User's Guide* for more information on how to set up an outbound partner relationship.
3	The system builds the outbound interchanges using data from the outbound partner relationship. **Reference** See How to Create a New Outbound Relationship in the *Gentran:Server User's Guide* for more information on how to set up an outbound partner relationship.

Pass-Through Build (No Build) Maps

Introduction

If the XML data created by your import map does *not* require enveloping (because the XML structure created by the import map is complete and does not require the addition of any other XML elements), you need to create a pass-through build map (also known as a "no build" map). This section contains guidelines to create a pass-through build map to pass the XML data through the map without creating envelopes.

Create two maps

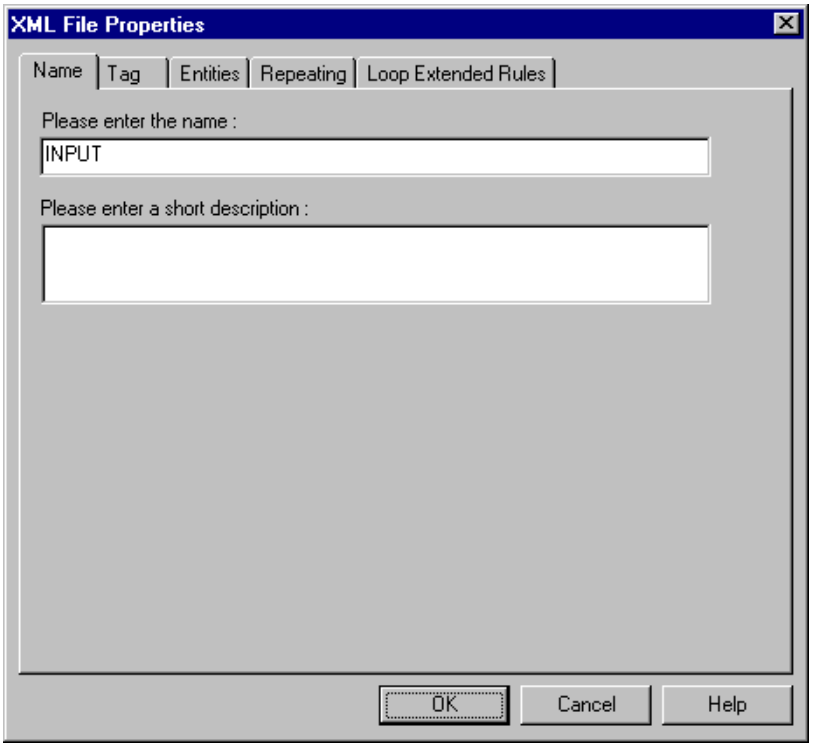
You need to create at least two maps—a transaction build map and an interchange build map. If your XML document structure contains information that you want to map to the Gentran:Server group table, you need to create a functional group build map as well.

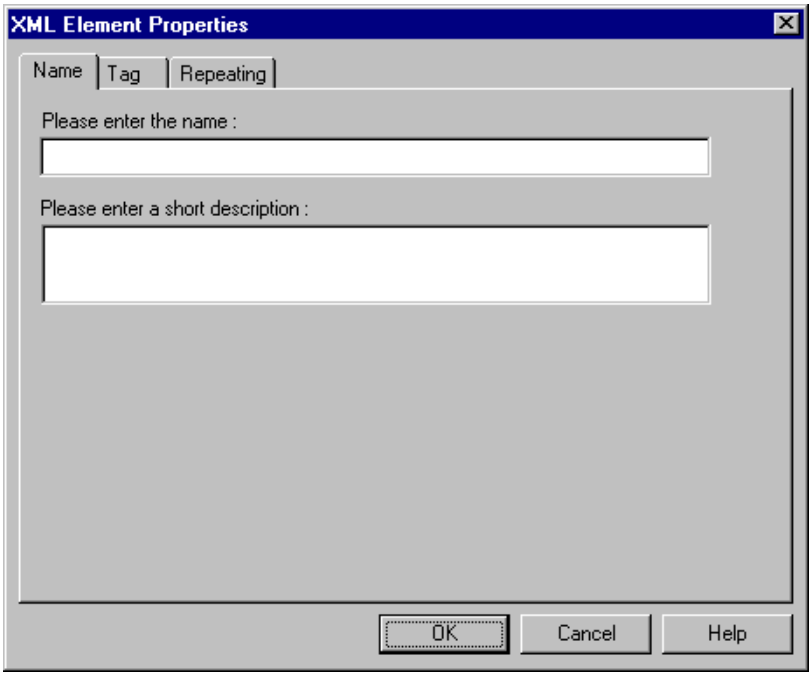
This table lists the specific details of the two maps you need to create.

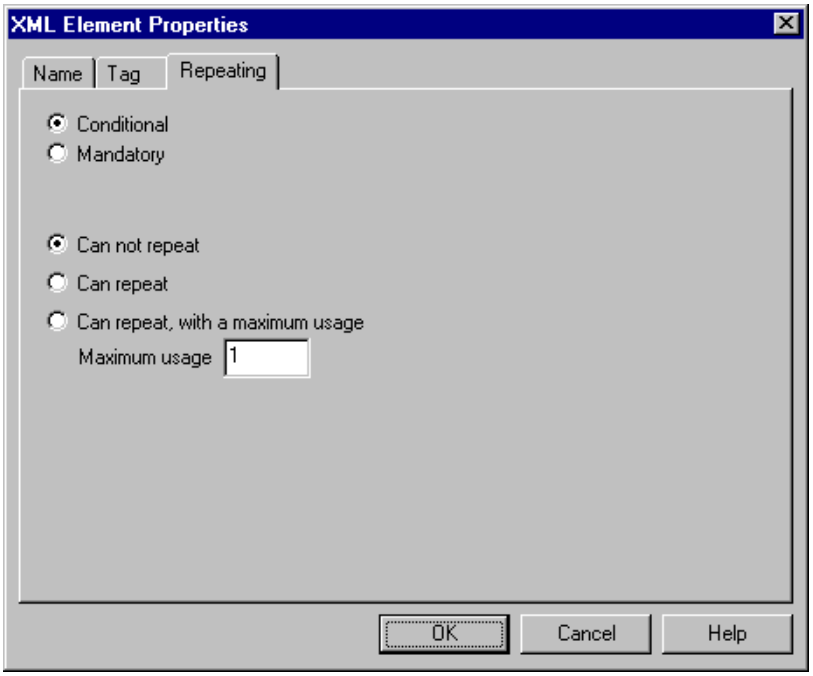
IF you are creating this type of map...	**THEN make these selections when creating the map...**
transaction build map	◗ Map type = Transaction Build ◗ Input format = XML ◗ Output format = XML
interchange build map	◗ Map type = Interchange Build ◗ Input format = XML ◗ Output format = XML

Modify the INPUT side of each map

Use this procedure to modify the INPUT side of each map you created.

Step	Action
1	Right-click the XML File icon on the INPUT side of the map and select Properties from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default). 
2	In the name box, type the name of the Root Element in your XML document.
3	Select the Tag tab. **System response** The system displays the tag options.
4	In the Tag box, type the tag of the Root Element in your XML document.
5	Click **OK**. **System response** The system saves your changes and closes the XML File Properties dialog box. (Continued on next page)

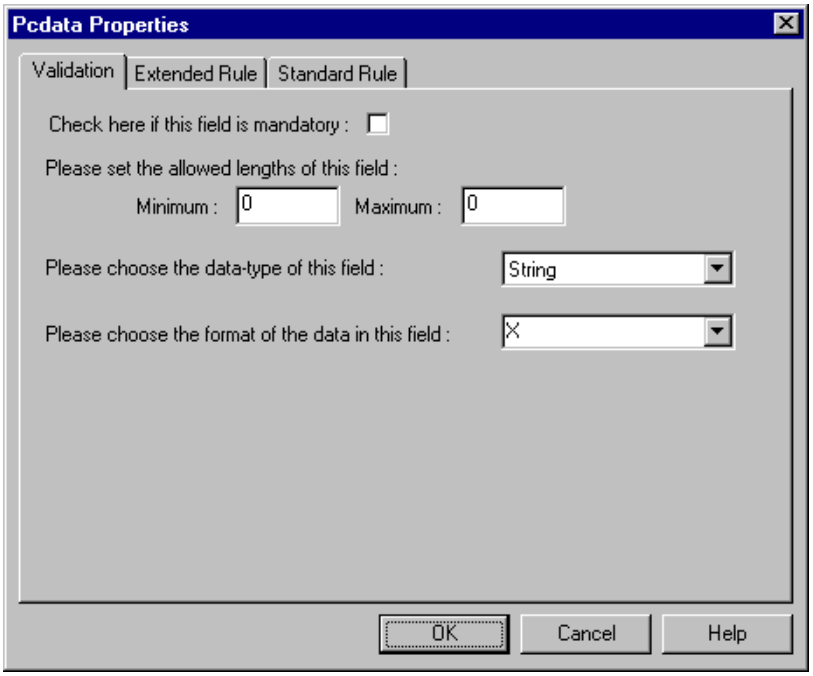
(Contd) Step	Action
6	Right-click the XML File icon and select Create Sub from the shortcut menu. **Reference** See *Overview: Creating Map Objects* on page 2 - 15 for more information on the Create Sub and Insert functions.
7	From the shortcut menu, select Element. **System response** The system displays the XML Element Properties dialog box (Name tab). 
8	In the name box, type the name of the first Element in your XML document that is passed to the build maps. (Continued on next page)

(Contd) Step	Action
9	Select the Repeating tab. **System response** The system displays the repeating options. 
10	Select the Can repeat option. **System response** The system appends the Loop Extended Rules tab to the XML Properties dialog box.
11	Select the Loop Extended Rules tab. **System response** The system displays the extended rule options. (Continued on next page)

(Contd) Step	Action

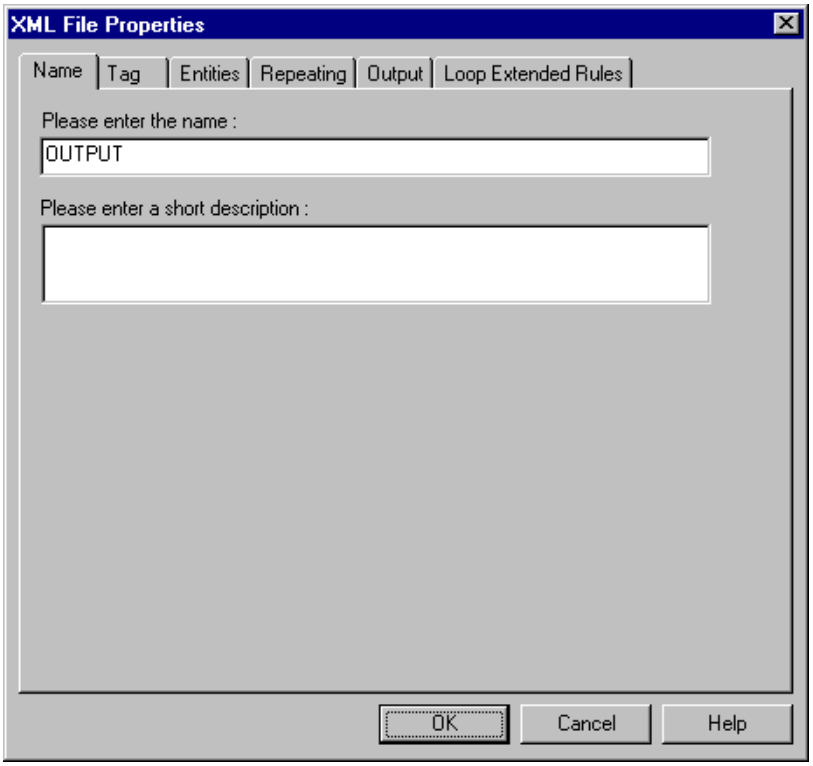
12	In the extended rule box,
	◗ accept the default On Begin option,
	◗ type the readblock/writeblock extended rule, and
	◗ click **OK**.
	System response
	The system saves your change and closes the XML Element Properties dialog box.
	Example
	`STRING [1024] buffer; //declares a string variable buffer`
	`buffer=""; //initialize the buffer variable to null`
	
	`fseek(0, 0, Begin); //seek to beginning of file to begin`
	`//writing at the beginning`
	`While readblock(buffer) do`
	`writeblock(buffer);`
	Reference
	See Using Extended Rules in the *Application Integration User's Guide* for more information on extended rules

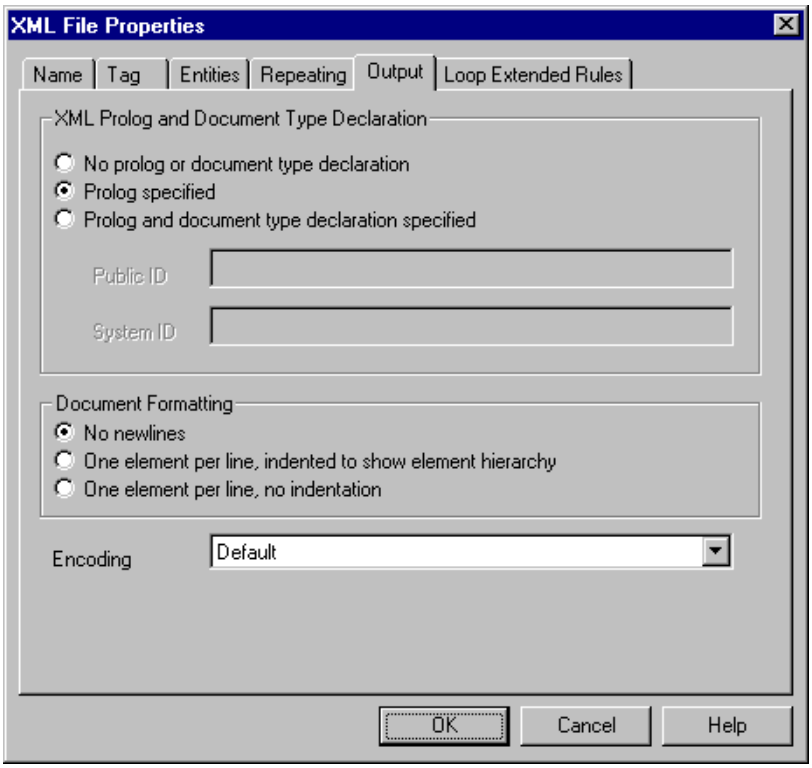
13	Right-click the element you created in Step 7, and select Create Sub from the shortcut menu. (Continued on next page)

(Contd) Step	Action
14	From the shortcut menu, select Pcdata. **System response** The system displays the Pcdata Properties dialog box (Validation tab). 
15	Select the appropriate options based on your XML document and click **OK**. **System response** The system saves your change and closes the Pcdata Properties dialog box.

Modify the OUTPUT side of each map

For each map you created above, you also need to modify the OUTPUT side. Use this procedure to modify each map.

Step	Action
1	Right-click the XML File icon on the OUTPUT side of the map and select Properties from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default).  **Reference** See *XML File Properties parts and functions* on page 2 - 18 for more information on the dialog box parts and functions.
2	In the name box, type the name of the Root Element in your XML document.
3	Select the Tag tab. **System response** The system displays the tag options.
4	In the Tag box, type the tag of the Root Element in your XML document. (Continued on next page)

(Contd) Step	Action
5	Select the Repeating tab. **System response** The system displays the tag options.
6	Select the Conditional option. **Note** You must make the OUTPUT File object conditional because there are no physical links to the output side of the map. And, if the OUTPUT File object was mandatory, the absence of physical links would generate an error.
7	Select the Output tab. **System response** The system displays the output options.  (Continued on next page)

<table>
<tr><th>(Contd) Step</th><th colspan="2">Action</th></tr>
<tr><td>8</td><td colspan="2">Use this table to determine the next step.</td></tr>
<tr><td></td><td>IF the map is...</td><td>THEN...</td></tr>
<tr><td></td><td>a transaction build map</td><td>◗ In the XML Prolog and Document Type Declaration section, select the No prolog option.

Note
You choose this option because the interchange build map runs after the transaction build map and will reformat the document.

◗ In the Document Formatting section, select the No newlines option.</td></tr>
<tr><td></td><td>an interchange build map</td><td>◗ In the XML Prolog and Document Type Declaration section, select the appropriate option based on your XML specifications.
◗ In the Document Formatting section, select the appropriate option to format the XML data to your specifications.</td></tr>
<tr><td>9</td><td colspan="2">Click OK.

Reference
The system saves your changes and closes the XML File Properties dialog box.</td></tr>
<tr><td>10</td><td colspan="2">Right-click the XML File icon and select Create Sub from the shortcut menu.

Note
You need to create a temporary XML element and pcdata so the map will compile correctly.

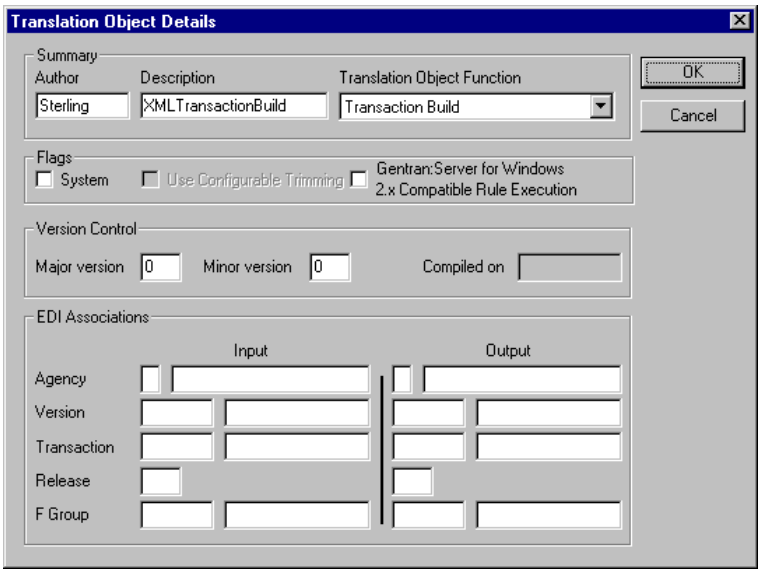
Reference
See Overview: Creating Map Objects on page 2 - 15 for more information on the Create Sub and Insert functions.

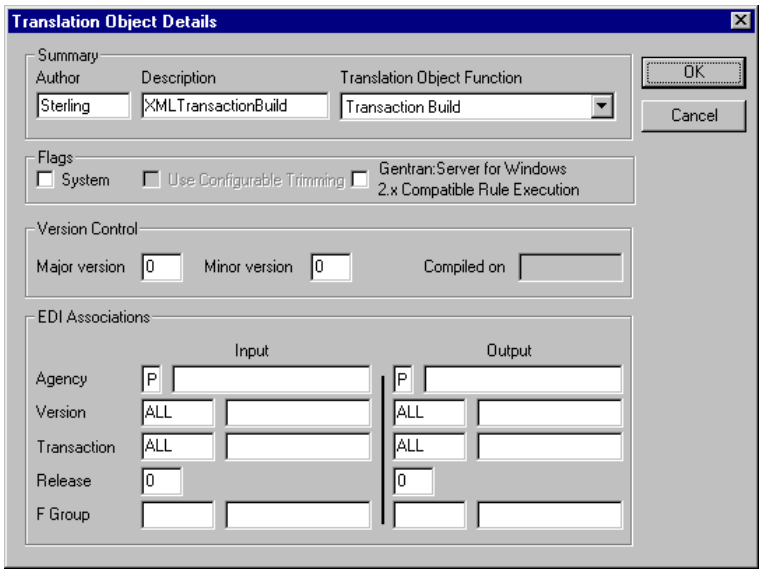
(Continued on next page)</td></tr>
</table>

(Contd) Step	Action
11	From the shortcut menu, select Element. **System response** The system displays the XML Element Properties dialog box (Name tab).
12	In the name box, type TEMP.
13	Click **OK**. **System response** The system saves the temporary element.
14	Right-click the element you created in Step 10 and select Create Sub from the shortcut menu.
15	From the shortcut menu, select Pcdata. **System response** The system displays the Pcdata Properties dialog box (Validation tab).
16	Click **OK**. **System response** The system closes the Pcdata Properties dialog box.

Modify the map details for each map

For each map you created, you also need to modify the Translation Object Details dialog box. Use this procedure to modify each map.

Step	Action
1	From the Edit menu, select Details. **System response** The system displays the Translation Object Details dialog box. 
2	In the EDI Associations section, type a letter that is not currently used to designate a standards agency (e.g., P) in the first Agency box for both Input and Output. **Notes** ◗ The information you enter must match your import specifications. ◗ Do not use X, U, or E in the Agency box. These are reserved for the ANSI X-12, UCS, and EDIFACT standards, respectively, and thus will cause the map to select incorrect Partner Editor build objects.
3	In the EDI Associations section, type ALL in the first Version box for both Input and Output. (Continued on next page)

(Contd) Step	Action
4	In the EDI Associations section, type ALL in the first Transaction box for both Input and Output. **System response** This diagram illustrates how the EDI Associations section of the Transaction Object Detail dialog box should look. 
5	Click **OK**. **System response** The system saves your changes and closes the Translation Object Details dialog box.

Enveloping Build Maps

Introduction

If the XML data created by your import map requires enveloping (the XML structure created by the import map is not complete), you need to create an enveloping build map to create envelopes around the XML data. This process is more advanced than the process of creating pass-through break maps, therefore this section contains guidelines (not step-by-step instructions) on creating enveloping build maps, which build envelopes around the XML data.

Note
You can only update the Gentran:Server database tables at the current level of processing (e.g., you can only update the Interchange_tb with an Interchange Break or Build Map).

Create two maps

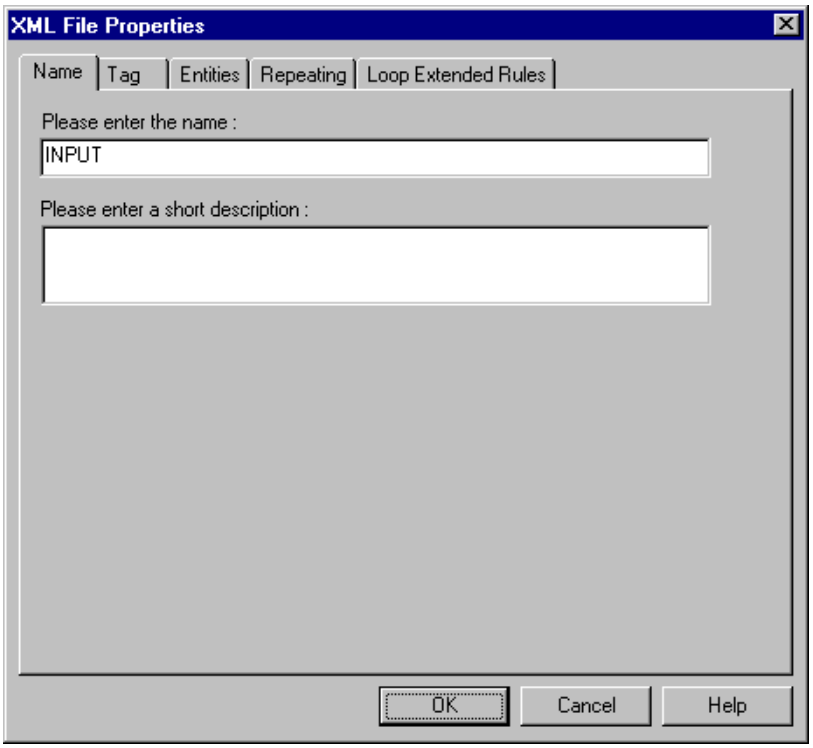
You need to create at least two maps—a transaction build map and an interchange build map. If your XML document structure contains information that you want to map to the Gentran:Server group table, you need to create a functional group build map as well.

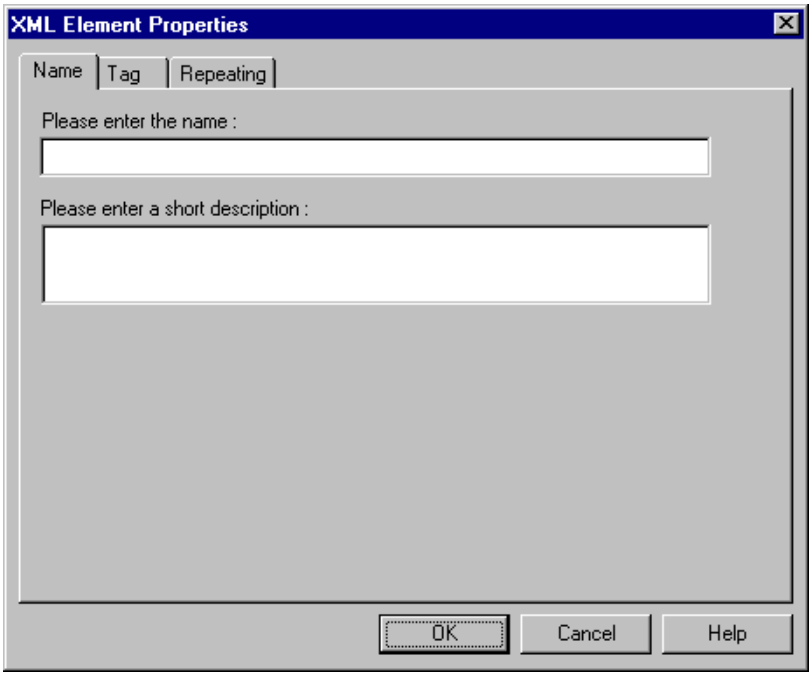
This table lists the specific details of the two maps you need to create.

IF you are creating this type of map...	THEN make these selections when creating the map...
transaction build map	◗ Map type = Transaction Build ◗ Input format = XML ◗ Output format = XML
interchange build map	◗ Map type = Interchange Build ◗ Input format = XML ◗ Output format = XML

Modify the INPUT side of each map

For each map you created above, you need to modify the INPUT side. Use this procedure to modify each map.

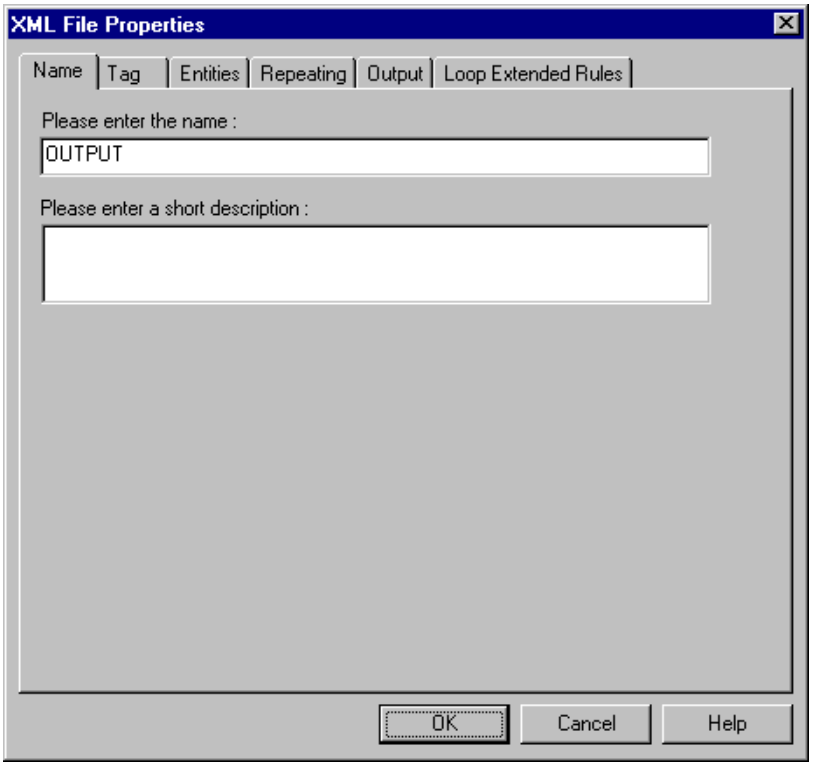
Step	Action
1	Right-click the XML File icon on the INPUT side of the map and select Properties from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default). 
2	In the name box, type the name of the Root Element in your XML document.
3	Select the Tag tab. **System response** The system displays the tag options.
4	In the Tag box, type the tag of the Root Element in your XML document. (Continued on next page)

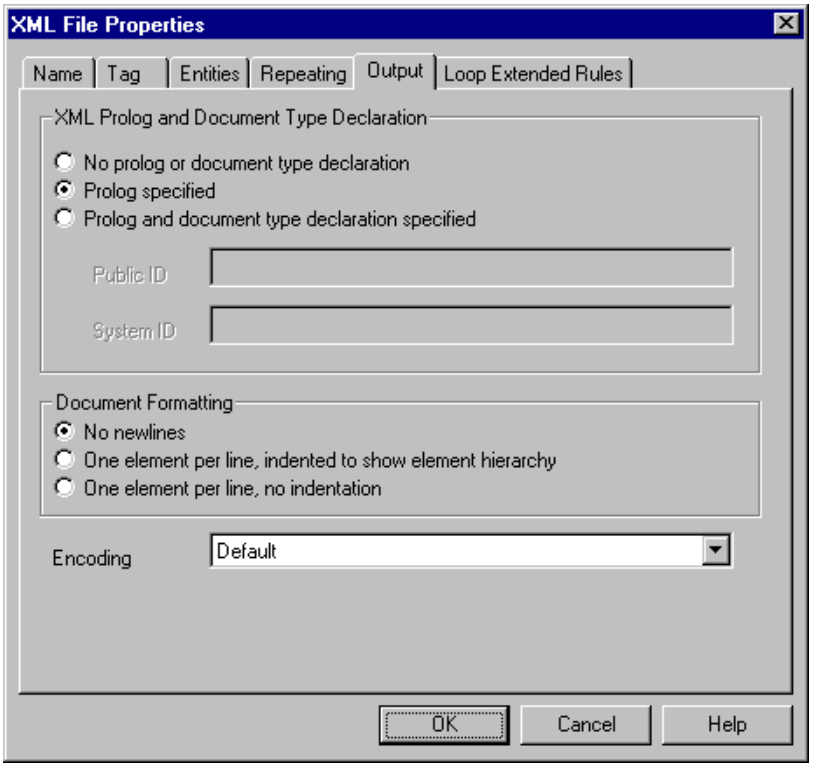
(Contd) Step	Action
5	Click **OK**. **System response** The system saves your changes and closes the XML File Properties dialog box.
6	Right-click the XML File icon and select Create Sub from the shortcut menu. **Reference** See *Overview: Creating Map Objects* on page 2 - 15 for more information on the Create Sub and Insert functions.
7	From the shortcut menu, select Element. **System response** The system displays the XML Element Properties dialog box (Name tab). 
8	In the name box, type the name of the first Element in your XML document that is passed to the build maps.
9	Click **OK**. **System response** The system saves your change and closes the XML Element Properties dialog box. (Continued on next page)

(Contd) Step	Action
10	Right-click the element you created in Step 7, and select Create Sub.
11	From the shortcut menu, select Pcdata. **System response** The system displays the Pcdata Properties dialog box (Validation tab).
12	Select the appropriate options based on your XML document and click **OK**. **System response** The system saves your change and closes the Pcdata Properties dialog box.

Modify the OUTPUT side of each map

For each map you created above, you also need to modify the OUTPUT side. Use this procedure to modify each map.

Step	Action
1	Right-click the XML File icon on the OUTPUT side of the map and select Properties from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default). 
2	In the name box, type the name of the Root Element in your XML document.
3	Select the Tag tab. **System response** The system displays the tag options.
4	In the Tag box, type the tag of the Root Element in your XML document. (Continued on next page)

(Contd) Step	Action
5	Select the Output tab. **System response** The system displays the output options.  (Continued on next page)

<table>
<tr><th>(Contd) Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="4">6</td><td colspan="2">Use this table to determine the next step.</td></tr>
<tr><td>IF the map is...</td><td>THEN...</td></tr>
<tr><td>a transaction build map</td><td>◗ In the XML Prolog and Document Type Declaration section, select the No prolog option.

Note
You choose this option because the interchange build map runs after the transaction build map and will reformat the document.

◗ In the Document Formatting section, select the No newlines option.</td></tr>
<tr><td>an interchange build map</td><td>◗ In the XML Prolog and Document Type Declaration section, select the appropriate option based on your XML specifications.
◗ In the Document Formatting section, select the appropriate option to format the XML data to your specifications.</td></tr>
<tr><td>7</td><td colspan="2">Click OK.

System response
The system saves your changes and closes the XML File Properties dialog box.</td></tr>
<tr><td>8</td><td colspan="2">On the OUTPUT side of the map, create the necessary header and trailer envelope structure.

Reference
See the Application Integration User's Guide for information on specific tasks.

(Continued on next page)</td></tr>
</table>

(Contd) Step	Action
9	Assign the necessary data values to the structure you created in the previous step. **Note** You can perform this task by using constants or selecting from the GenericEnvelopeSegment_tb database table. **Reference** See the *Application Integration User's Guide* for information on specific tasks.
10	Link the pcdata you created in *Modify the INPUT side of each map* on page A - 19 to every map component on the OUTPUT side of the map. **Note** The INPUT pcdata must always contain data to produce the map components to which it is linked on the OUTPUT side of the map.
11	Right-click the first envelope trailer element and select Properties from the shortcut menu. **System response** The system displays the XML Element Properties dialog box (Name tab).
12	Does the XML Element Properties dialog box contain a Loop Extended Rules tab? ◗ If *yes*, continue with Step 14. ◗ If *no*, select the Repeating tab and continue with the next step. **System response** The system displays the repeating options.
13	Verify that the Can repeat option is selected. **System response** The system appends the Loop Extended Rules tab to the XML Properties dialog box.
14	Select the Loop Extended Rules tab. **System response** The system displays the extended rule options. (Continued on next page)

(Contd) Step	Action

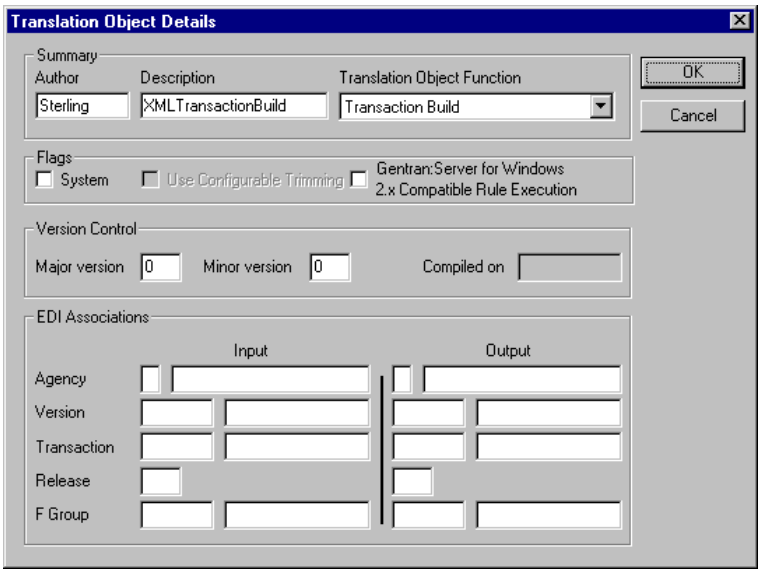
15	In the extended rule box,
	◗ accept the default On Begin option and
	◗ type the readblock/writeblock extended rule.
	Example
	`STRING [1024] buffer; //declares a string variable buffer`
	`buffer=""; //initialize the buffer variable to null`
	
	`fseek(0, 0, Begin); //seek to beginning of file to begin`
	`//writing at the beginning`
	`While readblock(buffer) do`
	`writeblock(buffer);`
	Reference
	See Using Extended Rules in the *Application Integration User's Guide* for more information on extended rules.

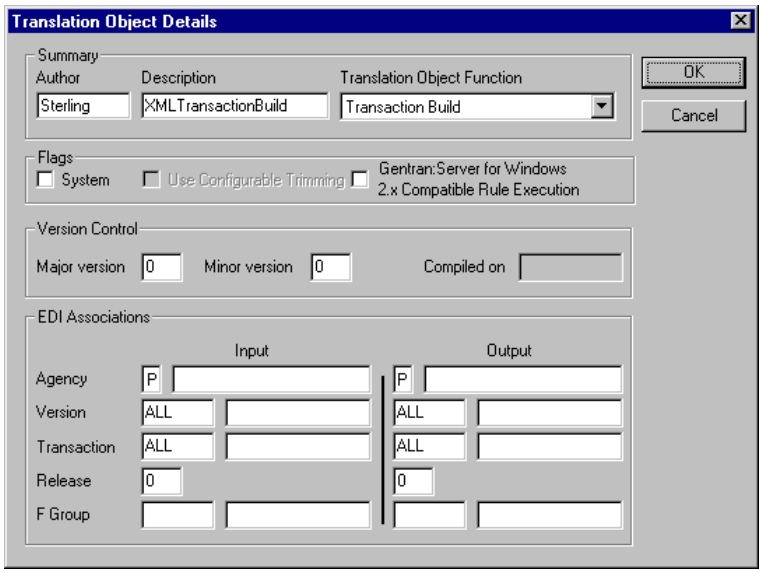
16	Click **OK**. **System response** The system saves your change and closes the XML Element Properties dialog box.
17	Right-click the XML File icon and select Create Sub from the shortcut menu. **Note** You need to create a temporary XML element and pcdata so the map will compile correctly. **Reference** See *Overview: Creating Map Objects* on page 2 - 15 for more information on the Create Sub and Insert functions.
18	From the shortcut menu, select Element. **System response** The system displays the XML Element Properties dialog box (Name tab).
19	In the name box, type TEMP.
20	Click **OK**. **System response** The system saves the temporary element.
21	Right-click the element you created in Step 17 and select Create Sub from the shortcut menu. (Continued on next page)

(Contd) Step	Action
22	From the shortcut menu, select Pcdata. **System response** The system displays the Pcdata Properties dialog box (Validation tab).
23	Click **OK**. **System response** The system closes the Pcdata Properties dialog box.

Modify the map details for each map

For each map you created above, you also need to modify the Translation Object Details dialog box. Use this procedure to modify each map.

Step	Action
1	From the Edit menu, select Details. **System response** The system displays the Translation Object Details dialog box. 
2	In the EDI Associations section, type a letter that is not currently used to designate a standards agency (e.g., P) in the first Agency box for both Input and Output. **Note** The information you enter must match your import specifications.
3	In the EDI Associations section, type ALL in the first Version box for both Input and Output. (Continued on next page)

(Contd) Step	Action
4	In the EDI Associations section, type ALL in the first Transaction box for both Input and Output. **System response** This diagram illustrates how the EDI Associations section of the Transaction Object Detail dialog box should look. 
5	Click **OK**. **System response** The system saves your changes and closes the Translation Object Details dialog box.

XML Break Maps

Overview

In this section

This section describes general guidelines for creating XML break maps. It is not intended to detail step-by-step instructions because each XML break map is different because the structure of every XML document is unique.

Reference
See *Creating an XML Map* on page 2 - 8 for more information on creating maps. See *Create two maps* on page A - 32 for more information on the specific information necessary to create break maps.

What is a break map?

Break maps are used to remove envelopes from an interchange during break processing. They are also used to assist the translator in collecting all information necessary to find an inbound partner relationship.

There are three different levels of break maps: interchange break, functional group break, and transaction break. You will only use functional group build maps if your XML structure contains data that you want to be mapped to the Gentran:Server group table.

Note
You will only use functional group build maps if your XML structure contains data that you want to be mapped to the Gentran:Server group table.

Pass-through vs. non-pass-through break maps

This table lists the difference between pass-through break maps and non-pass-through break maps.

Pass-Through Break Maps	Non-Pass-Through Break Maps
Your export map requires the entire XML document structure to be passed through the map. **Example** You use pass-through break maps when you intend to map your entire XML structure in your export map, and thus nothing should be removed from the data before it reaches the export map.	Your export map does *not* require the entire XML document structure to be passed through the map. **Example** You use non-pass-through break maps when you receive an XML document that does not require the entire XML structure.

Inbound processing stages

This table describes the stages of inbound processing.

Stage	Description
1	**The Split Process** The inbound data is processed through the Gentran:Server splitter, where all interchanges are validated and split into separate files. **Reference** See *Modifying the Splitter Configuration* on page 2 - 6 for more information on how to set up XML splitter specifications.
2	**The Break Process** The translator uses break maps to process the interchange data, locate the correct partner, locate the inbound relationship, and create document files. **Reference** See How to Create a New Inbound Relationship in the *Gentran:Server User's Guide* for more information on how to set up an inbound partner relationship.
3	**The Export Process** The translator uses the export map defined in the inbound relationship to translate data from one format to another. **References** ◗ See How to Create a New Inbound Relationship in the *Gentran:Server User's Guide* for more information on how to set up an inbound partner relationship, including automatic export options. ◗ See How to Export a Document in the *Gentran:Server User's Guide* for more information on the manual export process.

Pass-Through Break Maps

Introduction

If your export map requires the entire XML document structure to be passed through the map (you intend to map your entire XML structure in your export map, and thus nothing should be removed from the data before it reaches the export map), you need to create a pass-through break map. This section contains guidelines to create a pass-through break map to pass the entire XML document structure through the map.

You need to create at least two maps—a transaction break map and an interchange break map. If your XML document structure contains information that you want to map to the Gentran:Server group table, you need to create a functional group break map as well.

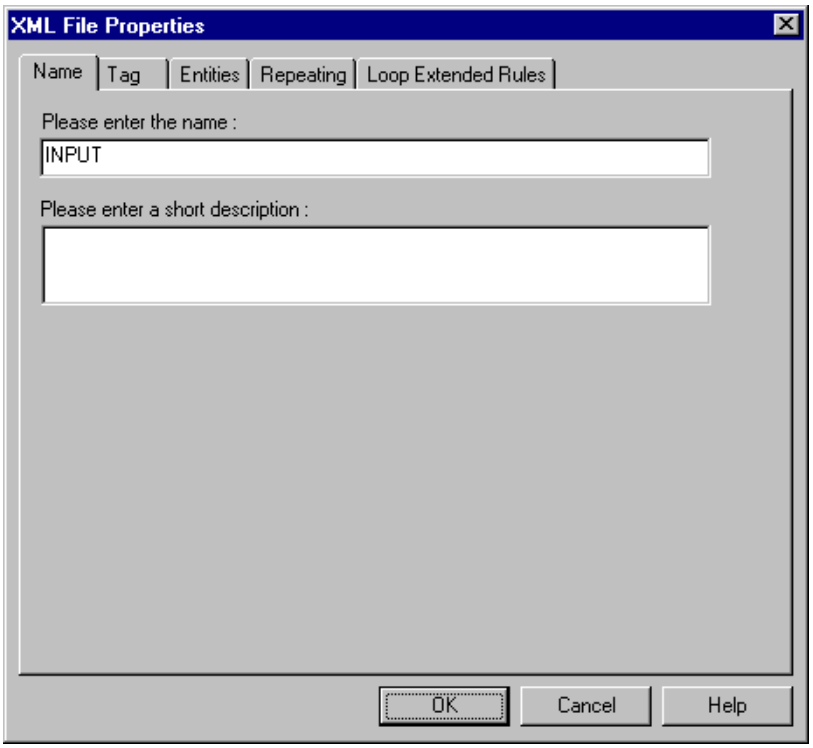
Create two maps

This table lists the specific details of the two maps you need to create.

IF you are creating this type of map...	THEN make these selections when creating the map...
transaction break map	◗ Map type = Transaction Break ◗ Input format = XML ◗ Output format = XML
interchange break map	◗ Map type = Interchange Break ◗ Input format = XML ◗ Output format = XML

Modify the INPUT side of each map

For each map you created above, you need to modify the INPUT side. Use this procedure to modify each map.

Step	Action
1	Right-click the XML File icon on the INPUT side of the map and select Properties from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default). 
2	In the name box, type the name of the Root Element in your XML document.
3	Select the Tag tab. **System response** The system displays the tag options.
4	In the Tag box, type the tag of the Root Element in your XML document. (Continued on next page)

(Contd) Step	Action
5	Click **OK**. **System response** The system saves your changes and closes the XML File Properties dialog box.
6	Create the necessary XML file structure to accommodate the necessary lookup keys and database updates. **Note** You may need to create temporary storage fields and use constants to hard-code specific values. **Reference** See the *Application Integration User's Guide* for information on specific tasks. (Continued on next page)

<table>
<tr><th>(Contd) Step</th><th>Action</th></tr>
<tr><td>7</td><td>
Add the necessary lookups and database updates.

Note
This list contains some of the different lookup and update keys that you may need to use in an interchange break map (your XML map may require more or less, depending on your XML specifications):
◗ Partner Lookup
◗ Agency Update
◗ Version Update
This list contains some of the different lookup and update keys that you may need to use in a transaction break map (your XML map may require more or less, depending on your XML specifications):
◗ Transaction Set ID Update

Reference
See the Application Integration User's Guide for information on specific tasks.

Note
Depending on the type of map, you must set specific Update standard rules for each appropriate map. For example, the Interchange_tb database table is accessed in an Interchange build map, the Group_tb is accessed in a Group build map, and the Document_tb is accessed in a Transaction Set build map.

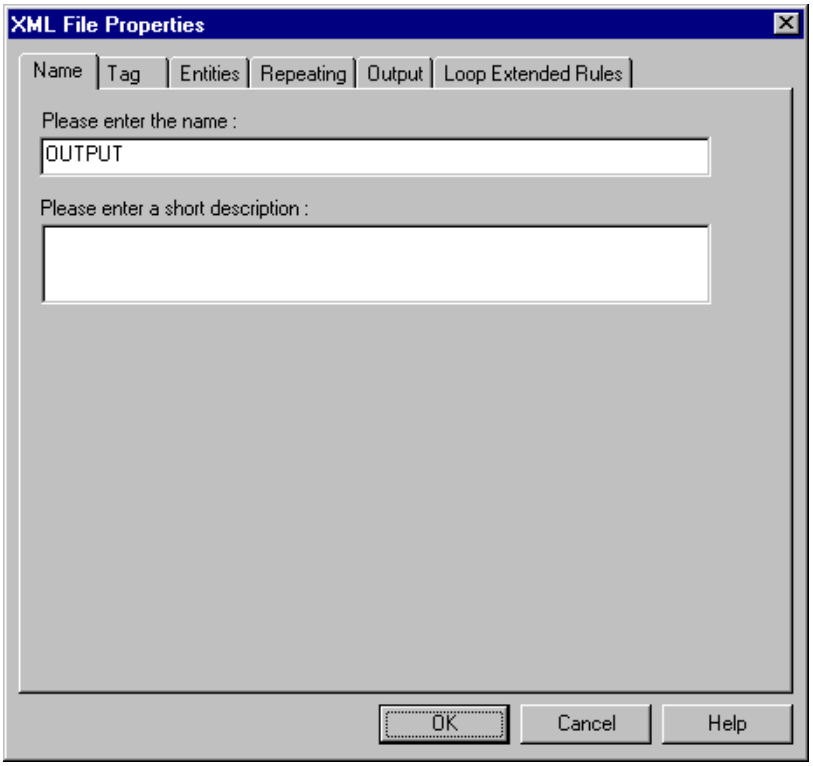
Use this table to determine where it is valid to use an Update standard rule or extended rule.
<table>
<tr><th>IF the map is of type...</th><th>THEN the update function is valid on the...</th></tr>
<tr><td>Export</td><td>Input side of the map only.</td></tr>
<tr><td>Import</td><td>Input or Output side of the map.</td></tr>
<tr><td>Break (Interchange, Group, or Transaction Set)</td><td>Input side of the map only.</td></tr>
<tr><td>Build (Interchange, Group, or Transaction Set)</td><td>Input side of the map only.</td></tr>
</table>
(Continued on next page)
</td></tr>
</table>

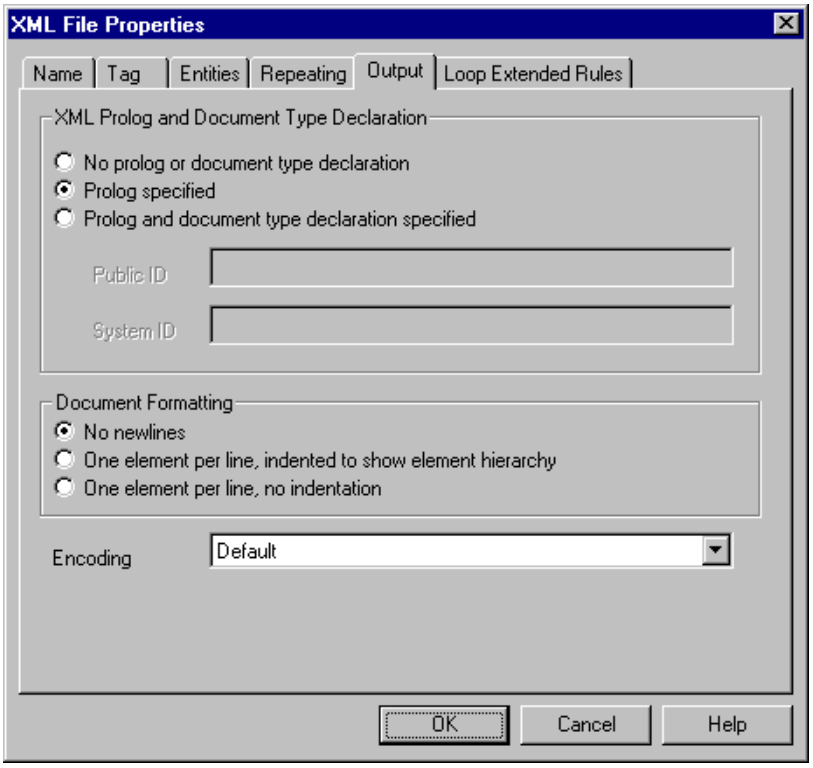
(Contd) Step	Action
8	Right-click the appropriate element in the XML file structure and select Properties from the shortcut menu. **System response** The system displays the XML Element Properties dialog box (Name tab).
9	Select the Loop Extended Rules tab. **System response** The system displays the extended rule options.

10	In the extended rule box,
	◗ accept the default On Begin option,
	◗ type the readblock/writeblock extended rule, and
	◗ click **OK**.
	
	System response
	The system saves your change and closes the XML Element Properties dialog box.
	
	Example
	`STRING [1024] buffer; //declares a string variable buffer`
	`buffer=""; //initialize the buffer variable to null`
	
	`fseek(0, 0, Begin); //seek to beginning of file to begin`
	`//writing at the beginning`
	`While readblock(buffer) do`
	`writeblock(buffer);`
	
	Reference
	See Using Extended Rules in the *Application Integration User's Guide* for more information on extended rules.

Modify the OUTPUT side of each map

For each map you created above, you also need to modify the OUTPUT side. Use this procedure to modify each map.

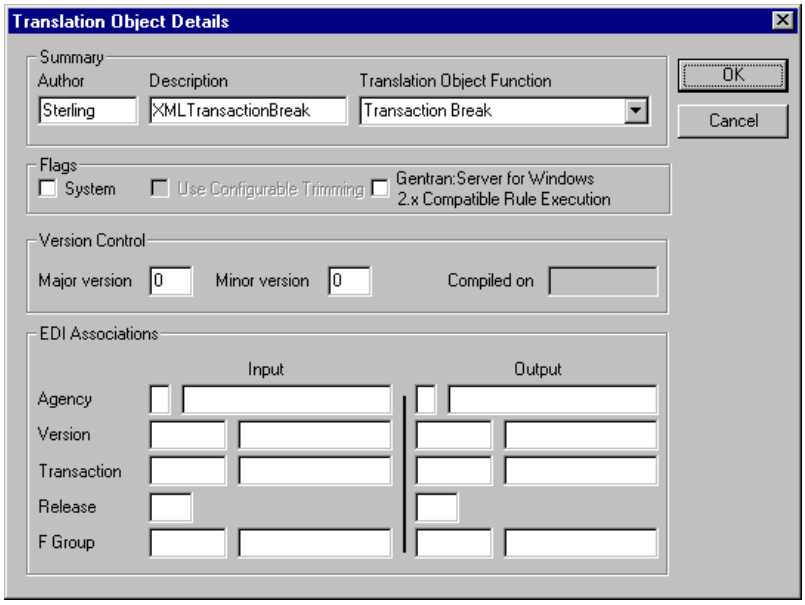
Step	Action
1	Right-click the XML File icon on the OUTPUT side of the map and select Properties from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default). 
2	In the name box, type the name of the Root Element in your XML document.
3	Select the Tag tab. **System response** The system displays the tag options.
4	In the Tag box, type the tag of the Root Element in your XML document. (Continued on next page)

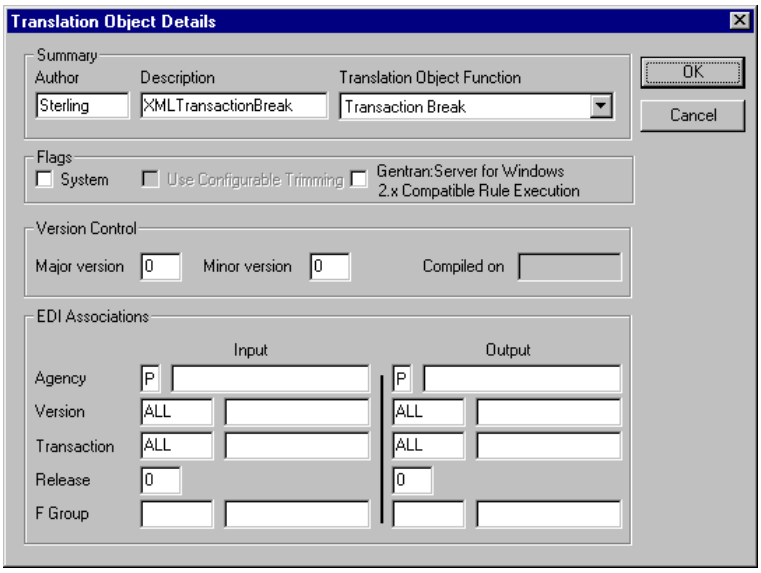
(Contd) Step	Action
5	Select the Output tab. **System response** The system displays the output options. 
6	◗ In the XML Prolog and Document Type Declaration section, select the No prolog option. ◗ In the Document Formatting section, select the No newlines option. ◗ Click **OK**. **System response** The system saves your changes and closes the XML File Properties dialog box. (Continued on next page)

(Contd) Step	Action
7	Right-click the XML File icon and select Create Sub from the shortcut menu. **Note** You need to create a temporary XML element and pcdata so the map will compile correctly. **Reference** See *Overview: Creating Map Objects* on page 2 - 15 for more information on the Create Sub and Insert functions.
8	From the shortcut menu, select Element. **System response** The system displays the XML Element Properties dialog box (Name tab).
9	In the name box, type TEMP.
10	Click **OK**. **System response** The system saves the temporary element.
11	Right-click the element you created in Step 7 and select Create Sub from the shortcut menu.
12	From the shortcut menu, select Pcdata. **System response** The system displays the Pcdata Properties dialog box (Validation tab).
13	Click **OK**. **System response** The system closes the Pcdata Properties dialog box.

Modify the map details for each map

For each map you created above, you also need to modify the Translation Object Details dialog box. Use this procedure to modify each map.

Step	Action
1	From the Edit menu, select Details. **System response** The system displays the Translation Object Details dialog box. 
2	In the EDI Associations section, type a letter that is not currently used to designate a standards agency (e.g., P) in the first Agency box for both Input and Output. **Note** The information you enter must match your import specifications.
3	In the EDI Associations section, type ALL in the first Version box for both Input and Output. (Continued on next page)

(Contd) Step	Action
4	In the EDI Associations section, type ALL in the first Transaction box for both Input and Output. **System response** This diagram illustrates how the EDI Associations section of the Transaction Object Detail dialog box should look. 
5	Click **OK**. **System response** The system saves your changes and closes the Translation Object Details dialog box.

Non-Pass-Through Break Maps

Introduction

If your export map does *not* require the entire XML document structure to be passed through the map, you need to create a non-pass-through break map to remove the XML document structure. This process is more advanced than the process of creating pass-through break maps, therefore this section contains guidelines (not step-by-step instructions) when creating a non-pass-through XML break map to remove envelope information.

Note
You can only update the Gentran:Server database tables at the current level of processing (e.g., you can only update the Interchange_tb with an Interchange Break or Build Map).

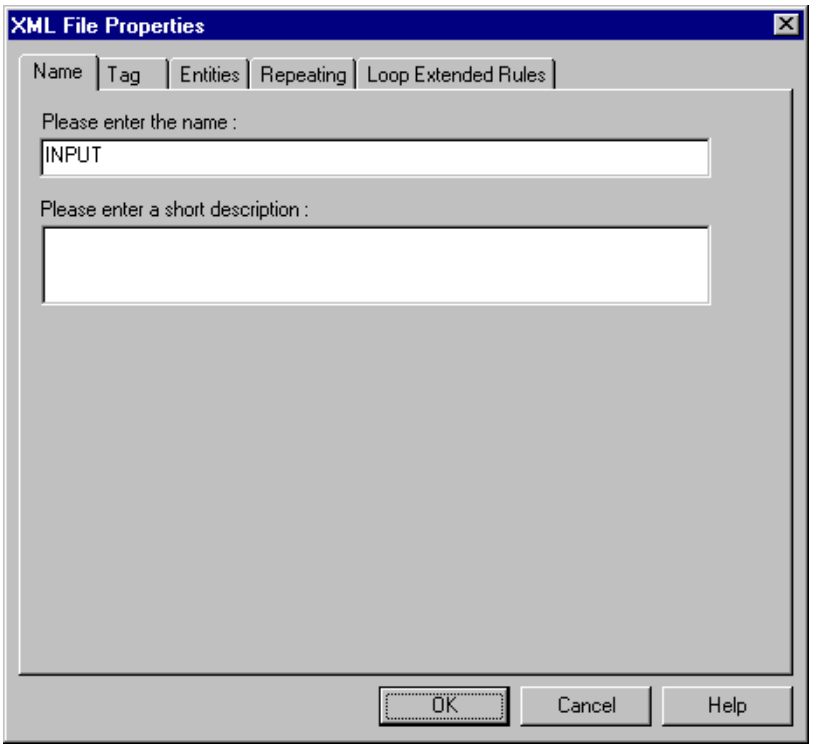
Create two maps

This table lists the specific details of the two maps you need to create.

IF you are creating this type of map...	THEN make these selections when creating the map...
transaction break map	◗ Map type = Transaction Break ◗ Input format = XML ◗ Output format = XML
interchange break map	◗ Map type = Interchange Break ◗ Input format = XML ◗ Output format = XML

Modify the INPUT side of each map

For each map you created above, you need to modify the INPUT side. Use this procedure to modify each map.

Step	Action
1	Right-click the XML File icon on the INPUT side of the map and select Properties from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default). 
2	In the name box, type the name of the Root Element in your XML document.
3	Select the Tag tab. **System response** The system displays the tag options.
4	In the Tag box, type the tag of the Root Element in your XML document. (Continued on next page)

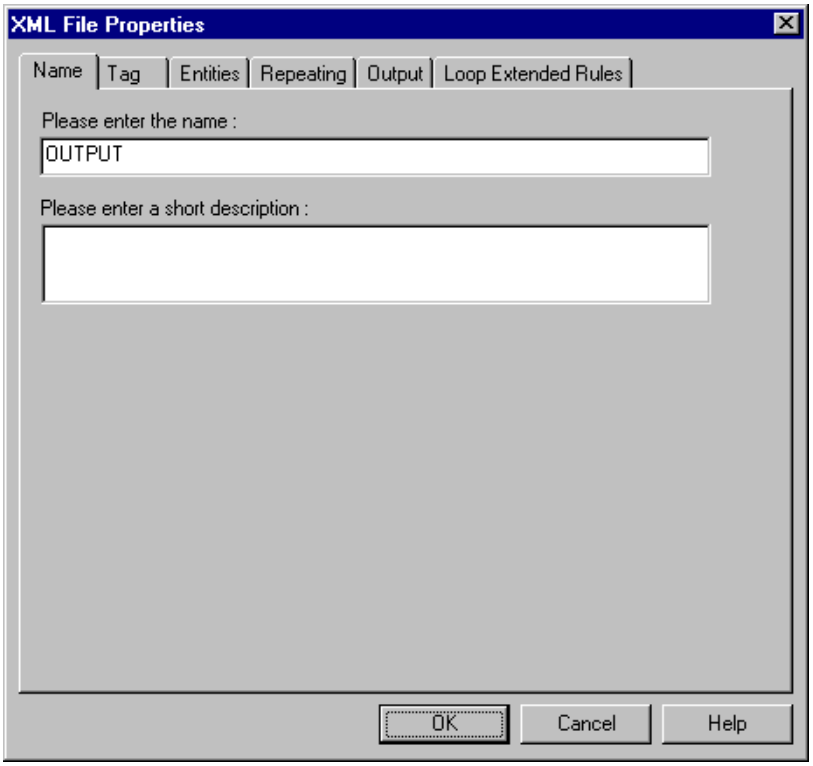
(Contd) Step	Action
5	Click **OK**. **System response** The system saves your changes and closes the XML File Properties dialog box.
6	Create the necessary XML file structure to accommodate the necessary lookup keys and database updates. **Note** You may need to create temporary storage fields and use constants to hard-code specific values. **Reference** See the *Application Integration User's Guide* for information on specific tasks.
7	Add the necessary lookups and database updates. **Note** This list contains some of the different lookup and update keys that you may need to use in an interchange break map (your XML map may require more or less, depending on your XML specifications): ◗ Partner Lookup ◗ Agency Update ◗ Version Update This list contains some of the different lookup and update keys that you may need to use in a transaction break map (your XML map may require more or less, depending on your XML specifications): ◗ Transaction Set ID Update **Reference** See the *Application Integration User's Guide* for information on specific tasks.
8	Right-click the appropriate element in the XML file structure and select Properties from the shortcut menu. **System response** The system displays the XML Element Properties dialog box (Name tab).
9	Select the Loop Extended Rules tab. **System response** The system displays the extended rule options. (Continued on next page)

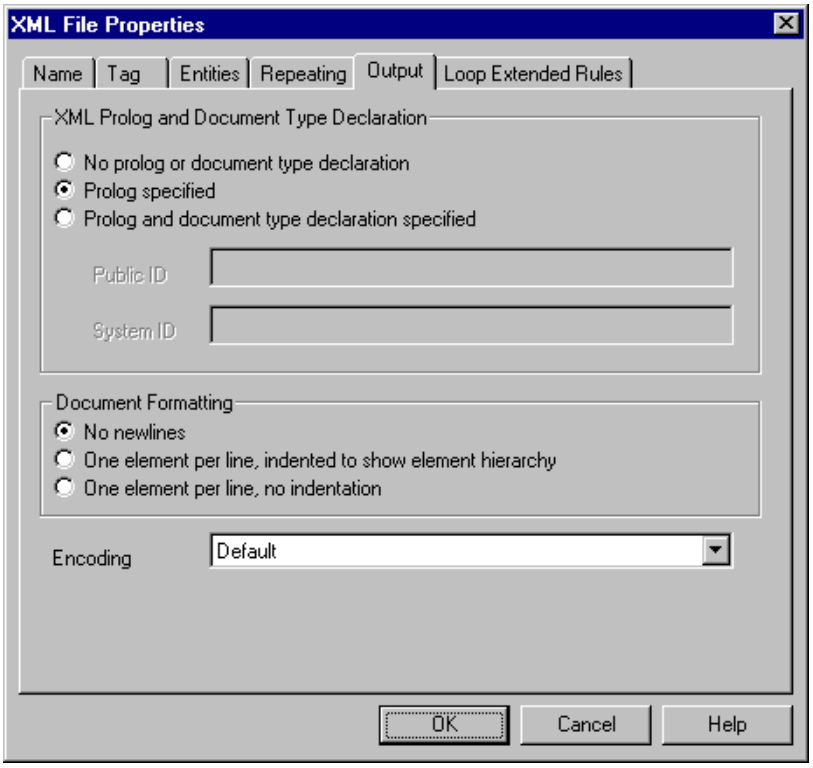
(Contd) Step	Action

10	In the extended rule box,
	◗ accept the default On Begin option,
	◗ type the readblock/writeblock extended rule, and
	◗ click **OK**.
	Note
	The readblock/writeblock extended rule must search for the start tag of the envelope trailer. Once it finds that tag, it must break out of the loop so the envelope trailers are removed from the data.
	Example
	`STRING [1024] buffer; //declares a string variable buffer`
	`buffer=""; //initialize the buffer variable to null`
	
	`While readblock(buffer) do`
	`  Begin`
	`    Seg_id=strstr(buffer,"<segment");`
	`      if Seg_id!=-1 then //found the envelope trailer`
	`                         //tag`
	`        break;  //break out of the loop and process map`
	`      else`
	`        writeblock(buffer); //didn't find envelope`
	`                            //trailer tag--write the`
	`                            //buffer contents to the`
	`                            //output file`
	`  End`
	System response
	The system saves your change and closes the XML Element Properties dialog box.
	Reference
	See Using Extended Rules in the *Application Integration User's Guide* for more information on extended rules.

Modify the OUTPUT side of each map

For each map you created above, you also need to modify the OUTPUT side. Use this procedure to modify each map.

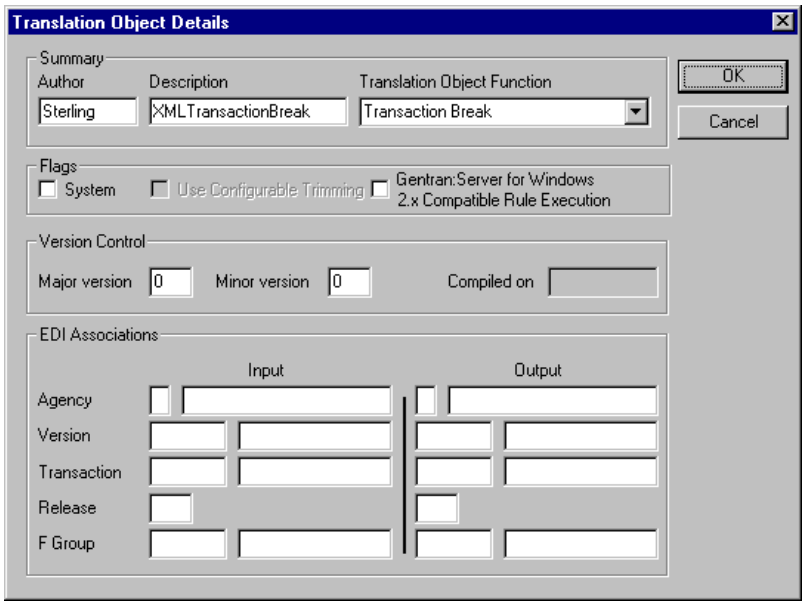
Step	Action
1	Right-click the XML File icon on the OUTPUT side of the map and select Properties from the shortcut menu. **System response** The system displays the XML File Properties dialog box (Name tab displayed by default). 
2	In the name box, type the name of the Root Element in your XML document.
3	Select the Tag tab. **System response** The system displays the tag options.
4	In the Tag box, type the tag of the Root Element in your XML document. (Continued on next page)

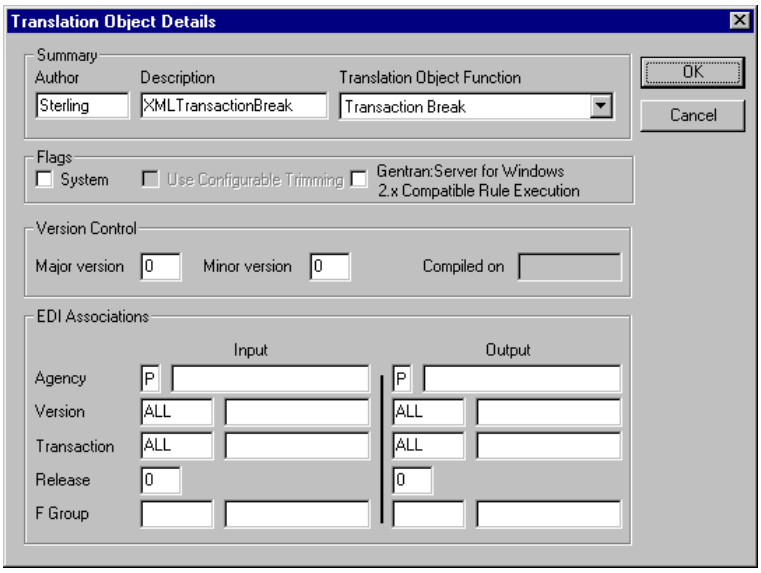
(Contd) Step	Action
5	Select the Output tab. **System response** The system displays the output options. 
6	◗ In the XML Prolog and Document Type Declaration section, select the No prolog option. ◗ In the Document Formatting section, select the No newlines option. ◗ Click **OK**. **System response** The system saves your changes and closes the XML File Properties dialog box.
7	Click **OK**. **System response** The system saves your changes and closes the XML File Properties dialog box. (Continued on next page)

(Contd) Step	Action
8	Right-click the XML File icon and select Create Sub from the shortcut menu. **Note** You need to create a temporary XML element and pcdata so the map will compile correctly. **Reference** See *Overview: Creating Map Objects* on page 2 - 15 for more information on the Create Sub and Insert functions.
9	From the shortcut menu, select Element. **System response** The system displays the XML Element Properties dialog box (Name tab).
10	In the name box, type TEMP.
11	Click **OK**. **System response** The system saves the temporary element.
12	Right-click the element you created in Step 8 and select Create Sub from the shortcut menu.
13	From the shortcut menu, select Pcdata. **System response** The system displays the Pcdata Properties dialog box (Validation tab).
14	Click **OK**. **System response** The system closes the Pcdata Properties dialog box.

Modify the map details for each map

For each map you created above, you also need to modify the Translation Object Details dialog box. Use this procedure to modify each map.

Step	Action
1	From the Edit menu, select Details. **System response** The system displays the Translation Object Details dialog box. 
2	In the EDI Associations section, type a letter that is not currently used to designate a standards agency (e.g., P) in the first Agency box for both Input and Output. **Note** The information you enter must match your import specifications.
3	In the EDI Associations section, type ALL in the first Version box for both Input and Output. (Continued on next page)

(Contd) Step	Action
4	In the EDI Associations section, type ALL in the first Transaction box for both Input and Output. **System response** This diagram illustrates how the EDI Associations section of the Transaction Object Detail dialog box should look. 
5	Click **OK**. **System response** The system saves your changes and closes the Translation Object Details dialog box.

Other Tasks

Introduction

Overview

After creating your XML build and break maps you need to perform the following tasks.

Task	Description
1	Compile the completed and saved XML build and break map. **Reference** See How to Compile a Map in the *Application Integration User's Guide* for information on compiling a translation object.
2	Register the build and break translation objects with the Gentran:Server system. **Reference** See How to Register a New Translation Object in the *Gentran:Server User's Guide* for information on registering a translation object.
3	Create or modify the appropriate partner relationship in Gentran:Server. **Reference** See *XML Partner Relationships* on page A - 52 for information on XML-specific partner relationships.

XML Partner Relationships

Overview

There are several tasks that you need to perform on XML-specific partner relationships. The tasks that are necessary depend on whether the partner relationship is inbound or outbound.

Reference
See Using Partners in the *Gentran:Server User's Guide* for more information on creating a partner relationship.

Inbound relationship

These are the tasks you need to perform to complete an inbound XML partner relationships.

Task	Description
1	When you create the inbound relationship, select the agency you entered on the Translation Object Details dialog box. **Reference** See How to Create a New Inbound Relationship in the *Gentran:Server User's Guide* for information on creating an inbound partner relationship.
2	Modify the splitter configuration. **Reference** See *Modifying the Splitter Configuration* on page 2 - 6 for more information.

Outbound relationship

These are the tasks you need to perform to complete an outbound XML partner relationships.

Task	Description
1	When you create the outbound relationship, select the agency you entered on the Translation Object Details dialog box. **Reference** See How to Create a New Outbound Relationship in the *Gentran:Server User's Guide* for information on creating an outbound partner relationship. (Continued on next page)

Task	Description
2	Select the XML build translation object at each appropriate level (transaction, group, interchange). **References** ◗ See Outbound Translation Object Dialog Box in the *Gentran:Server User's Guide* for more information on selecting a transaction build translation object. ◗ See Outbound Group Entry Dialog Box in the *Gentran:Server User's Guide* for more information on selecting a group build translation object (if necessary). ◗ See Outbound Interchange Entry Dialog Box in the *Gentran:Server User's Guide* for more information on selecting an interchange build translation object.
3	Set up the generic envelope segment. **Reference** See the online Help for more information on Partner Editor envelope dialog boxes.

APPENDIX

B

XML Tutorial

Contents

Overview

Introduction

In this appendix

This appendix contains a tutorial for the Gentran:Server for Windows Application Integration subsystem, using XML. This tutorial includes an inbound mapping example (invoice) and an outbound mapping example (invoice).

The purpose of this tutorial is to present the general mapping process, using examples that teach you a logical approach and methods that should be utilized when you create your own maps.

Reference
To access reference information, refer to the *Application Integration User's Guide*. For additional information on the various tasks described, refer to the Gentran:Server for Windows *User's Guide*.

Tutorial assumptions and scenario

The following are the assumptions for this tutorial and a description of the scenario we are utilizing.

- Your company is MWT Manufacturing Co., which manufactures a wide variety of supplies for pet stores. MWT Manufacturing Co. sells and buys pet supplies directly to and from large pet supply retail chains.
- Pet Zone, a large pet supply retailer, is the trading partner with whom your company is exchanging invoices (i.e., your company both buys and sells goods with Pet Zone).

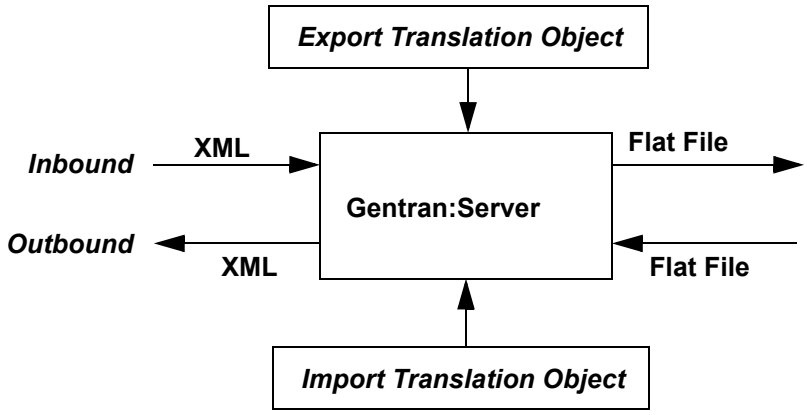
You need to create two maps—one that enables your company to translate the invoices that you receive from Pet Zone and one that allows you to translate your application file to the invoices that you send to Pet Zone.

For inbound processing, once the invoices from Pet Zone are translated into your application file format, they can be processed through your Order Processing System, and your company can ship the goods to Pet Zone. You have an existing application file layout from your order entry department that defines the information your system needs to process the order. And, from discussions with the Pet Zone, you know the data content of the orders that your partner is sending you.

For outbound processing, after you ship an order to Pet Zone you need to translate your application file to generate the corresponding invoice to send to Pet Zone. You have an existing application file layout from your accounts payable department, that defines the information your system generates for the invoice. And from discussions with the Pet Zone, you know the requirements for the invoices that you are sending them.

Diagram: Inbound and outbound translation process

This diagram illustrates the inbound and outbound translation process.



Import map

When you create an import map, you must define to Gentran:Server for Windows your flat file format and the XML format in which your partners expect to receive the documents.

Export map

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

Mapping analysis

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

This table describes the steps used in mapping analysis. These steps apply to translating both inbound and outbound data.

Step	Action
1	Analyze your file format.
2	Analyze your partner's file format.
3	Correlate your file and your partner's file formats.
4	Begin creating the map.

For this tutorial we have done the analysis for you.

Map Building Process

Overview This table provides an overview of the map-building process.

Stage	Description
1	**Prepare and Analyze** Obtain a layout of your flat file and determine how it corresponds with the XML. 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:Server for Windows Application Integration subsystem, you should establish the default date format, display, and confirmation options that the system uses. **Reference** See the *Application Integration User's Guide* for more information on setting these defaults.
3	**Register the Tutorial XML Build and Break Translation Objects** We provide you with XML build and break maps and translation objects to use in this tutorial in the GENSRVNT\Tutorial\XML folder. To use these translation objects you must register them with Gentran:Server. **Reference** See How to Register a New Translation Object in the *User's Guid*e for more information on registering translation objects.
4	**Set Up a Splitter Configuration Entry for XML** We provide you with tutorial XML break maps and translation objects in the GENSRVNT\Tutorial\XML folder. To use these translation objects you must select the build and break translation objects for the splitter entry you configured for XML. Also, you must change the Start tag for your XML splitter entry to PETTEST_INVOICE. **Reference** See *Modifying the Splitter Configuration* on page 2 - 6 for more information on setting up an XML Splitter entry.
5	**Create, Save, and Name a New Map** See *How to Create the Inbound Tutorial Map* on page B - 9 for more information on creating, saving, and naming a map. (Continued on next page)

(Contd) Stage	Description
6	**Define your Flat File Format** If you are creating an import or export map, you must define your flat file to the Application Integration subsystem. In Gentran:Server for Windows terminology, your flat 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). **Reference** See *How to Create the Inbound Tutorial Map* on page B - 9 for more information about defining your application file format by importing a DTD.
7	**Map the Appropriate Data for Each Application Field** To reconcile your flat file format with XML, you must identify each flat file component with its corresponding component in the XML file and select a method for mapping it. To map information to a field, you use linking, standard rules, extended rules, or a combination of all three. **Reference** See *How to Correlate your Flat File and the XML Invoice* on page B - 13 and *How to Perform Additional Mapping Requirements* on page B - 16 for more information about mapping your data.
8	**Compile the Translation Object** See *How to Compile a Map* on page B - 52 for more information on compiling the translation object and translation object naming conventions.
9	**Register the Translation Object with Gentran:Server** See How to Register a New Translation Object in the *Gentran:Server* User's Guide for more information on registering translation objects. (Continued on next page)

(Contd) Stage	Description
10	**Create or Import the Appropriate Trading Relationship** Establish the appropriate trading relationship in Gentran:Server for Windows for your trading partners. The trading relationship that you will use in this tutorial (PETZONE5) is loaded into the GENSRVNT\Tutorial\XML folder. You need to import the partner profile (PETZONE5.PAR) into Gentran:Server for Windows, before you begin this tutorial. **Reference** See the *Gentran:Server for Windows User's Guide* for more information on creating and importing 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. **Reference** See *Testing the Translation Objects* on page B - 56 for more information on test the translation objects.

Using the Tutorial

Tutorial building blocks

For these tutorials, we have provided building blocks (a DTD and a DDF) to enable you to build the tutorial translation objects more quickly. The following table describes how to locate those building blocks:

Type	File Name	Default Path Location
DTD	PET_XML.DTD	C:\GENSRVNT\TUTORIAL\XML\ PET_XML.DTD
DDF	PET_INV.DDF	C:\GENSRVNT\TUTORIAL\XML\ PET_INV.DDF

Tutorial test data

For these tutorials, we have provided test data for the translation objects you create. The following table describes how to locate that test data:

Translation Object Type	Test Data Name	Default Path Location
Inbound (export)	Inbound_XML_Invoice.INT	C:\GENSRVNT\TUTORIAL\XML\ Inbound_XML_Invoice.INT
Outbound (import)	Outbound_XML_Invoice.TXT	C:\GENSRVNT\TUTORIAL\XML\ Outbound_XML_Invoice.TXT

Header translation object

For these tutorials, we have provided you with the system import translation object you will use in the outbound process. The following table describes how to locate that translation object:

Translation Object Type	Translation Object Name	Default Path Location
Outbound (system import	HEADER.TPL	C:\GENSRVNT\TUTORIAL\HEADER.TPL

Inbound XML Invoice to Flat File

Overview

Introduction

For inbound processing, you receive invoices (in XML format) from Pet Zone, and they are then translated into your application file format so they can be processed and your company can ship the goods to Pet Zone. You have an existing application file layout from your order entry department that defines the information your system needs to process the order. And, from discussions with the Pet Zone, you know the data content of the orders that your partner is sending you.

How to Create the Inbound Tutorial Map

Introduction

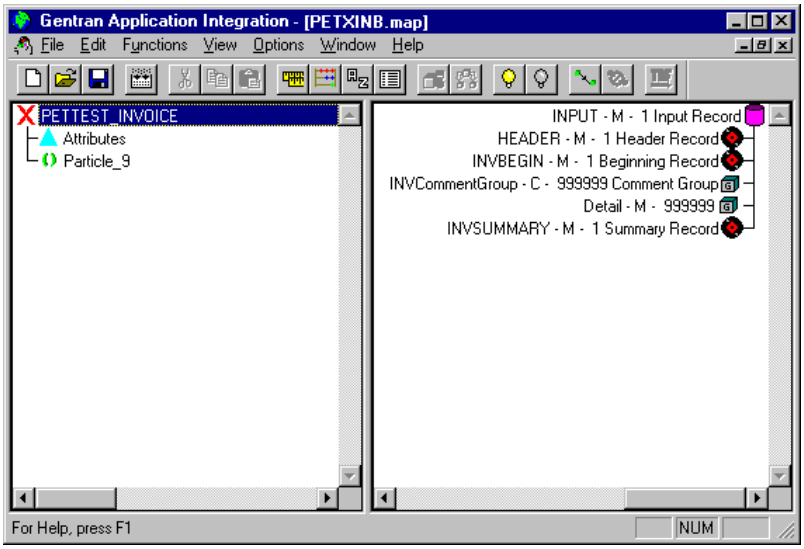
The New Map Wizard enables you to quickly and easily create a map. As part of the map creation process, the New XML Wizard enables you to create your format from a selected predefined document source type (i.e., a DTD).

Procedure

Use this procedure to create the inbound tutorial map.

Step	Action
1	From the Application Integration File menu, select **New**. **System response** The system displays the New Map Wizard.
2	◗ *What kind of map are you creating?* Select **Export**. ◗ *What is the name of the map?* Type **Pettest Inbound XML Invoice**, which is the unique name of the map. ◗ *What is your name?* Type your name if it differs from the user name prompted by the system. ◗ Click **Next**. **System response** The system displays the New Map Wizard - Input Format dialog box. **Note** You need to complete the format of the Input side of the map. This is the format of the data that is translated by the Gentran:Server system.
3	For the input side of the map you want to create a new data format using a syntax that you define so select **XML** (Extensible Markup Language), click **Customize** and continue with the next step. **System response** The system displays the New XML Wizard dialog box. (Continued on next page)

(Contd) Step	Action
4	Follow these steps for the appropriate dialog box and then continue with the next step. a. Select **DTD** as the document source type and click **Next**. b. Browse to the \\GENSRVNT\Tutorial\XML directory (click the **Browse (...)** button), select **PET_XML.DTD**, click **Open**, and then click **Next**. c. Select **PETTEST_INVOICE** as the doctype. **Note** The DTD does not explicitly define the root element, so you can choose from all the elements defined in the DTD. By default, the wizard selects the first element encountered in the DTD. d. Set the maximum length of data elements to **256** and click **Next**. **Note** You can specify the maximum length of data elements because this is not defined in the DTD. e. Click **Finish**. **Note** If the system needed to make changes to the DTD to make it compliant with Gentran:Server, the system informs you of the changes. Click **OK**. f. Click **Next**. **System response** The system displays the New Map Wizard - Output Format dialog box.
7	For the output side of the map you want to load the data format from a saved definition, so complete the following steps and then continue with the next step. a. Select the **Load the data format from a saved definition** option. b. Browse to the \\GENSRVNT\Tutorial\XML directory and select the **PET_INV.DDF** definition file. c. Click **Next**. **System response** The system displays the New Map Wizard dialog box. (Continued on next page)

(Contd) Step	Action
8	Click **Finish** to create the new map from the information you selected (this may take a few seconds). **System response** The system displays the new map in the Application Integration Window. 
9	From the File menu, select **Save**. **System response** The system displays the Save As dialog box.
10	In the File name box accept the name **Pettest Inbound XML Invoice.MAP**. **System response** The system adds the .MAP extension and saves the map.
11	From the Edit menu, select **Details**. **System response** The system displays the Transaction Object Details dialog box.
12	In the EDI Associations section (Input side), complete the following: ◗ In the first Agency box, type **P**. ◗ In the Version box, type **1.0**. ◗ In the Transaction box, type **INV**. ◗ In the Release box, type **0** (zero). ◗ Leave the F Group box blank. ◗ Continue with the next step. (Continued on next page)

(Contd) Step	Action
13	Do you want to change the map version? ◗ If *yes*, type the appropriate version numbers in the Major and Minor boxes and continue with the next step. ◗ If *no*, continue with the next step.
14	Click **OK**. **System response** The system saves your changes and exits the Translation Object Details dialog box.

Next step

After you finish creating the new map you can begin correlating the Input and Output sides.

Reference

Continue with *How to Correlate your Flat File and the XML Invoice* on page B - 13.

How to Correlate your Flat File and the XML Invoice

Overview

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

In this tutorial, we have done the correlation for you. Most fields in the two files can be correlated by linking (simple mapping), which enables you to map a map component from the input side of the map to a map component on the output side of the map. The link between two map components is visually represented with a line connecting them.

Linking procedure

Use this procedure to link components on the Input side of the map to the Output side the purchase order date.

Step	Action
1	If the Linking function is not currently active, click the Link button on the Main Toolbar to turn on the Linking function.
2	Click the INPUT map component. When you click the map component, the cursor changes to a “Link Arrow”: 
3	Click the OUTPUT map component to which you want to link the INPUT one.
4	A line is displayed, visually illustrating the link between the INPUT and OUTPUT map components.

HEADER map components to be linked

This table correlates all the HEADER record Output map components with their corresponding Input attribute. Link the map components in this table by using the Application Integration Link function.

Link this INPUT attribute...	To this HEADER record OUTPUT field...
partner	PARTNER_ID
transaction	TRANSACTION_SET
version	VERSION

INVBEGIN map components to be linked

This table correlates all the INVBEGIN record Output fields with their corresponding Input pcdata. Link the map components in this table by using the Application Integration Link function.

Link this INPUT pcdata...	To this INVBEGIN record OUTPUT field...
HEADER\InvoiceDate	INVDATE
HEADER\InvoiceNumber	INVNUMBEG
HEADER\PODate	PODATE
HEADER\PONumber	PONUM
HEADER\BillOfLading	BOLNUM
HEADER\ShipDate	SHIPDATE
HEADER\VendorAddress\Name	VENDNAME
HEADER\VendorAddress\Street	VENDADD1
HEADER\VendorAddress\City	VENDCITY
HEADER\VendorAddress\State	VENDSTATE
HEADER\VendorAddress\ZipCode	VENDPOSTALCODE
HEADER\ShipTo\Name_2	SHIPTONAME
HEADER\ShipTo\Street_2	SHIPTOADD1
HEADER\ShipTo\City_2	SHIPTOCITY
HEADER\ShipTo\State_2	SHIPTOSTATE
HEADER\ShipTo\ZipCode_2	SHIPTOPOSTALCODE
HEADER\RemitTo\Name_3	REMITTONAME
HEADER\RemitTo\Street_3	REMITTOADD1
HEADER\RemitTo\City_3	REMITTOCITY
HEADER\RemitTo\State_3	REMITTOSTATE
HEADER\RemitTo\ZipCode_3	REMITTOPOSTALCODE
HEADER\TermsOfSale\DiscountPercent	TERMS_WHOLE
HEADER\TermsOfSale\DiscountDaysDue	TERMSDISDUE
HEADER\TermsOfSale\TermsNetDays	TERMSNETDUE

INVCOMMENT2 map components to be linked

This table correlates all the INVCOMMENT2 record Output fields with their corresponding Input Pcdata. Link the map components in this table by using the Application Integration Link function.

Link this INPUT pcdata...	To this INVCOMMENT2 record OUTPUT field...
HEADER\Comments	MESSAGETEXT
LINEITEM\AssignedIdentification	ITEMNUM
LINEITEM\QuantityShipped	QTYSHIP
LINEITEM\QuantityOrdered	QTYORD
LINEITEM\UOM	UOM
LINELITEM\UnitPrice	UNITPRICE
LINEITEM\ProductCode	CUSTPROCODE
LINEITEM\UPCCode	UPCCODE
LINEITEM\ItemDescription	ITEMDESC
LINEITEM\TermsTypeCode	TERMSTYPE
LINEITEM\TermsDiscountPercent	TERMSDISP

INVSUMMARY map components to be linked

This table correlates all the INVSUMMARY record Output fields with their corresponding Input pcdata. Link the map components in this table by using the Application Integration Link function.

Link this INPUT pcdata...	To this INVSUMMARY record OUTPUT field...
HEADER\InvoiceNumber	INVNUMSUM
SUMMARY\TotalInvoiceAmount	TOTALAMOUNT
SUMMARY\NumberOfLineItems	TRANSTOTAL

Next step

Continue with *How to Perform Additional Mapping Requirements* on page B - 16.

How to Perform Additional Mapping Requirements

Introduction

For this tutorial you need to use more advanced mapping techniques to map some of the Input map components to the Output side of the map. The mapping techniques that you will use are as follows:

- **Standard rules**, which give you access to mapping operation functions that are more complex than simple linking, but less involved than extended rules.
- **Extended rules**, which enable you to use a Gentran:Server for Windows proprietary programming language to perform virtually any mapping operation you require.

Reference
See the *Application Integration User's Guide* for more information on using standard and extended rules.

Additional mapping requirements for the INPUT side

You need to change the maximum number of times that two INPUT elements (Comments and LineItem) can repeat because this enables you to link those INPUT map components with OUTPUT ones that do not have the same maximum usage specified.

Use this procedure to specify the maximum usage for the Comments and LineItem elements.

Step	Action
1	On the INPUT side of the inbound map, right-click the Comments element (located within the Header element) and select **Properties** from the shortcut menu. **System response** The system displays the XML Element Properties dialog box.
2	Select the **Repeating** tab.
3	Select the **Can repeat, with a maximum usage** option.
4	In the Maximum Usage box, type **999999**.
5	Click **OK** to change the maximum usage of the Comments element.
6	On the INPUT side of the inbound map, right-click the LineItem element and select **Properties** from the shortcut menu. **System response** The system displays the XML Element Properties dialog box.
7	Select the **Repeating** tab.
8	Select the **Can repeat, with a maximum usage** option.

(Continued on next page)

(Contd) Step	Action
9	In the Maximum Usage box, type **999999**.
10	Click **OK** to change the maximum usage of the LineItem element.

Additional mapping requirements for OUTPUT fields

This table lists the fields on the OUTPUT side of the map on which you will perform additional mapping requirements via standard and extended rules.

Record	Field	Mapping to be performed
HEADER	STANDARD	Extended rule to map "P" into the field.
HEADER	TEST_PROD	Extended rule to map "P" into the field.
INVBEGIN	BILLTOACCTBEG	Use Constant standard rule to map constant value "INV5008" to field and indicate field is qualified by INVNUMBEG.
INVBEGIN	DOCTYPEBEG	Use Constant standard rule to map constant value "Invoice" to field and indicate field is qualified by INVNUMBEG.
INVCOMMENT2	BILLTOACCTCM2	Use Constant standard rule to map constant value "INV5008" to field and indicate field is qualified by MESSAGETEXT.
INVCOMMENT2	DOCTYPECM2	Use Constant standard rule to map constant value "Invoice" to field and indicate field is qualified by MESSAGETEXT.
INVCOMMENT2	INVNUMCM2	Extended rule to map from InvoiceNumber.
INVDETAIL	BILLTOACCTDET	Use Constant standard rule to map constant value "INV5008" to field and indicate field is qualified by ITEMNUM.
INVDETAIL	DOCTYPEDET	Use Constant standard rule to map constant value "Invoice" to field and indicate field is qualified by ITEMNUM.
INVDETAIL	INVNUMDET	Extended rule to map from InvoiceNumber.
INVSUMMARY	BILLTOACCTSUM	Use Constant standard rule to map constant value "INV5008" to field and indicate field is qualified by TOTALAMOUNT.
INVSUMMARY	DOCTYPSUM	Use Constant standard rule to map constant value "Invoice" to field and indicate field is qualified by TOTALAMOUNT.

Constants

Since you are creating an export map to create inbound data that will Use Constants in the qualifier fields on the XML (output) side of the map.

For this tutorial, the following constants are already defined for you (via the DDF you used to create the map) so you can use them to create the required qualifier elements.

Note
See the *Application Integration User's Guide* if you want to know how to define or edit constant values.

Fields Using Constant	Constant ID	Constant Value	Fields Qualified
BILLTOACCTBEG BILLTOACCTCM2 BILLTOACCTDET BILLTOACCTSUM		INV5008	INVNUMBEG MESSAGETEXT ITEMNUM TOTALAMOUNT
DOCTYPEBEG DOCTYPECM2 DOCTYPEDET DOCTYPSUM		Invoice	INVNUMBEG MESSAGETEXT ITEMNUM TOTALAMOUNT

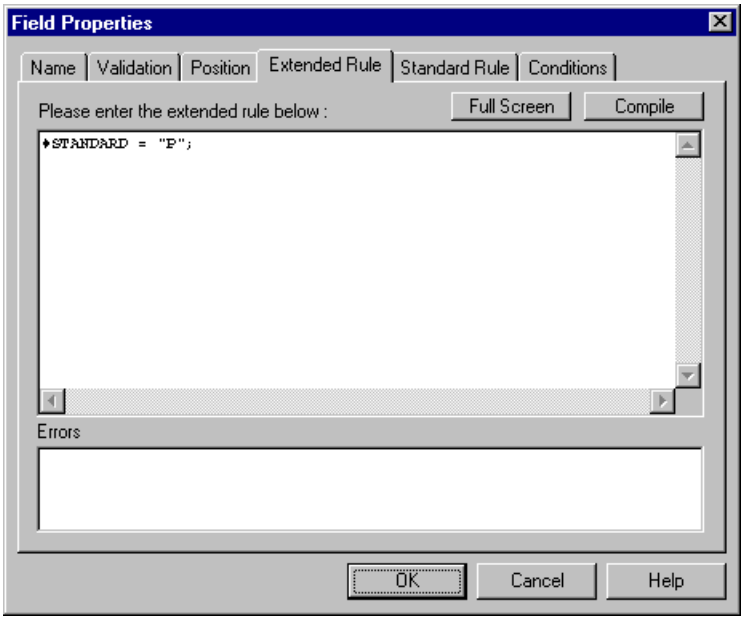
How to Map the HEADER\STANDARD Field

Introduction

You use an extended rule to map the value "P" into the STANDARD field in the HEADER record.

Procedure

Use this procedure to create the extended rule.

Step	Action
1	Double-click the STANDARD field in the Header record element to access the Field Properties dialog box.
2	Select the **Extended Rule** tab.
3	In the Rule list, type the following: `#STANDARD = "P";` This is an example of how the Field Properties dialog box looks. 
4	Click **Compile** to validate the syntax of the extended rule. Every rule in the map is compiled when you compile the translation object, after you complete the map. However, the system allows you to compile each rule individually, so that you can verify the accuracy of the rule after you create it.
5	Click **OK** to add the extended rule to the STANDARD field.

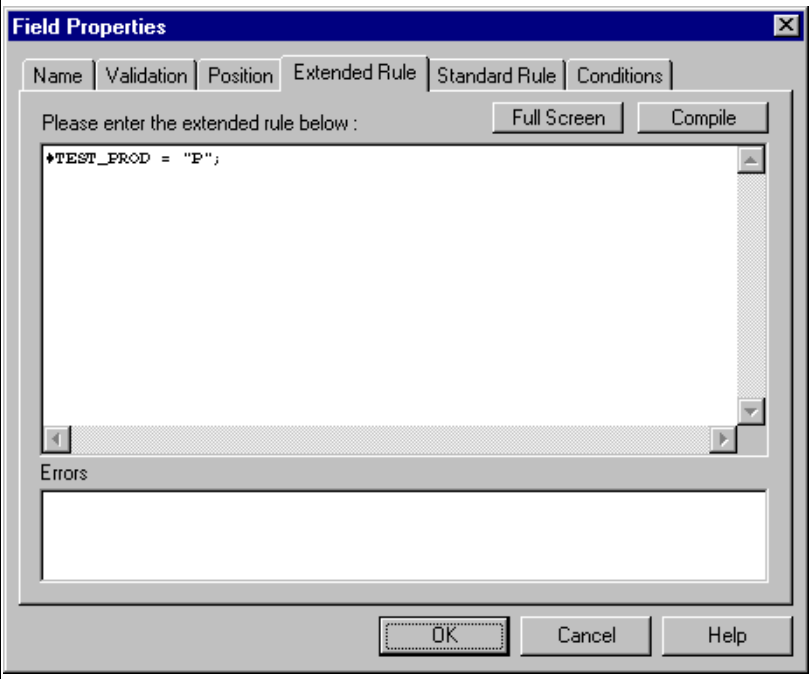
How to Map the HEADER\TEST_PROD Field

Introduction

You use an extended rule to map the value "P" into the TEST_PROD field in the HEADER record.

Procedure

Use this procedure to create the extended rule.

Step	Action
1	Double-click the TEST_PROD field in the Header record element to access the Field Properties dialog box.
2	Select the **Extended Rule** tab.
3	In the Rule list, type the following: `#TEST_PROD = "P";` This is an example of how the Field Properties dialog box looks. 
4	Click **Compile** to validate the syntax of the extended rule. Every rule in the map is compiled when you compile the translation object, after you complete the map. However, the system allows you to compile each rule individually, so that you can verify the accuracy of the rule after you create it.
5	Click **OK** to add the extended rule to the TEST_PROD field.

How to Map the INVBEGIN\BILLTOACCTBEG Field

Introduction

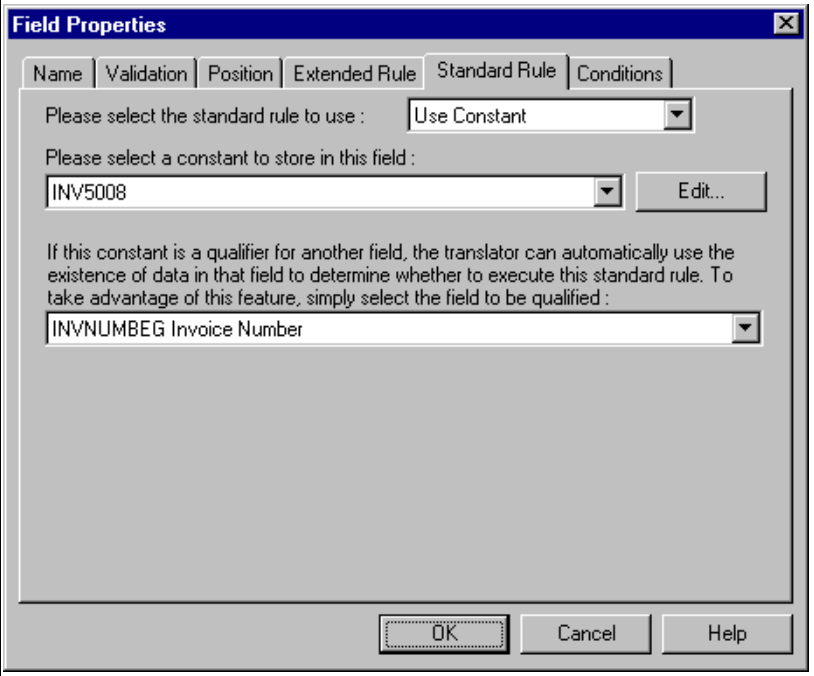
You need to set the Bill To Code for the BILLTOACCTBEG field in the INVBEGIN record so the system is able to distinguish the account that should be billed for this invoice.

You already defined all the constants that you are using in this map, so now you can map the appropriate constant to the BILLTOACCTBEG field using a Use Constant standard rule.

Procedure

Use this procedure to set the Bill To Code in the INVBEGIN record.

Step	Action
1	Double-click the BILLTOACCTBEG field in the INVBEGIN record to access the Field Properties dialog box.
2	Select the **Standard Rule** tab.
3	From the standard rule list, select **Use Constant**.
4	From the Constants list, select the “INV5008” constant, to identify this field as containing Bill To Account information. (Continued on next page)

(Contd) Step	Action
5	From the Qualifies list, select the "INVNUMBEG," to set up a qualifying relationship with the Invoice Number field. **Note** The Qualifies list contains only the other *active* fields in the same record as the qualifying field. **System response** This is an example of how the Field Properties dialog box looks. 
6	Click **OK** and the "INV5008" code is loaded to the BILLTOACCTBEG field and the qualifying relationship between the BILLTOACCTBEG and INVNUMBEG fields is established.

How to Map the INVBEGIN\DOCTYPEBEG Field

Introduction

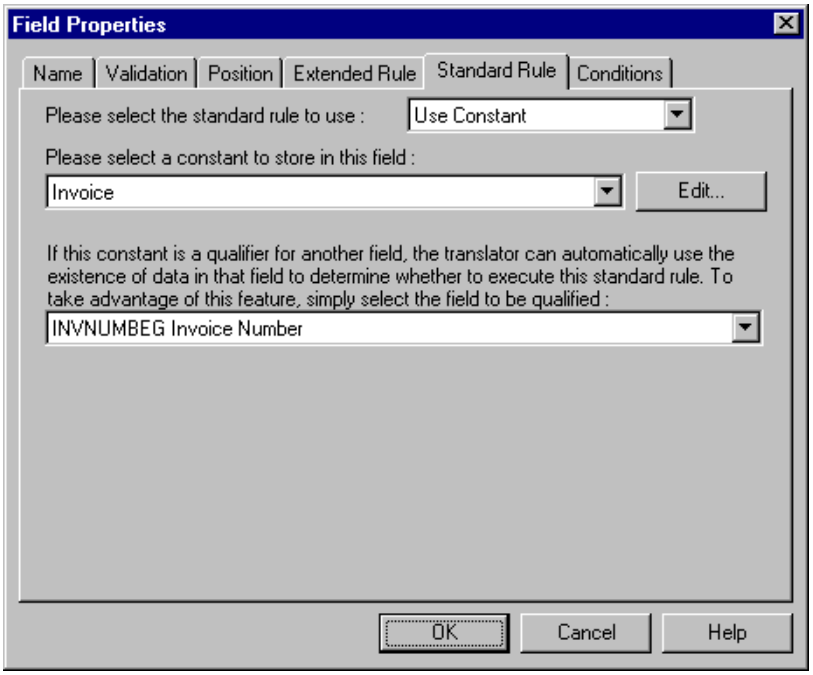
You need to set the Bill To Code for the DOCTYPEBEG field in the INVBEGIN record so the system is able to distinguish the account that should be billed for this invoice.

You already defined all the constants that you are using in this map, so now you can map the appropriate constant to the DOCTYPEBEG field using a Use Constant standard rule.

Procedure

Use this procedure to set the Bill To Code in the INVBEGIN record.

Step	Action
1	Double-click the DOCTYPEBEG field in the INVBEGIN record to access the Field Properties dialog box.
2	Select the **Standard Rule** tab.
3	From the standard rule list, select **Use Constant**.
4	From the Constants list, select the “Invoice” constant, to identify this field as containing Document Type information. (Continued on next page)

(Contd) Step	Action
5	From the Qualifies list, select the "INVNUMBEG," to set up a qualifying relationship with the Invoice Number field. **Note** The Qualifies list contains only the other *active* fields in the same record as the qualifying field. **System response** This is an example of how the Field Properties dialog box looks. 
6	Click **OK** and the "Invoice" code is loaded to the DOCTYPEBEG field and the qualifying relationship between the DOCTYPEBEG and INVNUMBEG fields is established.

How to Map the INVCOMMENT2\BILLTOACCTCM2 Field

Introduction

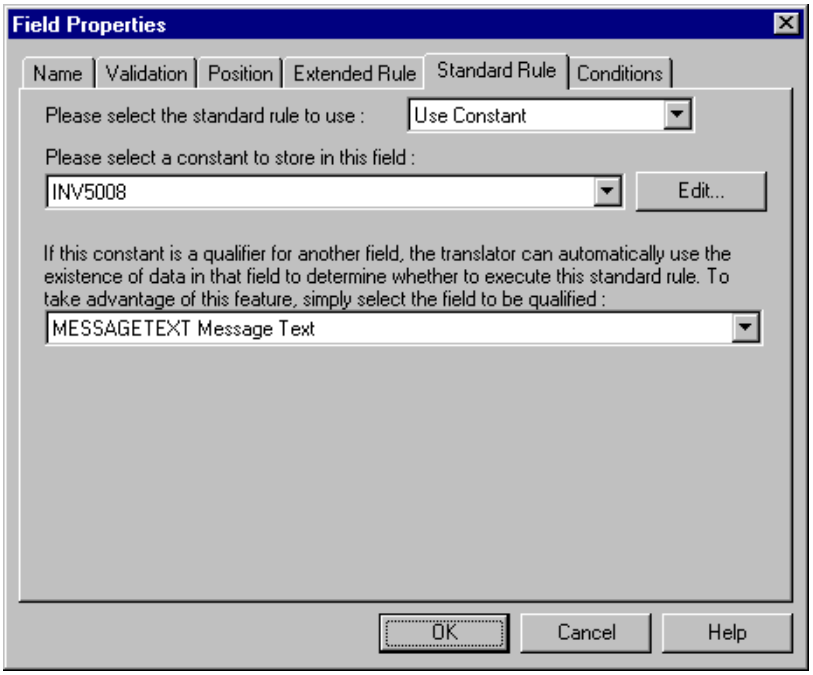
You need to set the Bill To Code for the BILLTOACCTCM2 field in the INVCOMMENT2 record so the system is able to distinguish the account that should be billed for this invoice.

You already defined all the constants that you are using in this map, so now you can map the appropriate constant to the BILLTOACCTCM2 field using a Use Constant standard rule.

Procedure

Use this procedure to set the Bill To Code in the INVCOMMENT2 record.

Step	Action
1	Double-click the BILLTOACCTCM2 field in the INVCOMMENT2 record to access the Field Properties dialog box.
2	Select the **Standard Rule** tab.
3	From the standard rule list, select **Use Constant**.
4	From the Constants list, select the “INV5008” constant, to identify this field as containing Bill To Account information. (Continued on next page)

(Contd) Step	Action
5	From the Qualifies list, select the "MESSAGETEXT," to set up a qualifying relationship with the Message Text field. **Note** The Qualifies list contains only the other *active* fields in the same record as the qualifying field. **System response** This is an example of how the Field Properties dialog box looks. 
6	Click **OK** and the "INV5008" code is loaded to the BILLTOACCTCM2 field and the qualifying relationship between the BILLTOACCTCM2 and MESSAGETEXT fields is established.

How to Map the INVCOMMENT2\DOCTYPECM2 Field

Introduction

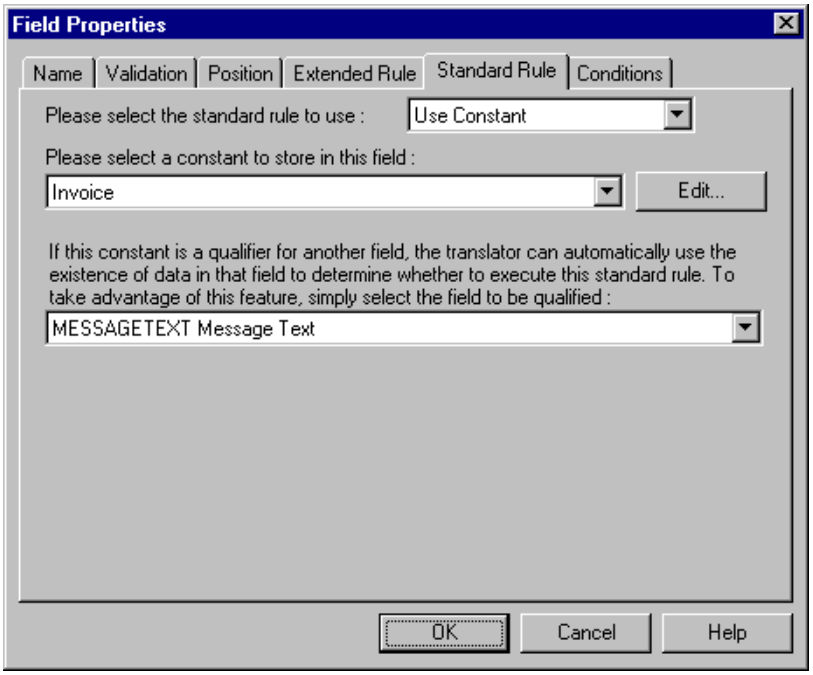
You need to set the Bill To Code for the DOCTYPECM2 field in the INVCOMMENT2 record so the system is able to distinguish the account that should be billed for this invoice.

You already defined all the constants that you are using in this map, so now you can map the appropriate constant to the DOCTYPECM2 field using a Use Constant standard rule.

Procedure

Use this procedure to set the Bill To Code in the INVCOMMENT2 record.

Step	Action
1	Double-click the DOCTYPECM2 field in the INVCOMMENT2 record to access the Field Properties dialog box.
2	Select the **Standard Rule** tab.
3	From the standard rule list, select **Use Constant**.
4	From the Constants list, select the “Invoice” constant, to identify this field as containing Document Type information. (Continued on next page)

(Contd) Step	Action
5	From the Qualifies list, select the "MESSAGETEXT," to set up a qualifying relationship with the Message Text field. **Note** The Qualifies list contains only the other *active* fields in the same record as the qualifying field. **System response** This is an example of how the Field Properties dialog box looks. 
6	Click **OK** and the "Invoice" code is loaded to the DOCTYPECM2 field and the qualifying relationship between the DOCTYPECM2 and MESSAGETEXT fields is established.

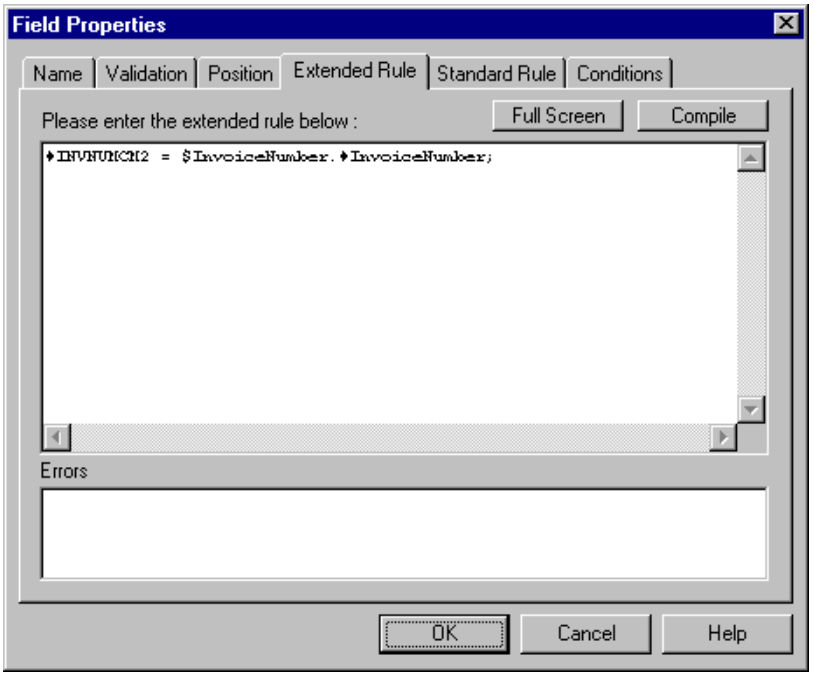
How to Map the INVCOMMENT2\INVNUMCM2 Field

Introduction

You use an extended rule to map the value in the InvoiceNumber field on the Input side of the map into the INVNUMCM2 field in the INVCOMMENT2 record.

Procedure

Use this procedure to create the extended rule.

Step	Action
1	Double-click the INVNUMCM2 field in the Header record element to access the Field Properties dialog box.
2	Select the **Extended Rule** tab.
3	In the Rule list, type the following: `#INVNUMCM2 = $InvoiceNumber.#InvoiceNumber;` This is an example of how the Field Properties dialog box looks.  (Continued on next page)

(Contd) Step	Action
4	Click **Compile** to validate the syntax of the extended rule. Every rule in the map is compiled when you compile the translation object, after you complete the map. However, the system allows you to compile each rule individually, so that you can verify the accuracy of the rule after you create it.
5	Click **OK** to add the extended rule to the INVNUMCM2 field.

How to Map the INVDETAIL\BILLTOACCTDET Field

Introduction

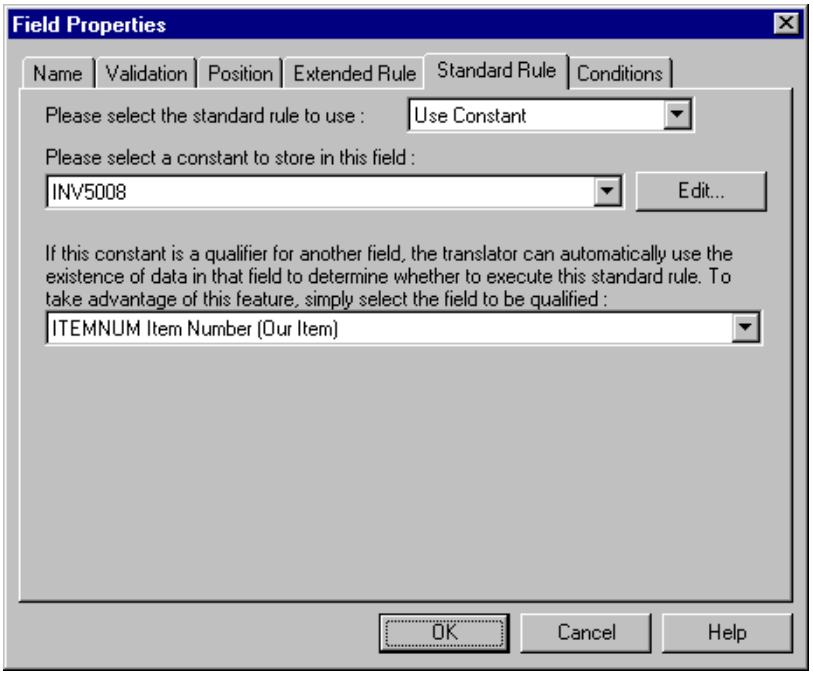
You need to set the Bill To Code for the BILLTOACCTDET field in the INVDETAIL record so the system is able to distinguish the account that should be billed for this invoice.

You already defined all the constants that you are using in this map, so now you can map the appropriate constant to the BILLTOACCTDET field using a Use Constant standard rule.

Procedure

Use this procedure to set the Bill To Code in the INVDETAIL record.

Step	Action
1	Double-click the BILLTOACCTDET field in the INVDETAIL record to access the Field Properties dialog box.
2	Select the **Standard Rule** tab.
3	From the standard rule list, select **Use Constant**.
4	From the Constants list, select the “INV5008” constant, to identify this field as containing Bill To Account information. (Continued on next page)

(Contd) Step	Action
5	From the Qualifies list, select the "ITEMNUM," to set up a qualifying relationship with the Item Number field. **Note** The Qualifies list contains only the other *active* fields in the same record as the qualifying field. **System response** This is an example of how the Field Properties dialog box looks. 
6	Click **OK** and the "INV5008" code is loaded to the BILLTOACCTDET field and the qualifying relationship between the BILLTOACCTDET and ITEMNUM fields is established.

How to Map the INVDETAIL\DOCTYPEDET Field

Introduction

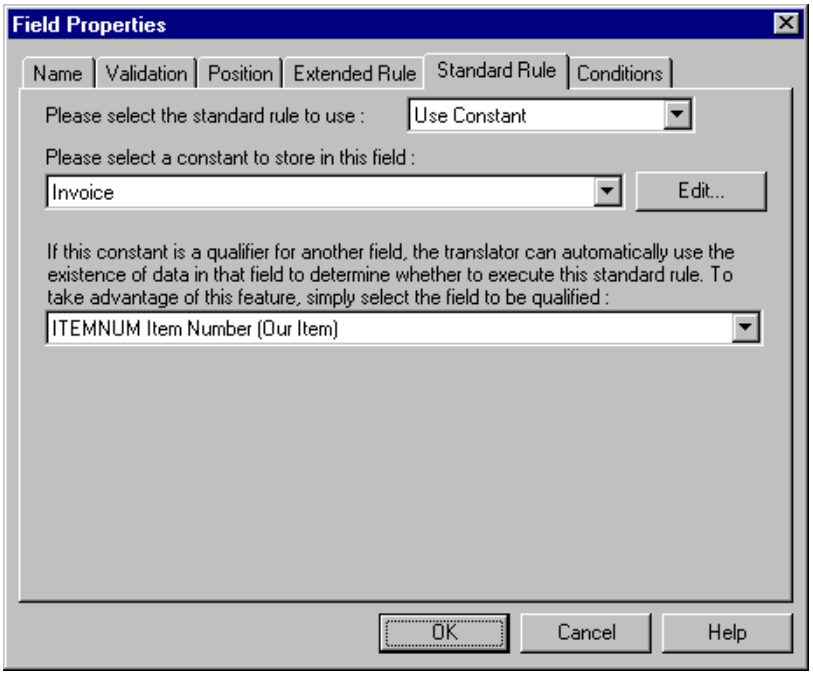
You need to set the Bill To Code for the DOCTYPEDET field in the INVDETAIL record so the system is able to distinguish the account that should be billed for this invoice.

You already defined all the constants that you are using in this map, so now you can map the appropriate constant to the DOCTYPEDET field using a Use Constant standard rule.

Procedure

Use this procedure to set the Bill To Code in the INVDETAIL record.

Step	Action
1	Double-click the DOCTYPEDET field in the INVDETAIL record to access the Field Properties dialog box.
2	Select the **Standard Rule** tab.
3	From the standard rule list, select **Use Constant**.
4	From the Constants list, select the “Invoice” constant, to identify this field as containing Document Type information. (Continued on next page)

(Contd) Step	Action
5	From the Qualifies list, select the "ITEMNUM," to set up a qualifying relationship with the Item Number field. **Note** The Qualifies list contains only the other *active* fields in the same record as the qualifying field. **System response** This is an example of how the Field Properties dialog box looks. 
6	Click **OK** and the "Invoice" code is loaded to the DOCTYPEDET field and the qualifying relationship between the DOCTYPEDET and ITEMNUM fields is established.

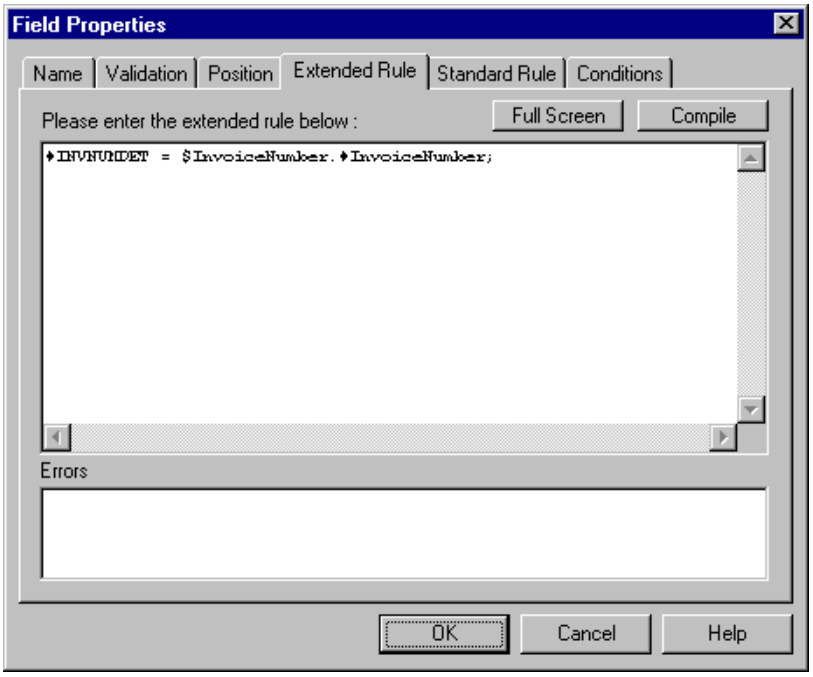
How to Map the INVDETAIL\INVNUMDET Field

Introduction

You use an extended rule to map the value in the InvoiceNumber field on the Input side of the map into the INVNUMDET field in the INVDETAIL record.

Procedure

Use this procedure to create the extended rule.

Step	Action
1	Double-click the INVNUMDET field in the Header record element to access the Field Properties dialog box.
2	Select the **Extended Rule** tab.
3	In the Rule list, type the following: `#INVNUMDET = $InvoiceNumber.#InvoiceNumber;` This is an example of how the Field Properties dialog box looks.  (Continued on next page)

(Contd) Step	Action
4	Click **Compile** to validate the syntax of the extended rule. Every rule in the map is compiled when you compile the translation object, after you complete the map. However, the system allows you to compile each rule individually, so that you can verify the accuracy of the rule after you create it.
5	Click **OK** to add the extended rule to the INVNUMDET field.

How to Map the INVSUMMARY\BILLTOACCTSUM Field

Introduction

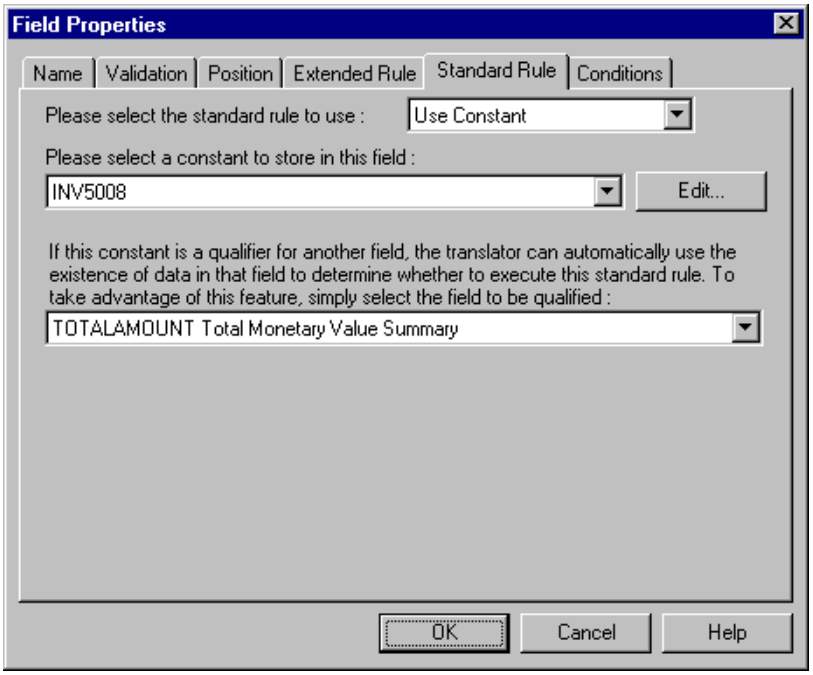
You need to set the Bill To Code for the BILLTOACCTSUM field in the INVSUMMARY record so the system is able to distinguish the account that should be billed for this invoice.

You already defined all the constants that you are using in this map, so now you can map the appropriate constant to the BILLTOACCTSUM field using a Use Constant standard rule.

Procedure

Use this procedure to set the Bill To Code in the INVSUMMARY record.

Step	Action
1	Double-click the BILLTOACCTSUM field in the INVSUMMARY record to access the Field Properties dialog box.
2	Select the **Standard Rule** tab.
3	From the standard rule list, select **Use Constant**.
4	From the Constants list, select the “INV5008” constant, to identify this field as containing Bill To Account information. (Continued on next page)

(Contd) Step	Action
5	From the Qualifies list, select the "TOTALAMOUNT," to set up a qualifying relationship with the Total Invoice Amount field. **Note** The Qualifies list contains only the other *active* fields in the same record as the qualifying field. **System response** This is an example of how the Field Properties dialog box looks. 
6	Click **OK** and the "INV5008" code is loaded to the BILLTOACCTSUM field and the qualifying relationship between the BILLTOACCTSUM and TOTALAMOUNT fields is established.

How to Map the INVSUMMARY\DOCTYPSUM Field

Introduction

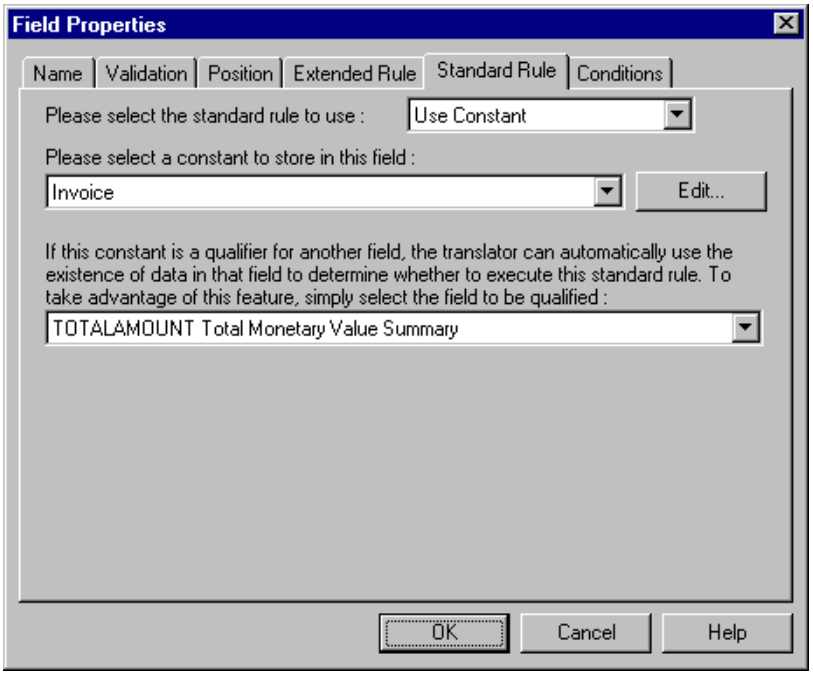
You need to set the Bill To Code for the DOCTYPSUM field in the INVSUMMARY record so the system is able to distinguish the account that should be billed for this invoice.

You already defined all the constants that you are using in this map, so now you can map the appropriate constant to the DOCTYPSUM field using a Use Constant standard rule.

Procedure

Use this procedure to set the Bill To Code in the INVSUMMARY record.

Step	Action
1	Double-click the DOCTYPSUM field in the INVSUMMARY record to access the Field Properties dialog box.
2	Select the **Standard Rule** tab.
3	From the standard rule list, select **Use Constant**.
4	From the Constants list, select the “Invoice” constant, to identify this field as containing Document Type information. (Continued on next page)

(Contd) Step	Action
5	From the Qualifies list, select the "TOTALAMOUNT," to set up a qualifying relationship with the Total Invoice Amount field. **Note** The Qualifies list contains only the other *active* fields in the same record as the qualifying field. **System response** This is an example of how the Field Properties dialog box looks. 
6	Click **OK** and the "Invoice" code is loaded to the DOCTYPSUM field and the qualifying relationship between the DOCTYPSUM and TOTALAMOUNT fields is established.

Next step

After you create the export map and perform the appropriate mapping operations, you need to finalize the map. To complete the mapping process, you need to save the map, compile the translation object, print and review the mapping report, and test the map.

Reference

Continue with *Finalizing the Maps* on page B - 51.

Outbound Flat File to XML Invoice

Overview

Introduction

For outbound processing, you need to send invoices (currently in flat file format) to Pet Zone. Therefore, you need to translate the invoices into XML format so Pet Zone will be able to process them and pay your company for the goods you shipped. You have an existing application file layout from your accounts payable department that defines the invoice information your system needs to send to Pet Zone. And, from discussions with the Pet Zone, you know the content of the invoice data that they expect to receive from you.

System import map

When you are translating data outbound, you need to build an import map and a system import map. The system import map is used by Gentran:Server to find the partner relationship for a document, 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.

In this tutorial, we provide you with the system import map and compiled translation object you need to use.

Reference
The default full path for the header translation object is:

```
C:\GENSRVNT\TUTORIAL\HEADER.TPL
```

There are two ways to build the key in a system import map:

- the five-field key
- the three-field key.

The method that we recommend, and provide you with in this tutorial, requires five specific fields in the header record:

- partner key
- standard
- version
- transaction set
- test/production status

(Continued on next page)

System import map (contd.)

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. Typically, you use this method when you are defining your application from scratch and can easily add the specific fields you need that are not already present in the header record.

Reference

See Setting Up the Import Process in the *Application Integration User's Guide* for more information on how to create a system import map.

How to Create the Outbound Tutorial Map

Introduction

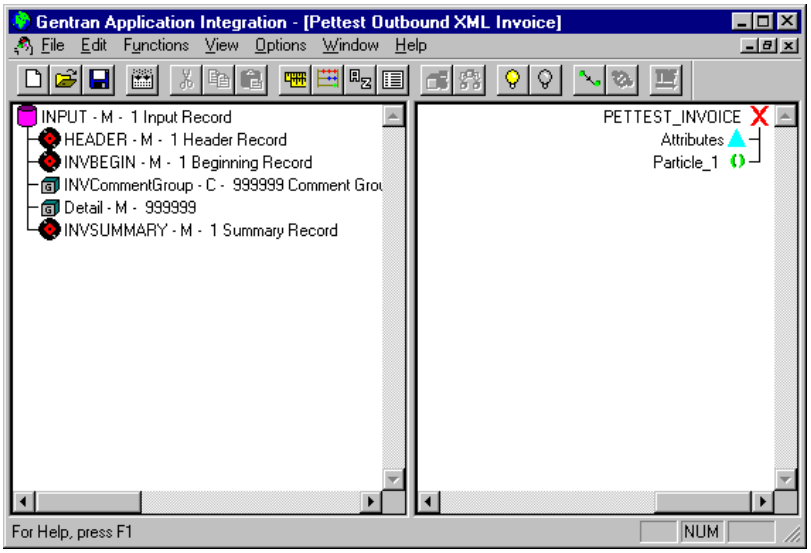
The New Map Wizard enables you to quickly and easily create a map. As part of the map creation process, the New XML Wizard enables you to create your format from a selected predefined document source type (i.e., a DTD).

Procedure

Use this procedure to create the outbound tutorial map.

Step	Action
1	From the Application Integration File menu, select **New**. **System response** The system displays the New Map Wizard.
2	◗ *What kind of map are you creating?* Select **Import**. ◗ *What is the name of the map?* Type **Pettest Outbound XML Invoice**, which is the unique name of the map. ◗ *What is your name?* Type your name if it differs from the user name prompted by the system. ◗ Click **Next**. **System response** The system displays the New Map Wizard - Input Format dialog box. **Note** You need to complete the format of the Input side of the map. This is the format of the data that is translated by the Gentran:Server system.
3	For the input side of the map you want to load your application data format from a saved definition, so complete the following steps and then continue with the next step. a. Select the **Load the data format from a saved definition** option. b. Browse to the \\GENSRVNT\Tutorial\XML directory and select the **PET_INV.DDF** definition file. c. Click **Next**. **System response** The system displays the New Map Wizard - Output Format dialog box. (Continued on next page)

(Contd) Step	Action
4	For the output side of the map you want to create a new data format using a syntax that you define so select **XML** (Extensible Markup Language), click **Customize** and continue with the next step. **System response** The system displays the New XML Wizard dialog box.
5	Follow these steps for the appropriate dialog box and then continue with the next step. a. Select **DTD** as the document source type and click **Next**. b. Browse to the \\GENSRVNT\Tutorial\XML directory (click the **Browse** (**...**) button), select **PET_XML.DTD**, click **Open**, and then click **Next**. c. Select **PETTEST_INVOICE** as the doctype. **Note** The DTD does not explicitly define the root element, so you can choose from all the elements defined in the DTD. By default, the wizard selects the first element encountered in the DTD. d. Set the maximum length of data elements to **256** and click **Next**. **Note** You can specify the maximum length of data elements because this is not defined in the DTD. e. Click **Finish**. **Note** If the system needed to make changes to the DTD to make it compliant with Gentran:Server, the system informs you of the changes. Click **OK**. f. Click **Next**. **System response** The system displays the New Map Wizard dialog box. (Continued on next page)

(Contd) Step	Action
6	Click **Finish** to create the new map from the information you selected (this may take a few seconds). **System response** The system displays the new map in the Application Integration Window. 
7	From the File menu, select **Save**. **System response** The system displays the Save As dialog box.
8	In the File name box accept the name **Pettest Outbound XML Invoice**. **System response** The system adds the .MAP extension and saves the map.
9	From the Edit menu, select **Details**. **System response** The system displays the Transaction Object Details dialog box.
10	In the EDI Associations section (Output side), complete the following: ◗ In the first Agency box, type **P**. ◗ In the Version box, type **1.0**. ◗ In the Transaction box, type **INV**. ◗ In the Release box, type **0** (zero). ◗ Leave the F Group box blank. ◗ Continue with the next step. (Continued on next page)

(Contd) Step	Action
11	Do you want to change the map version? ◗ If *yes*, type the appropriate version numbers in the Major and Minor boxes and continue with the next step. ◗ If *no*, continue with the next step.
12	Click **OK**. **System response** The system saves your changes and exits the Translation Object Details dialog box.

Next step

After you finish creating the new map you can begin correlating the Input and Output sides.

Reference

Continue with *How to Correlate your Flat File and the XML Invoice* on page B - 47.

How to Correlate your Flat File and the XML Invoice

Overview

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

In this tutorial, we have done the correlation for you. Most fields in the two files can be correlated by linking (simple mapping), which enables you to map a map component from the input side of the map to a map component on the output side of the map. The link between two map components is visually represented with a line connecting them.

Linking procedure

Use this procedure to link components on the Input side of the map to the Output side the purchase order date.

Step	Action
1	If the Linking function is not currently active, click the Link button on the Main Toolbar to turn on the Linking function.
2	Click the INPUT map component. When you click the map component, the cursor changes to a “Link Arrow”: 
3	Click the OUTPUT map component to which you want to link the INPUT one.
4	A line is displayed, visually illustrating the link between the INPUT and OUTPUT map components.

Attributes to be linked

This table correlates all the Output attributes with their corresponding Input field. Link the fields in this table by using the Application Integration Link function.

Link this INPUT field...	To this OUTPUT attribute...
PARTNER_ID	partner
VERSION	version
TRANSACTION_SET	transaction

Header pcdatas to be linked

This table correlates all the Output Header pcdatas with their corresponding Input field. Link the fields in this table by using the Application Integration Link function.

Link this INPUT field...	To this Header OUTPUT pcdata...
INVDATE	HEADER\InvoiceDate
INVNUMBEG	HEADER\InvoiceNumber
PODATE	HEADER\PODate
PONUM	HEADER\PONumber
BOLNUM	HEADER\BillOfLading
SHIPDATE	HEADER\ShipDate
VENDNAME	HEADER\VendorAddress\Name
VENDADD1	HEADER\VendorAddress\Street
VENDCITY	HEADER\VendorAddress\City
VENDSTATE	HEADER\VendorAddress\State
VENDPOSTALCODE	HEADER\VendorAddress\ZipCode
SHIPTONAME	HEADER\ShipTo\Name_2
SHIPTOADD1	HEADER\ShipTo\Street_2
SHIPTOCITY	HEADER\ShipTo\City_2
SHIPTOSTATE	HEADER\ShipTo\State_2
SHIPTOPOSTALCODE	HEADER\ShipTo\ZipCode_2
REMITTONAME	HEADER\RemitTo\Name_3
REMITTOADD1	HEADER\RemitTo\Street_3
REMITTOCITY	HEADER\RemitTo\City_3
REMITTOSTATE	HEADER\RemitTo\State_3
REMITTOPOSTALCODE	HEADER\RemitTo\ZipCode_3
TERMS_WHOLE	HEADER\TermsOfSale\DiscountPercent
TERMSDISDUE	HEADER\TermsOfSale\DiscountDaysDue
TERMSNETDUE	HEADER\TermsOfSale\TermsNetDays

LineItem pcdatas to be linked

This table correlates all the LineItem Output pcdatas with their corresponding Input field. Link the fields in this table by using the Application Integration Link function.

Link this INPUT field...	To this LineItem OUTPUT pcdata...
MESSAGETEXT	HEADER\Comments
ITEMNUM	LINEITEM\AssignedIdentification
QTYSHIP	LINEITEM\QuantityShipped
QTYORD	LINEITEM\QuantityOrdered
UOM	LINEITEM\UOM
UNITPRICE	LINELITEM\UnitPrice
CUSTPROCODE	LINEITEM\ProductCode
UPCCODE	LINEITEM\UPCCode
ITEMDESC	LINEITEM\ItemDescription
TERMSTYPE	LINEITEM\TermsTypeCode
TERMSDISP	LINEITEM\TermsDiscountPercent

Summary pcdatas to be linked

This table correlates all the Output Summary pcdatas with their corresponding Input field. Link the fields in this table by using the Application Integration Link function.

Link this INPUT field...	To this Summary OUTPUT pcdata...
TOTALAMOUNT	SUMMARY\TotalInvoiceAmount
TRANSTOTAL	SUMMARY\NumberOfLineItems

Next step

After you create the import map and perform the appropriate mapping operations, you need to finalize the map. To complete the mapping process, you need to save the map, compile the translation object, print and review the mapping report, and test the map.

Reference

Continue with *Finalizing the Maps* on page B - 51.

How to Perform Additional Mapping Requirements

Introduction

For this tutorial you do not need to use standard or extended rules on the outbound map but you do need to perform the additional requirements listed in this section.

Additional mapping requirements for the OUTPUT side

You need to change the maximum number of times that two OUTPUT elements (Comments and LineItem) can repeat because this enables you to link those OUTPUT map components with INPUT ones that do not have the same maximum usage specified.

Use this procedure to specify the maximum usage for the Comments and LineItem elements.

Step	Action
1	On the OUTPUT side of the inbound map, right-click the Comments element (located within the Header element) and select **Properties** from the shortcut menu. **System response** The system displays the XML Element Properties dialog box.
2	Select the **Repeating** tab.
3	Select the **Can repeat, with a maximum usage** option.
4	In the Maximum Usage box, type **999999**.
5	Click **OK** to change the maximum usage of the Comments element.
6	On the OUTPUT side of the inbound map, right-click the LineItem element and select **Properties** from the shortcut menu. **System response** The system displays the XML Element Properties dialog box.
7	Select the **Repeating** tab.
8	Select the **Can repeat, with a maximum usage** option.
9	In the Maximum Usage box, type **999999**.
10	Click **OK** to change the maximum usage of the LineItem element.

Finalizing the Maps

Overview

Introduction

After you create the inbound and outbound maps and perform the appropriate mapping operations, you need to finalize the them. To complete the mapping process, you need to save the maps, compile the translation objects, print and review the Gentran:Server for Windows reports, and test the maps.

How to Compile a Map

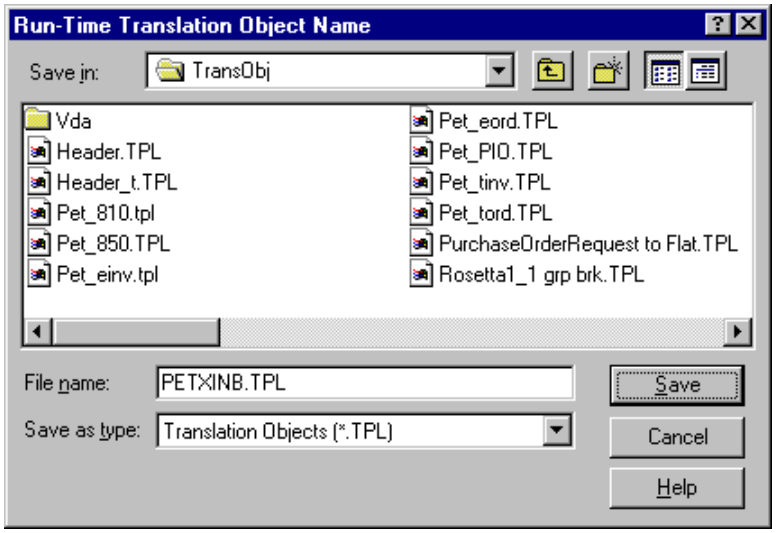
Overview

The Compile function compiles the map and generates a translation object. The map that you created using Gentran:Server for Windows 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:Server for Windows system before you can use it.

Procedure

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

Step	Action
1	From the File menu, select **Save** to save the source map prior to using the Compile function.
2	From the File menu, select **Compile** to display the Run-Time Translation Object Name dialog box.  (Continued on next page)

(Contd) Step	Action
3	In the File name box for the inbound map, type **Pettest Inbound XML Invoice.tpl**. In the File name box for the outbound map, type **Pettest Outbound XML Invoice.tpl**. **Note** This is the name of the translation object, using the default .TPL file extension and we recommend that you name the translation object (.TPL file) the same file name (1-8 characters long) as you named the map (.MAP file). Preserving the same file name (with different file extensions) means that the relationship between the source map and the compiled translation object remains evident. **Caution** Do not overlay the source map with the compiled translation object. Use the .TPL file extension to distinguish the translation object.
4	Change the Drive and Folder where the compiled translation object is stored, if necessary. **Caution** Do not store the compiled translation object in the GENSRVNT\RegTransObj subfolder. This subfolder is reserved for storing a copy of each translation object you register with Gentran:Server for Windows.
5	Click **Save** and the system compiles the map and generates a translation object. The Compile Error dialog box is displayed. Verify that no errors occurred (scroll down to the bottom of the list). Click **OK** to exit the dialog box. The date on which the translation object was compiled is automatically loaded into the Compiled on box on the Translation Object Details dialog box.
6	From the File menu, select **Save** to save the source map with the Compiled on date. **Note** You must register this translation object with the Gentran:Server for Windows system before you can use it. **Reference** See the Gentran:Server for Windows *User Guide* for more information on registering a translation object.

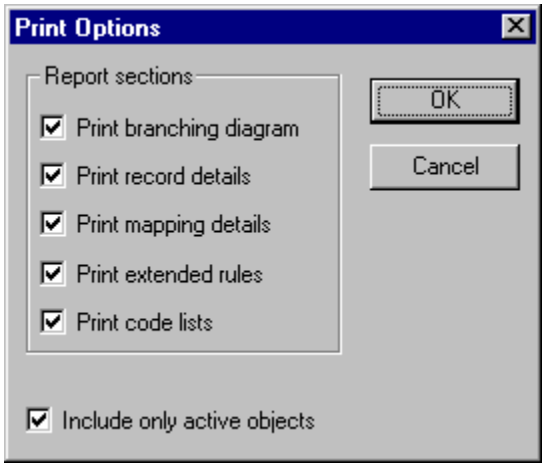
How to Print the Report

Overview

The Gentran:Server for Windows report enables you to validate and review the map, and make modifications as needed. If you review the report and determine that the map is incorrect, you should refine the mapping process as many times as necessary (make the modifications, save the map, recompile the translation object, and print the report again).

Procedure

Use this procedure to print the Gentran:Server for Windows report.

Step	Action
1	From the File menu select **Print** to access the Print Options dialog box. 
2	Accept the defaults on this dialog box (all options checked), so the report includes all sections and only the activated map components. Click **OK** and the Print dialog box 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 box is displayed. (Continued on next page)

(Contd) Step	Action
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 box. Click **OK** on the Print dialog box and the mapping report is printed. Click **Options** if you need to access printer setup options information. The Options dialog box is displayed.
5	Set the appropriate options. Click **OK** to return to the Print Setup dialog box. Click **OK** on the Print Setup dialog box to return to the Print dialog. Click **OK** on the Print dialog box and the mapping report is printed.

Testing the Translation Objects

Overview

After you compile the map, print and verify the report, and register the translation object with Gentran:Server for Windows, you should test the translation object to verify that the data is translated correctly. To test the compiled translation object, you should obtain test data from your partners and process the data. You should also verify acknowledgement processing (if applicable) and verify communications with your trading partner.

Inbound procedure

Use this procedure to test the inbound (Export) translation object you just created.

Reference
See the Gentran:Server for Windows *User Guide* for more information on registering a translation object, importing a partner, selecting inbound translation objects, and exporting files.

Step	Action
1	Register the following translation objects with Gentran:Server: ◗ Pettest Inbound XML Invoice.TPL ◗ XML_INT_BRK.TPL The default full path for the break translation object is: `C:\GENSRVNT\TUTORIAL\XML\XML_INT_BRK.TPL`
2	On the System Configuration Splitter tab, create a new splitter configuration entry with the following parameters: ◗ Type: XML ◗ Start Tag: PETTEST_INVOICE ◗ Interchange Break: XML_INT_BRK ◗ Functional Group Break: <none> ◗ Transaction Break: XML_TRN_BRK ◗ F/A Extract: <none> And click **Apply** and then **OK** to exit the System Configuration program. **Reference** See How to Define a New Splitter Entry in the *Administration Guide* for more information.
3	Import the PETZONE5.PAR partner relationship into Gentran:Server. The default full path for the partner file is: `C:\GENSRVNT\TUTORIAL\XML\PETZONE5.PAR` (Continued on next page)

(Contd) Step	Action
4	Verify (in Partner Editor) that the Pettest Outbound XML Invoice Export translation object is selected for the Inbound Relationship.
5	Use the Process File option in Gentran:Server for Windows to process the data file (Inbound_XML_Invoice.INT) through the translation object. The data file is located in the TUTORIAL\XML subfolder under the folder where Gentran:Server for Windows is installed. The default full path for the data file is: `C:\GENSRVNT\TUTORIAL\XML\Inbound_XML_Invoice.INT`
6	After the document is translated, it is located in the In Documents in Gentran:Server for Windows. Export the document to ensure that it was translated correctly.

Outbound procedure

Use this procedure to test the system import (HEADER.TPL) and import translation objects.

Reference
See the Gentran:Server for Windows *User Guide* for more information on registering a translation object, importing a partner, selecting inbound translation objects, and exporting files.

Step	Action

1	Register the following translation objects with Gentran:Server:
	◗ Pettest Outbound XML Invoice.TPL
	◗ XML_INT_BLD.TPL
	◗ XML_TRN_BLD.TPL
	
	The default full path for the build translation objects is:
	`C:\GENSRVNT\TUTORIAL\XML\XML_INT_BLD.TPL`
	`C:\GENSRVNT\TUTORIAL\XML\XML_TRN_BLD.TPL`

2	Register the Header.TPL and Pettest Outbound XML Invoice.TPL translation objects with Gentran:Server. **Note** The default full path for the header translation object is: `C:\GENSRVNT\TUTORIAL\HEADER.TPL` (Continued on next page)

Step	Action
3	If you have not already done so, import the PETZONE5.PAR partner relationship into Gentran:Server. The default full path for the partner file is: `C:\GENSRVNT\TUTORIAL\XML\PETZONE5.PAR`
4	Verify (in Partner Editor) that the Pettest Outbound XML Invoice Import translation object is selected for the Outbound Relationship.
5	Ask your system administrator to add the system import translation object (Header.TPL) to the System Configuration program (Imports tab) with the following parameters: ◗ File path: <drive>:\GENSRVNT\Tutorial\XML*.int ◗ Translation Object: System Import HDR And click **Apply** and then **OK** to exit the System Configuration program. **Reference** See How to Define a New Import Specification in the *Administration Guide* for more information about how your system administrator must modify the system configuration program.
6	Use the Import option in Gentran:Server for Windows to process the data file (Outbound_XML_Invoice.TXT) through the translation object. The data file is located in the TUTORIAL subfolder under the folder where Gentran:Server for Windows is installed. The default full path for the data file is: `c:\GENSRVNT\TUTORIAL\XML\Outbound_XML_Invoice.TXT`
7	After the document is translated, it is located in the Workspace in Gentran:Server for Windows. View the EDI data to ensure that the document was translated correctly.
8	Post the document to the Out Documents browser.
9	Then in the Out Documents browser, highlight the Tutorial document, click **View**, and after viewing the document click **Send**.