

IBM Sterling Gentran:Server for Windows



XML User Guide

Version 5.3.1

IBM Sterling Gentran:Server for Windows



XML User Guide

Version 5.3.1

Note

Before using this information and the product it supports, read the information in "Notices" on page 81.

This edition applies to the 5.3.1 version of IBM Sterling Gentrans:Server for Microsoft Windows and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1996, 2012.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Sterling Gentra:Server XML

Overview	1
XML with Sterling Gentra:Server Overview	1
XML Map Objects	2
Inbound XML Translation Encoding Support	3
Outbound XML Translation Encoding Support	4
Data Definition Format Overview	5
Importing a DDF	6
Exporting a DDF	6

Chapter 2. Using XML with Sterling Gentra:Server.

XML Process Overview	9
Translator Functionality Changes for XML Data	9
Modifying the Affected Maps	10
Modifying the Splitter Configuration	10
About the New XML Wizard	11
Map Types.	11
Creating an XML Map.	12
About Creating Map Objects.	15
About XML File Objects	15
XML File Properties	16
XML File Properties - Name Tab	16
XML File Properties - Tag Tab	16
XML File Properties - Entities Tab	17
XML File Properties - Repeating Tab	17
XML File Properties - Output Tab	17
XML File Properties - Loop Extended Rules Tab	18
XML File Properties - Decimal Point Tab.	19
Modifying XML File Properties.	19
Entity Properties.	20
Entity Properties - Name Tab	20
Entity Properties - Tag Tab	20
Entity Properties - Entity Tab	21
Creating an Entity	21
XML Element Properties	21
XML Element Properties - Name Tab	21
XML Element Properties - Tag Tab.	22
XML Element Properties - Key Field Tab.	22
XML Element Properties - Repeating Tab	23
XML Element Properties - Looping Tab	24
Creating an Element	24
Content Particle Properties	25
Content Particle Properties - Name Tab	25
Content Particle Properties - Type Tab	26
Content Particle Properties - Repeating Tab.	26
Content Particle Properties - Loop Extended Rules Tab	26
Creating a Content Particle	27
Pcdata Properties	28
Pcdata Properties - Validation Tab	28
Pcdata Properties - Extended Rule Tab	29
Pcdata Properties - Standard Rule Tab	29
Creating a pcdata	29
Attribute Properties	30

Attribute Properties - Name Tab	30
Attribute Properties - Tag Tab	31
Attribute Properties - Type Tab	31
Attribute Properties - Validation Tab	32
Attribute Properties - Extended Rule Tab	33
Attribute Properties - Standard Rule Tab.	33
Creating an Attribute	33

Chapter 3. XML Build and Break Maps 35

About XML Build and Break Maps	35
About XML Build Maps	36
Pass-Through Build (No Build) Maps.	37
Modifying the Input Side of Pass-through Build Maps	37
Modifying the Output Side of Pass-through Build Maps	38
Modifying the Map Details for Pass-through Build Maps	39
Enveloping Build Maps	40
Modifying the Input Side of Enveloping Build Maps	41
Modifying the Output Side of Enveloping Build Maps	41
Modifying the Map Details for Enveloping Build Maps	43
About XML Break Maps	44
Pass-Through Break Maps	45
Modifying the Input Side of Pass-Through Break Maps	46
Modifying the Output Side of Pass-Through Break Maps	47
Modifying the Map Details for Pass-Through Break Maps	48
Non-Pass-Through Break Maps.	49
Modifying the Input Side of Non-Pass-Through Break Maps	49
Modifying the Output Side of Non-Pass-Through Break Maps	51
Modifying the Map Details for Non-Pass-Through Break Maps.	51
Additional Tasks.	52
XML Partner Relationships	53

Chapter 4. XML Tutorial 55

About the XML Tutorial	55
Map Building Process	57
Data for the Tutorial	58
Inbound XML Invoice to Flat File	59
Creating the Inbound Tutorial Map	59
Correlating your Flat File and the XML Invoice	61
Invoice Map Components to Link	61
Additional Mapping Requirements	63
Performing Additional Mapping for the Input Side	64
Mapping the HEADER\STANDARD Field	65

Mapping the HEADER\TEST_PROD Field	65
Mapping the INVBEGIN\BILLTOACCTBEG Field.	66
Mapping the INVBEGIN\DOCTYPEBEG Field	66
Mapping the INVCOMMENT2\ BILLTOACCTCM2 Field	67
Mapping the INVCOMMENT2\DOCTYPECM2 Field.	67
Mapping the INVCOMMENT2\INVNUMCM2 Field.	68
Mapping the INVDETAIL\BILLTOACCTDET Field.	68
Mapping the INVDETAIL\DOCTYPEDET Field	68
Mapping the INVDETAIL\INVNUMDET Field	69
Mapping the INVSUMMARY\BILLTOACCTSUM Field.	69
Mapping the INVSUMMARY\DOCTYPESUM Field.	70

Outbound Flat File to XML Invoice	70
Creating the Outbound Tutorial Map	71
Correlating your Outbound Flat File and the XML Invoice	73
Outbound Map Components to be Linked	73
Performing Additional Mapping for the Output Side	75
About Finalizing a Map	76
Compiling a Map	76
Printing the Report.	77
Testing the Inbound Translation Object	78
Testing the Outbound Translation Object	78

Notices 81

Index 85

Chapter 1. Sterling Gentran:Server XML Overview

XML with Sterling Gentran:Server Overview

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.

IBM® Sterling Gentran:Server® for Microsoft Windows enables you to manage and translate documents, regardless of the document format (such as XML, traditional electronic commerce, your proprietary format).

The Sterling Gentran:Server XML implementation conforms to the rules of the XML language 1.0 specification, as published by the World Wide Web Consortium (except as specified below). In the interest of flexibility, Sterling 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. For example, you can define an address element twice—once under Ship To and once under Bill To.

Sterling Gentran:Server supports XML with the following specifications:

- The XML document must meet the well-formed document criteria that is specified for XML.

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

- Sterling 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, Sterling 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, Sterling Gentran:Server ignores the items listed as not supported.
- Sterling Gentran:Server can read UTF-8 and UTF-16 encoded files if a compatible code page is loaded on your computer.

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 also available when you right-click a map object.

XML Map Objects

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

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

Icon	Description
	<p>The XML File icon represents the XML document that Sterling 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.</p>
	<p>An XML element contains related elements and/or content particles. In addition, an element can contain one pcddata 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.</p> <p>A repeating element that contains another repeating element corresponds to a nested looping structure.</p>
	<p>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 pcddata. 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.</p> <p>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).</p>
	<p>A pcddata object contains character data. Only one pcddata object can be defined per element or content particle.</p> <p>Sterling Gentran:Server automatically names the pcddata object with the name of the parent element or content particle.</p> <p>When a pcddata has an operation performed against it (link, standard rule, or as an extended rule storage field), the system displays a red checkmark over the pcddata icon.</p>
	<p>An attribute container object does not correspond to an XML feature. Sterling Gentran:Server uses attribute container objects to contain the attributes of an XML element. This object has no properties.</p> <p>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.</p>
	<p>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.</p> <p>Note: Attributes do not have to occur in sequence in the input data.</p> <p>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.</p>

Inbound XML Translation Encoding Support

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

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.

For example: `<?xml version="1.0" encoding="Big5"?>`

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
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

Outbound XML Translation Encoding Support

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

You configure the encoding in the properties of the root tag of the XML map. See [About XML File Objects](#) for more information.

- UTF-8 - Unicode Transformation Format, 8-bit encoding form
- UTF-16 - Unicode Transformation Format, 16-bit encoding form
- Default - 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.

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

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

Default Encoding	Windows Code Page
UTF-7	65000
UTF-8	N/A
UTF-16	N/A
Shift-Jis	932
Big5	950
GB2312	936
dos-862	862
dos-720	720
EUC-JP	51932
HZ-GB-2312	52936
ISO-8859-1	1252
ISO-8859-2	28592
ISO-8859-4	28594
ISO-8859-5	28585
ISO-8859-7	28597
ISO-8859-9	28599
ISO-2022-JP	50220
ISO-2022-KR	50225
KOI8-R	20866
KSC_5601	949
Windows-874	874
Windows-1250	1250
Windows-1251	1251
Windows-1252	1252
Windows-1253	1253

Default Encoding	Windows Code Page
Windows-1254	1254
Windows-1255	1255
Windows-1256	1256
Windows-1257	1257
Windows-1258	1258

Data Definition Format Overview

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

Sterling 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 output file formats.
See *Creating an XML Map* 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).
See *Importing a DDF* for more information on loading a DDF file definition.

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

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 (such as groups, records, and fields) and their attributes (such as names, descriptions, and data types). The DDF is expressed in XML, as defined in the Gentran_DDF DTD.

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

DDF restrictions

Sterling 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, Sterling Gentran:Server saves the standard rule. Sterling Gentran:Server saves every code list and constant, regardless of whether it is used in a rule.
See the *IBM Sterling Gentran:Server for Microsoft Windows Application Integration User Guide* for more information on code lists and the Use code standard rule.
- To use DDF with Sterling 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, Sterling Gentran:Server cannot support DDF and will not display it as an available option.

- Sterling Gentran:Server assigns acceptable defaults when attributes are not included.
For example: If a value is not specified for Minimum Length, the system assigns zero.

DDF Document Type Definition

The Sterling Gentran:Server Data Definition Format DTD (Gentran_DDF) describes how Sterling 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 DVD.

Importing a DDF

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

About this task

See Exporting a DDF for more information on saving a file definition.

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

Use this procedure to import a DDF.

Procedure

1. Right-click the File Format icon (either the input or output side of the map) and select **Open File Definition**.

Note: If you already used Sterling 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.

The system displays the Open File Definition dialog box.

2. Select DDF as the default file extension and navigate to or enter the file definition you want to load.
3. Click **Open**.

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.

Exporting a DDF

Sterling 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.

About this task

See Importing a DDF for more information on loading a file definition.

Use this procedure to export a DDF.

Procedure

1. Right-click the File Format icon (either the input or output side of the map) and select **Save File Definition**.

The system displays the Save File Definition dialog box.

2. Select DDF as the default file extension and navigate to or enter the file definition you want to save.

3. Click **Save**.

The system saves the file format definition.

Chapter 2. Using XML with Sterling Gentran:Server

XML Process Overview

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

Stage	Description
1	Install the Sterling Gentran:Server XML option on every machine in your system.
2	Define the XML standards that you and your trading partners will use. Make sure it is installed on every machine in your system.
3	Create a splitter entry to identify and extract XML documents. See Modifying the Splitter Configuration for more information.
4	Create a map to translate XML documents. See Creating an XML Map for more information.
5	If necessary, build the input and output sides of the map using one of these methods: <ul style="list-style-type: none">• Load the data format from a saved definition. See Importing a DDF for more information.• Create map objects manually. See About XML File Objects for more information.• Load a map side with the New Map Wizard Customize option. See Creating an XML Map for more information.
6	Determine if the Sterling Gentran:Server translator changes require you to modify your maps. See Translator Functionality Changes for XML Data for more information.

Translator Functionality Changes for XML Data

This topic describes the translator changes implemented with Sterling Gentran:Server version 5.2. 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 skips it and continues processing the rest of the data.

This differs in behavior from previous Sterling 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).

This 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.

See “Modifying the Affected Maps” for information about how to modify those types of maps.

Modifying the Affected Maps

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

About this task

Use this procedure to modify the affected maps.

Procedure

1. On the Pre-Session On-Begin rules, create and initialize a global integer variable. This variable is used to store the file position from the point in the Input file where the ReadBlock/WriteBlock should be performed.

For example:

```
Integer Input_pos;
Input_pos = 0;
```

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. This stores the file position immediately following the end tag of that element.

For 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. This places the file pointer back to the correct position in the Input file so the ReadBlock/WriteBlock can be performed. For more information on the Ftell/Fseek extended rule functions, please see Alphabetical Language Reference in the *Application Integration User Guide*.

For example:

```
Fseek(0,Input_pos,current);
```

Modifying the Splitter Configuration

Sterling 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.

About this task

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.

Before receiving inbound XML data, you need to create a Sterling 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.

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

Procedure

1. From any machine on which the System Configuration program is installed, start the **System Configuration** program.
The system displays the System Configuration dialog box (Controllers tab).
2. Select the **Splitter** tab.
The splitter parameters are displayed.
3. Click **New**.
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.

About the New XML Wizard

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 (such as a DTD).

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 5.2 or greater installed on your machine.

Map Types

The following table defines the map types:

Map Type	Function
Import	Used for outbound maps.
Export	Used for inbound maps.
Turnaround	Used for EDI to EDI maps.
Transaction build	Used in advanced mapping to build transaction envelopes.
Transaction break	Used in advanced mapping to separate documents.

Map Type	Function
Functional group build	Used in advanced mapping to build functional group envelopes.
Functional group break	Used in advanced mapping to separate functional groups.
Interchange build	Used in advanced mapping to build interchange envelopes.
Interchange break	Used in advanced mapping to separate interchanges.
F/A Inbound	Used in advanced mapping to reconcile functional acknowledgements.
F/A Outbound	Used in advanced mapping to generate functional acknowledgements.

Creating an XML Map

The general steps for creating a map are the same, but the selections you make depend on what type of map you are creating.

About this task

Use this procedure to create an XML map.

Procedure

1. Select **File > New**.

The system displays the New Map Wizard.

2. Enter the following information and click **Next**:

- Select the map type.
- Type the unique name of the map. The system adds the .MAP extension.
- Type your name if it differs from the user name prompted by the system.

Note: You need to complete the format of the Input side of the map (steps 3 - 9). This is the format of the data that is translated by the Sterling Gentran:Server system.

3. For the input side of the map, do one of the following:

- To create a new data format using a syntax that you define, go to step 4.
- To load the data format from a saved definition, go to step 8.

4. Select one of the following input format options:

- **Delimited EDI** (Electronic Data Interchange file)
- **ODBC** (Open Database Connectivity)
- **Positional** (VDA, GENCOD, application files)
- **XML** (Extensible Markup Language)

Notes:

- 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.

5. Do one of the following:

- To customize the format of Delimited EDI map, click **Customize**. The system displays the New Delimited EDI Wizard. Continue with step 6.
 - To customize the format of Delimited EDI or XML map, click **Customize**. The system displays the New XML Wizard. Continue with step 7.
 - Otherwise, click **Next** and continue with step 10.
6. For Delimited EDI maps, do the following in the 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**.

Continue with step 9.

7. For XML maps, do the following in the New XML Wizard:
 - 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**.
 - d. Click **Finish**.

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

Continue with step 10.

8. Select **Load the data format from a saved definition**.
9. Navigate to or type the path and filename of the saved definition and click **Open**.

Note: You can select either a .DDF or .IFD file. If the DDF is invalid, the system displays a message box explaining the problem and terminates the import.

10. For the output side of the map, do one of the following:
 - Create a new data format using a syntax that you define, go to step 11.
 - To load the data format from a saved definition, go to step 15.
11. Select one of the following output format options:
 - **Delimited EDI** (Electronic Data Interchange file)
 - **ODBC** (Open Database Connectivity)
 - **Positional** (VDA, GENCOD, application files)
 - **XML** (Extensible Markup Language)

Notes:

- 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.
12. Do one of the following:
 - To customize the format of Delimited EDI map, click **Customize**. The system displays the New Delimited EDI Wizard or New XML Wizard dialog box. Continue with step 13.
 - To customize the format of Delimited EDI or XML map, click **Customize**. The system displays the New Delimited EDI Wizard or New XML Wizard dialog box. Continue with step 14.
 - Otherwise, go to Step 17.
 13. For Delimited EDI maps, do the following in the 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).
 - d. Click **Next**.
 - e. Click **Finish**.

Continue with Step 17.
 14. For XML maps, do the following in the New XML Wizard:
 - 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**.
 - d. Click **Finish**.

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

Continue with Step 17.

15. Select **Load the data format from a saved definition**.
16. Navigate to or type the file name and click **Open**.

Note: You can select either a .DDF or .IFD file.

17. Click **Finish** to load the standards information you selected and create the new map (this may take a few seconds).
The system displays the new map in the Application Integration Window.

What to do next

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.

About Creating Map Objects

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	<ul style="list-style-type: none">• Element• Content Particle• PCDATA• Attribute
Element	<ul style="list-style-type: none">• Element• Content Particle• PCDATA• Attribute
Content Particle	<ul style="list-style-type: none">• Element• Content Particle• PCDATA
PCDATA	N/A
Attribute Container	Attribute
Attribute	N/A

Create Sub vs. Insert functions

You use two different Sterling 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 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.

About XML File Objects

The XML File object represents the XML document that Sterling Gentran:Server is mapping, including the root element. This object is created automatically by Sterling 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: This dialog is accessible through the XML File Properties dialog box.

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

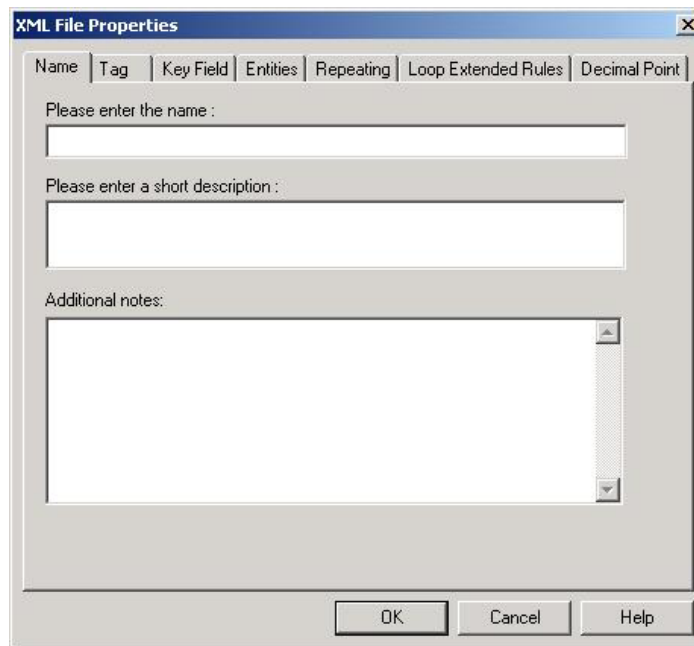
Sterling Gentran:Server allows you to define internal general parsed entities, according to the XML definition.

XML File Properties

XML File Properties - Name Tab

The Name tab of the XML File Properties dialog box allows you to provide names and descriptions for your XML files.

This diagram illustrates the XML File Properties dialog box Name tab.



This table describes the parts of the XML File Properties dialog box Name tab.

Part	Function
Name	Identifies the XML file.
Description	Describes the XML file. This box is used to differentiate the XML file from similar files.

XML File Properties - Tag Tab

The Tag tab of the XML File Properties dialog box allows you to identify the XML tags.

This table describes the parts of the XML File Properties dialog box Tag tab.

Part	Function
Tag	Identifies the XML tag for the root element of the document. Default is INPUT or OUTPUT.

XML File Properties - Entities Tab

The Entities tab of the XML File Properties dialog box allows you to view a list of entities and to add, change, delete, and copy entities.

This table describes the parts of the XML File Properties dialog box Entities tab.

Part	Function
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.

XML File Properties - Repeating Tab

The Repeating tab of the XML File Properties dialog box allows you to specify which objects are conditional and which are mandatory.

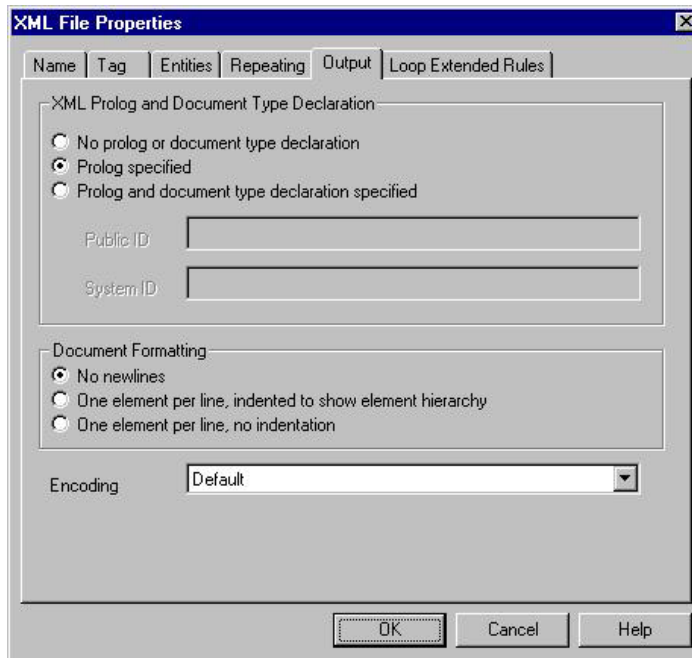
This table describes the parts of the XML File Properties dialog box Repeating tab.

Part	Function
Conditional	Indicates that the XML file object is not mandatory.
Mandatory	Indicates that the XML file object must appear in the map.

XML File Properties - Output Tab

The Output tab of the XML File Properties dialog box allows you to specify the prolog, DTD, document formatting, and encoding for the output side of the map.

This diagram illustrates the XML File Properties dialog box Name tab.



This table describes the parts of the XML File Properties dialog box Output tab (appears only for the output side of the map).

Part	Function
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.
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.

XML File Properties - Loop Extended Rules Tab

The Loop Extended Rules tab of the XML File Properties dialog box allows you to define extended rules and specify when they are processed.

This table describes the parts of the XML File Properties dialog box Loop Extended Rules tab.

Part	Function
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.

XML File Properties - Decimal Point Tab

The Decimal Point tab of the XML File Properties dialog box allows you to specify what character you want to use as a decimal point.

This table describes the parts of the XML File Properties dialog box Decimal Point tab.

Part	Function
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

About this task

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

Procedure

1. Right-click the XML File icon and select **Properties**.
The system displays the XML File Properties dialog box.
2. To create an entity, select the **Entities** tab. See *Creating an Entity* for more information.
3. To modify the output options for the XML file, select the **Output** tab.
Otherwise, 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
5. To specify an extended rule for the XML file, select the **Loop Extended Rules** tab and define the rule. See the *Application Integration User Guide* for more information on extended rules.

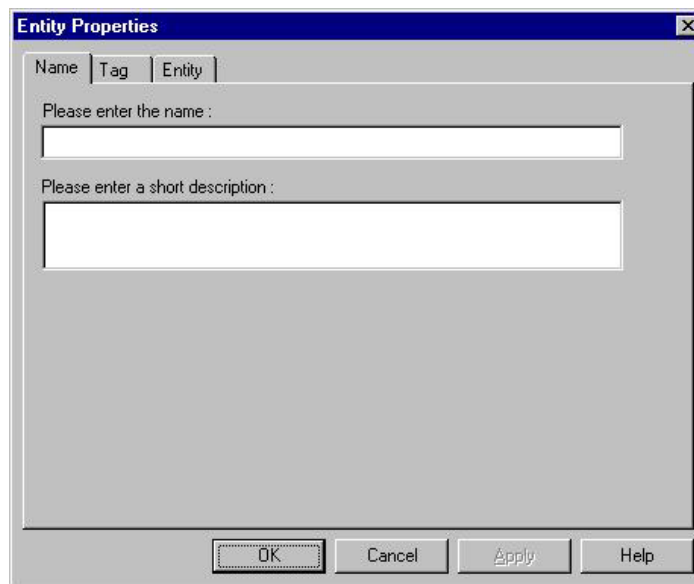
6. To specify that each decimal point in your data is defined and generated as a something other than the default period, do the following:
 - Select the **Decimal Point** tab.
 - Select the **Define Decimal Point** check box.
 - In the Decimal Point Character box, type the value you want to use as a decimal point—for example, a comma.
 - Click **OK**. You must click OK, even if the delimiter fields are empty, to reset the delimiters.
7. Click **OK**.
The system saves your changes and closes the XML File Properties dialog box.

Entity Properties

Entity Properties - Name Tab

The Name tab of the Entity Properties dialog box allows you to provide names and descriptions for the map entities.

This diagram illustrates the Entity Properties dialog box Name tab.



This table describes the parts of the Entity Properties dialog box Name tab.

Part	Function
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.

Entity Properties - Tag Tab

The Tag tab of the Entity Properties dialog box allows you to define entity tags.

This table describes the parts of the Entity Properties dialog box Name tab.

Part	Function
Tag	Defines the entity identification tag, as it appears in the XML document. The system uses the entity name by default.

Entity Properties - Entity Tab

The Entity tab of the Entity Properties dialog box allows you to define the entity value.

This table describes the parts of the Entity Properties dialog box Name tab.

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

Creating an Entity

About this task

Use this procedure to create an entity.

Procedure

1. Right-click the XML File icon and select **Properties**.
The system displays the XML File Properties dialog box.
2. Select the **Entities** tab to access the entity options.
3. Click **New**.
The system displays the Entity Properties dialog box.
4. On the **Name** tab, enter a unique entity name and a 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.
7. In the Entity Value box, type the entity data.
8. Click **OK**.
The system saves the entity and closes the Entity Properties dialog box.
9. Click **OK**.
The system closes the XML File Properties dialog box.

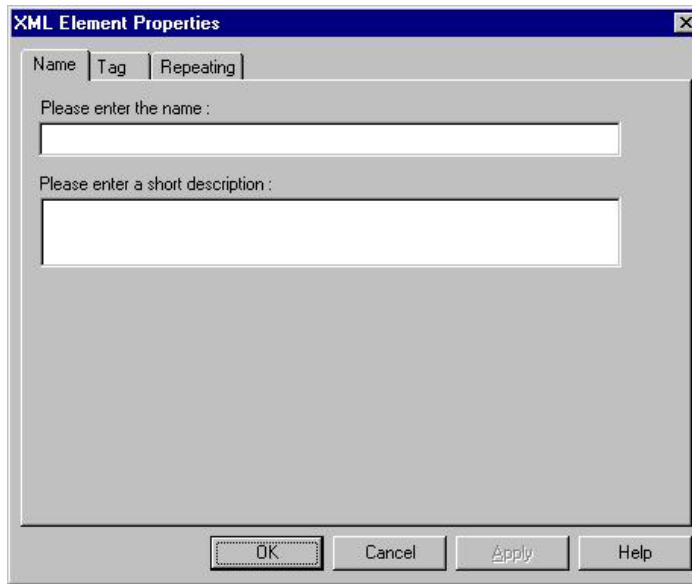
XML Element Properties

XML Element Properties - Name Tab

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.

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



This table describes the parts of the XML Element Properties dialog box Name tab.

Part	Function
Name	<p>Defines the element name.</p> <p>Notes:</p> <ul style="list-style-type: none"> • 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	<p>Describes the element. This box is used to provide a brief explanation of the element that allows you to differentiate it from similar elements.</p>

XML Element Properties - Tag Tab

The Tag tab of the XML Element Properties dialog box allows you to define element tags.

This table describes the parts of the XML Element Properties dialog box Tag tab.

Part	Function
Tag	<p>Defines the element identification tag, as it appears in the XML document.</p> <p>Sterling Gentran:Server validates the tag against the characters that XML allows for element names.</p> <p>The system uses the element name by default.</p>

XML Element Properties - Key Field Tab

The Key Field tab of the XML Element Properties dialog box appears only if the element contains attributes.

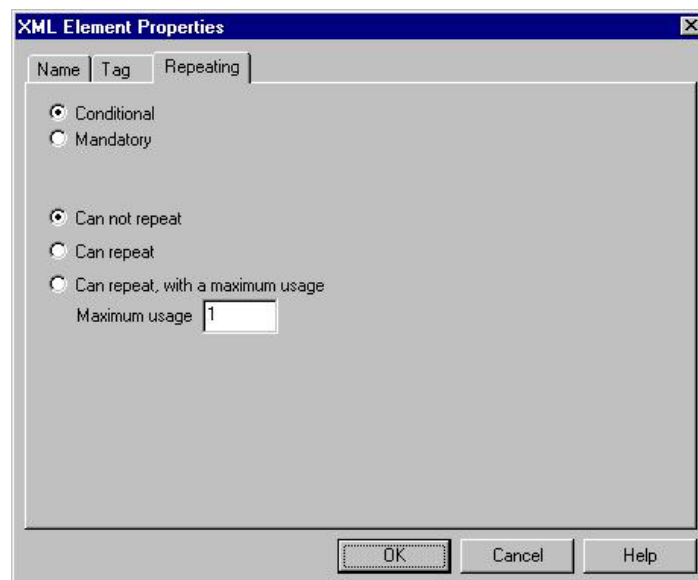
This table describes the parts of the XML Element Properties dialog box Key Field tab.

Part	Function
Field	Specifies all the attributes that are defined for this element. 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. Click Edit (at the right of the Use constant list) to access the Translation Object Constants dialog box.
Use codelist/Edit	Indicates that the system must match the element if the contents of the selected attribute match the selected code list. 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. If the specified condition is not met, the element does not conform to the definition, and processing continues.

XML Element Properties - Repeating Tab

The Repeating tab of the XML Element Properties dialog box allows you to specify which elements are conditional and which are mandatory.

This diagram illustrates the Element Properties dialog box Repeating tab.



This table describes the parts of the XML Element Properties dialog box Repeating tab.

Part	Function
Conditional	Indicates that the element is not mandatory.
Mandatory	Indicates that the element must appear in the map.

Part	Function
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).

XML Element Properties - Looping Tab

The Looping tab of the XML Element Properties dialog box appears only if the element repeats. This tab allows you to define extended rules and specify when they are processed.

This table describes the parts of the XML Element Properties dialog box Looping tab.

Part	Function
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. 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 an Element

About this task

Use this procedure to create an element.

Procedure

1. Right-click a map object and select either **Create Sub > Element** or **Insert > Element**.

See About Creating Map Objects for more information on the Create Sub and Insert functions.

The system displays the XML Element Properties dialog box.

2. On the **Name** tab, enter a unique element name and description (if applicable).
3. 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.

4. Select the **Repeating** tab.

5. Select either the Conditional or Mandatory option to specify whether the element is required in the map.
6. Select the appropriate repeating option for the element.
7. If you need to specify the number of times the element can repeat (loop), type that number in the Maximum usage box.
8. For repeating elements (loops): To specify an extended rule for this element, select the **Loop Extended Rules** tab and define the rule.
See the *Application Integration User Guide* for more information on extended rules.
9. Click **OK**.
The system saves the element and closes the XML Element Properties dialog box.

Content Particle Properties

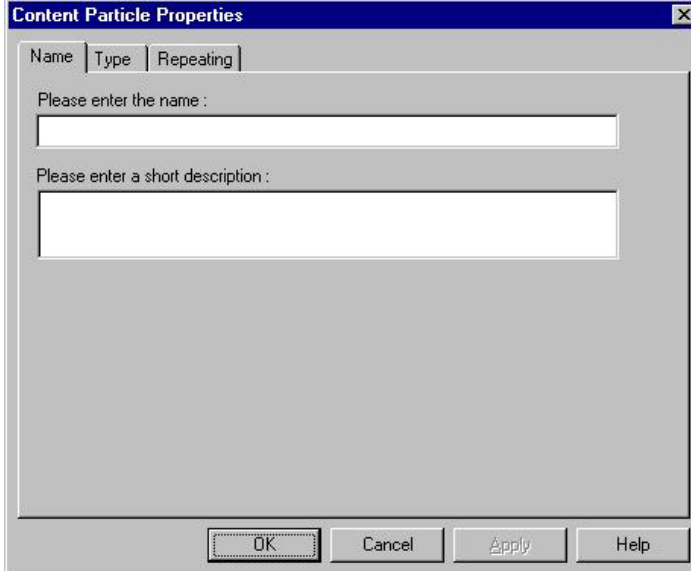
Content Particle Properties - Name Tab

In Sterling 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 pcddata. 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.

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



The screenshot shows a dialog box titled "Content Particle Properties" with a close button (X) in the top right corner. The dialog has three tabs: "Name", "Type", and "Repeating", with "Name" selected. Below the tabs, there are two text input fields. The first is labeled "Please enter the name:" and the second is labeled "Please enter a short description:". At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

This table describes the parts of the Content Particle Properties dialog box Name tab.

Part	Function
Name	Defines the name of the content particle. 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.

Content Particle Properties - Type Tab

The Type tab of the XML Element Properties dialog box allows you to define the type of child object.

This table describes the parts of the Content Particle Properties dialog box Type tab.

Part	Function
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.

Content Particle Properties - Repeating Tab

The Repeating tab of the Content Particle Properties dialog box allows you to specify which content particles are conditional and which are mandatory.

This table describes the parts of the Content Particle Properties dialog box Repeating tab.

Part	Function
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).

Content Particle Properties - Loop Extended Rules Tab

The Loop Extended Rules tab of the Content Particle Properties dialog box appears only if the element repeats. This tab allows you to define extended rules and specify when they are processed.

This table describes the parts of the Content Particle Properties dialog box Loop Extended Rules tab.

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

Creating a Content Particle

About this task

Use this procedure to create a content particle.

Procedure

1. Right-click a map object and select either **Create Sub > Content Particle** or **Insert > Content Particle**.
See About Creating Map Objects for more information on the Create Sub and Insert functions.
The system displays the Content Particle Properties dialog box.
2. On the **Name** tab, enter a unique content particle name and description (if applicable).
3. Select the **Type** tab to access the content particle type options.
4. Select the appropriate option to define what the child objects of the content particle represent.
5. Select the **Repeating** tab.
6. Select either Conditional or Mandatory to specify whether the content particle is required in the map.
7. Select the appropriate repeating option for the content particle.
8. In the Maximum usage box, type the number of times the content particle can repeat (loop).
9. For repeating content particles (loops): To specify an extended rule for this content particle, select the **Loop Extended Rules** tab and define the rule.
See the *Application Integration User Guide* for more information on extended rules.
10. Click **OK**.
The system saves the content particle and closes the Content Particle Properties dialog box.

Pcdata Properties

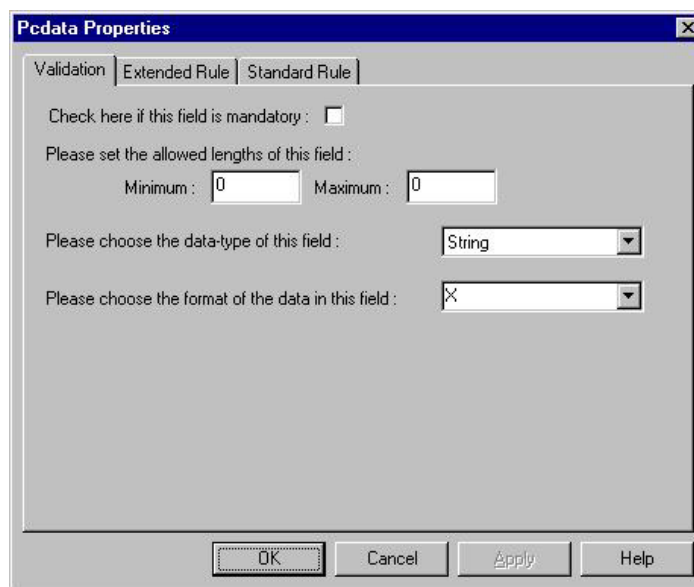
Pcdata Properties - Validation Tab

A pcdta object contains character data in an XML document. Only one pcdta object can be defined per element or content particle.

Sterling Gentran:Server automatically names the pcdta object with the name of the parent element or content particle.

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

This diagram illustrates the Pcdata Properties dialog box Validation tab.



This table describes the parts of the Pcdata Properties dialog box Validation tab.

Part	Function
Mandatory	Indicates whether the pcdta is mandatory.
Minimum	Specifies the minimum length of the pcdta.
Maximum	Specifies the maximum length of the pcdta.
Data-type	Specifies the type of data. Valid values are: <ul style="list-style-type: none">• String (alphanumeric element)• Number (numeric or real element)• Date/Time (date or time element)

Part	Function
Format	<p>Specifies how the pcddata is formatted.</p> <p>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.</p> <p>When you installed Sterling Gentran:Server, you assigned a default format to the string fields. This format serves as the basis for character validation. Most U.S. users use a default format that corresponds to ASCII characters (for example, the X syntax token). Most users of Asian or European languages and encoded character sets should use the Free Format (0x01-0xFF).</p>

Pcddata Properties - Extended Rule Tab

The Extended Rule tab of the Pcddata Properties dialog box allows you to define and compile extended rules.

This table describes the parts of the Pcddata Properties dialog box Extended Rule tab.

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

Pcddata Properties - Standard Rule Tab

The Standard Rule tab of the Pcddata Properties dialog box allows you to specify standard rules.

This table describes the parts of the Pcddata Properties dialog box Standard Rule tab.

Part	Function
Standard rule	Specifies a standard rule that will affect this field or element during processing. Each of the rules are mutually exclusive. See the <i>Application Integration User Guide</i> for more information on standard rules.

Creating a pcddata

About this task

Use this procedure to create a pcddata.

Procedure

1. Right-click a map object and select either **Create Sub > PCDATA** or **Insert > PCDATA**.
See About Creating Map Objects for more information on the Create Sub and Insert functions.
The system displays the PCDATA Properties dialog box.
2. On the **Validation** tab, specify the following:
 - whether the pcdData is required or not
 - minimum length
 - maximum length
 - type of data
 - how the data is formatted
3. To specify an extended rule for this pcdData, select the **Extended Rule** tab and define the rule.
See the *Application Integration User Guide* for more information on extended rules.
4. To specify a standard rule for this pcdData, select the **Standard Rule** tab and define the rule.
See the *Application Integration User Guide* for more information on standard rules.
5. Click **OK**.
The system saves the pcdData and closes the PCDATA Properties dialog box.

Attribute Properties

Attribute Properties - Name Tab

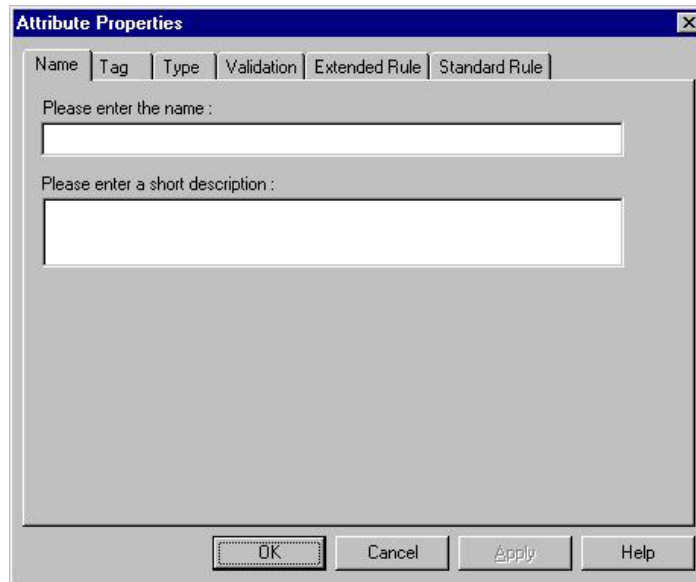
In Sterling Gentran:Server, each attribute is contained in an attribute container. An element can only have one attribute container object, but the attribute container object can enclose many attribute objects.

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.

The attribute container object does not correspond to an XML feature. Sterling 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.

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

This diagram illustrates the Attribute Properties dialog box Name tab.



This table describes the parts of the Attribute Properties dialog box Name tab.

Part	Function
Name	Defines the name of the attribute. Do not use spaces or dashes (-) for the record name. You can use the underscore (_) to separate words.
Description	Describes the attribute. This box is used to provide a brief explanation of the attribute that allows you to differentiate it from similar attributes.

Attribute Properties - Tag Tab

The Tag tab of the Attribute Properties dialog box allows you to define attribute tags.

This table describes the parts of the PCDATA Properties dialog box Tag tab.

Part	Function
Tag	Defines the attribute identification tag, as it appears in the XML document. Sterling 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.

Attribute Properties - Type Tab

The Type tab of the Attribute Properties dialog box allows you to define attribute tags, whether or not it is mandatory, and whether there is a default value.

This table describes the parts of the PCDATA Properties dialog box Type tab.

Part	Function
Attribute Type	<p>Specifies the type of data that can be used in this attribute:</p> <ul style="list-style-type: none"> CDATA - Character data (a string of characters). ENUMERATED - 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 not found in the list. ID - A valid and unique identifier. IDREF - A reference to a unique identifier. 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	<p>Indicates that a default value exists for this attribute.</p> <p>You must define the default value. If the incoming data does not contain a value for this attribute, Sterling Gentran:Server creates it with the default value.</p>
Fixed	<p>Indicates that the default value of this attribute is fixed (cannot be changed).</p> <p>You must define the default value. If the incoming data does not match this value, the document is not valid.</p>
Default value	Specifies the default value for the attribute.

Attribute Properties - Validation Tab

The Validation tab of the Attribute Properties dialog box allows you to define the minimum and maximum length, the data type, and the formatting of the attribute.

This table describes the parts of the PCDATA Properties dialog box Validation tab.

Part	Function
Minimum	Specifies the minimum length of the attribute.
Maximum	Specifies the maximum length of the attribute.
Data-type	<p>Specifies the type of data. Valid values are:</p> <ul style="list-style-type: none"> String (alphanumeric element) Number (numeric or real element) Date/Time (date or time element)

Part	Function
Format	<p>Specifies how the attribute is formatted.</p> <p>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.</p> <p>When you installed Sterling Gentran:Server, you assigned a default format to the string fields. This format serves as the basis for character validation. Most U.S. users use a default format that corresponds to ASCII characters (for example, the X syntax token). Most users of Asian or European languages and encoded character sets should use the Free Format (0x01-0xFF).</p>

Attribute Properties - Extended Rule Tab

The Extended Rule tab of the Attribute Properties dialog box allows you to define and compile extended rules.

This table describes the parts of the PCDATA Properties dialog box Extended Rule tab.

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

Attribute Properties - Standard Rule Tab

The Standard Rule tab of the Attribute Properties dialog box allows you to specify standard rules.

This table describes the parts of the Attribute Properties dialog box Standard Rule tab.

Part	Function
Standard rule	<p>Specifies a standard rule that will affect this field or element during processing. Each of the rules are mutually exclusive.</p> <p>See the <i>Application Integration User Guide</i> for more information on standard rules.</p>

Creating an Attribute About this task

Use this procedure to create an attribute.

Procedure

1. Right-click a map object and select either **Create Sub > Attribute** or **Insert > Attribute**.
See *About Creating Map Objects* for more information on the Create Sub and Insert functions.
The system displays the Attribute Properties dialog box.
 2. On the **Name** tab, enter a unique attribute name and description (if applicable).
 3. If necessary, select the **Tag** tab and change the value in the Tag box.
This must match the attribute tag in the XML document.
 4. On the **Type** tab, specify the following:
 - attribute type
 - default usage of the attribute
 - 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.
5. On the **Validation** tab, specify the following:
 - minimum length
 - maximum length
 - type of data
 - how the data is formatted
 6. To specify an extended rule for this attribute, select the **Extended Rule** tab and define the rule.
See the *Application Integration User Guide* for more information on extended rules.
 7. To specify a standard rule for this attribute, select the **Standard Rule** tab and define the rule.
See the *Application Integration User Guide* for more information on standard rules.
 8. Click **OK**.
The system saves the attribute and closes the Attribute Properties dialog box.

Chapter 3. XML Build and Break Maps

About XML Build and Break Maps

Using the Sterling Gentran:Server XML may require you to create customized build and break maps that are specific to your XML document.

Depending on your XML layout (for example, if you use many doctypes), you may need to create a set of build and break maps for each doctype.

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 Sterling 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).

See the following for more information:

- Registering a New Translation Object in the *IBM Sterling Gentran:Server for Microsoft Windows User Guide*
- Modifying the Splitter Configuration

Use of Wildcard Segments in Build and Break Maps

In Sterling 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.

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

For 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 long. 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.

See the section on Using Extended Rules in the *Application Integration User Guide* for more information.

About XML Build Maps

The following topics describe 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.

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 Sterling 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
<p>The XML data created by your import map does not require enveloping.</p> <p>For example:</p> <p>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.</p>	<p>The XML data created by your import map requires enveloping.</p> <p>For example:</p> <p>You use enveloping build maps if the XML structure created by the import map is not complete.</p>

Outbound processing stages

This table describes the stages of outbound processing.

Stage	Description
1	<p>The system determines which system import map to use.</p> <p>See the <i>Application Integration User Guide</i> for more information on creating a system import map.</p> <p>See the <i>Administration Guide</i> for more information on setting up an XML import specification.</p>
2	<p>The system locates the partner and relationship, checks the data for compliance, and translates the import data file.</p> <p>See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for more information on how to set up an outbound partner relationship.</p>
3	<p>The system builds the outbound interchanges using data from the outbound partner relationship.</p> <p>See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for more information on how to set up an outbound partner relationship.</p>

Pass-Through Build (No Build) Maps

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.

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 Sterling 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	<ul style="list-style-type: none">• Map type = Transaction Build• Input format = XML• Output format = XML
interchange build map	<ul style="list-style-type: none">• Map type = Interchange Build• Input format = XML• Output format = XML

Modifying the Input Side of Pass-through Build Maps

For each map you create, you need to modify both the input and output sides.

About this task

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

Procedure

1. Right-click the XML File icon on the INPUT side of the map and select **Properties**.
The system displays the XML File Properties dialog box.
2. In the name box, type the name of the root element in your XML document.
3. Select the **Tag** tab.
4. In the Tag box, type the tag of the root element in your XML document.
5. Click **OK**.
The system saves your changes and closes the XML File Properties dialog box.
6. Right-click the XML File icon and select **Create Sub > Element**.
See About Creating Map Objects for more information on the Create Sub and Insert functions.
7. In the name box, type the name of the first element in your XML document that is passed to the build maps.
8. Select the **Repeating** tab.
9. Select **Can repeat**.
The system appends the Loop Extended Rules tab to the XML Properties dialog box.
10. Select the **Loop Extended Rules** tab.
11. In the extended rule box, do the following:
 - Accept the default On Begin option.
 - Type the readblock/writeblock extended rule.
 - Click **OK**.For 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);
```

See the *Application Integration User Guide* for more information on extended rules
The system saves your change and closes the XML Element Properties dialog box.
12. Right-click the element you created and select **Create Sub > PCDATA**.
13. Select the appropriate options based on your XML document and click **OK**.
The system saves your change and closes the PCDATA Properties dialog box.

Modifying the Output Side of Pass-through Build Maps

For each map you create, you need to modify both the input and output sides.

About this task

Use this procedure to modify each map.

Procedure

1. Right-click the XML File icon on the output side of the map and select **Properties**.
2. In the name box, type the name of the Root Element in your XML document.
3. On the **Tag** tab, type the tag of the Root Element in your XML document.
4. On the **Repeating** tab, select **Conditional**.

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.

5. Select the **Output** tab.
6. For a transaction build map, do the following:
 - In the XML Prolog and Document Type Declaration section, select **No prolog**. 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 **No newlines**.
7. For an interchange build map, do the following:
 - 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.
8. Click **OK**.

The system saves your changes and closes the XML File Properties dialog box.
9. Right-click the XML File icon and select **Create Sub > Element**.

Note: You need to create a temporary XML element and pcddata so the map will compile correctly. See About Creating Map Objects for more information on the Create Sub and Insert functions.

The system displays the XML Element Properties dialog box.

10. In the name box, type **TEMP**.
11. Click **OK**.

The system saves the temporary element.
12. Right-click the element you created and select **Create Sub > Pcddata**.

The system displays the Pcddata Properties dialog box.
13. Click **OK**.

The system closes the Pcddata Properties dialog box.

Modifying the Map Details for Pass-through Build Maps

For each map you created, you also need to use the Translation Object Details dialog box to modify details of the maps.

About this task

Use this procedure to modify each map.

Procedure

1. Select **Edit > Details**.

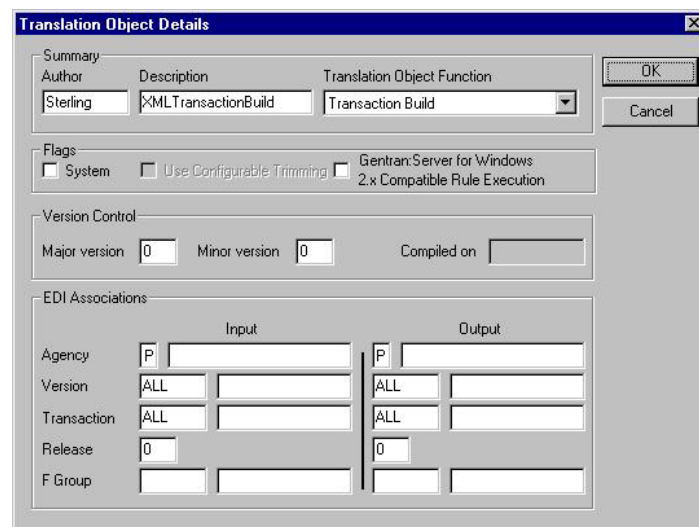
The system displays the Translation Object Details dialog box.

- In the EDI Associations section, type a letter that is not currently used to designate a standards agency (for example, 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 will cause the map to select incorrect Partner Editor build objects.
- In the EDI Associations section, type **ALL** in the first Version box for both Input and Output.
 - In the EDI Associations section, type **ALL** in the first Transaction box for both Input and Output.

This diagram illustrates how the EDI Associations section of the Transaction Object Detail dialog box should look.



- Click **OK**.
The system saves your changes and closes the Translation Object Details dialog box.

Enveloping Build Maps

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 Sterling Gentran:Server database tables at the current level of processing (for example, you can only update the Interchange_tb with an Interchange Break or Build Map).

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 Sterling 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	<ul style="list-style-type: none"> • Map type = Transaction Build • Input format = XML • Output format = XML
interchange build map	<ul style="list-style-type: none"> • Map type = Interchange Build • Input format = XML • Output format = XML

Modifying the Input Side of Enveloping Build Maps

For each map you create, you need to modify both the input and output sides.

About this task

Use this procedure to modify each map.

Procedure

1. Right-click the XML File icon on the input side of the map and select **Properties**.
The system displays the XML File Properties dialog box.
2. In the name box, type the name of the Root Element in your XML document.
3. Select the **Tag** tab.
4. In the Tag box, type the tag of the Root Element in your XML document.
5. Click **OK**.
The system saves your changes and closes the XML File Properties dialog box.
6. Right-click the XML File icon and select **Create Sub > Element**.
See About Creating Map Objects for more information on the Create Sub and Insert functions.
The system displays the XML Element Properties dialog box.
7. In the name box, type the name of the first Element in your XML document that is passed to the build maps.
8. Click **OK**.
The system saves your change and closes the XML Element Properties dialog box.
9. Right-click the element you created and select **Create Sub > PCDATA**.
The system displays the PCDATA Properties dialog box.
10. Select the appropriate options based on your XML document and click **OK**.
The system saves your change and closes the PCDATA Properties dialog box.

Modifying the Output Side of Enveloping Build Maps

For each map you create, you need to modify both the input and output sides.

About this task

Use this procedure to modify each map.

Procedure

1. Right-click the XML File icon on the output side of the map and select **Properties**.
The system displays the XML File Properties dialog box.
2. In the name box, type the name of the Root Element in your XML document.
3. Select the **Tag** tab.
4. In the Tag box, type the tag of the Root Element in your XML document.
5. Select the **Output** tab.
6. For transaction build maps, do the following:
 - In the XML Prolog and Document Type Declaration section, select **No prolog**. 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 **No newlines**.
7. For interchange build maps, do the following:
 - 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.
8. Click **OK**.
The system saves your changes and closes the XML File Properties dialog box.
9. On the output side of the map, create the necessary header and trailer envelope structure.
See the *Application Integration User Guide* for information on specific tasks.
10. 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.

See the *Application Integration User Guide* for information on specific tasks.

11. Link the pcd data you created in Modifying the Input Side of Enveloping Build Maps to every map component on the output side of the map.

Note: The input pcd data must always contain data to produce the map components to which it is linked on the output side of the map.

12. Right-click the first envelope trailer element and select **Properties**.
The system displays the XML Element Properties dialog box (Name tab).
13. If the XML Element Properties dialog box does not contain a Loop Extended Rules tab, select the **Repeating** tab and verify that **Can repeat** is selected.
14. Select the **Loop Extended Rules** tab.
15. In the extended rule box, do the following:
 - Accept the default On Begin option.
 - Type the readblock/writeblock extended rule.

For 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);
```


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

16. Click **OK**.

The system saves your change and closes the XML Element Properties dialog box.

17. Right-click the XML File icon and select **Create Sub > Element**.

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

See About Creating Map Objects for more information on the Create Sub and Insert functions.

The system displays the XML Element Properties dialog box (Name tab).

18. In the name box, type **TEMP**.

19. Click **OK**.

The system saves the temporary element.

20. Right-click the element you created and select **Create Sub > Pcddata**.

21. The system displays the Pcddata Properties dialog box (Validation tab).

22. Click **OK**.

The system closes the Pcddata Properties dialog box.

Modifying the Map Details for Enveloping Build Maps

For each map you created, you also need to use the Translation Object Details dialog box to modify details of the maps.

About this task

Use this procedure to modify each map.

Procedure

1. Select **Edit > Details**.

The system displays the Translation Object Details dialog box.

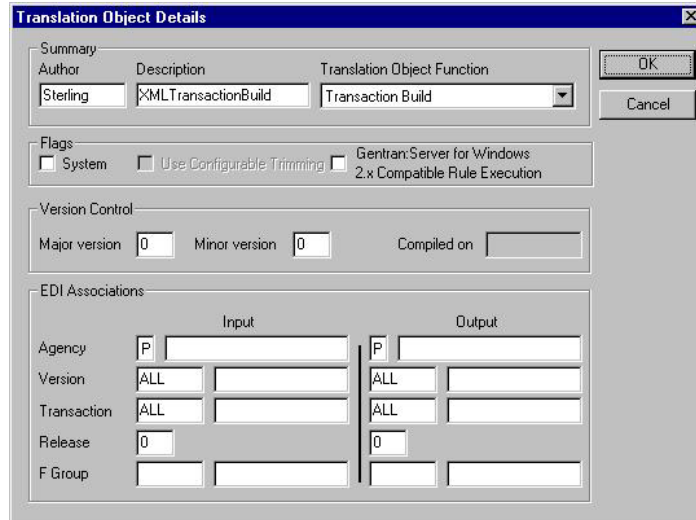
2. In the EDI Associations section, type a letter that is not currently used to designate a standards agency (such as P) in the first Agency box for both Input and Output.

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.

4. In the EDI Associations section, type **ALL** in the first Transaction box for both Input and Output.

This diagram illustrates how the EDI Associations section of the Transaction Object Detail dialog box should look.



5. Click OK.

The system saves your changes and closes the Translation Object Details dialog box.

About XML Break Maps

The following topics describe 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.

See *Creating an XML Map* for more information on creating maps. See *“Pass-Through Break Maps”* on page 45 for more information on the specific information necessary to create break maps.

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 Sterling Gentran:Server group table.

You will only use functional group build maps if your XML structure contains data that you want to be mapped to the Sterling 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
<p>Your export map requires the entire XML document structure to be passed through the map.</p> <p>For example:</p> <p>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.</p>	<p>Your export map does not require the entire XML document structure to be passed through the map.</p> <p>For example:</p> <p>You use non-pass-through break maps when you receive an XML document that does not require the entire XML structure.</p>

Inbound processing stages

This table describes the stages of inbound processing.

Stage	Description
1	<p>The Split Process</p> <p>The inbound data is processed through the Sterling Gentran:Server splitter, where all interchanges are validated and split into separate files.</p> <p>See <i>Modifying the Splitter Configuration</i> for more information on how to set up XML splitter specifications.</p>
2	<p>The Break Process</p> <p>The translator uses break maps to process the interchange data, locate the correct partner, locate the inbound relationship, and create document files.</p> <p>See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for more information on how to set up an inbound partner relationship.</p>
3	<p>The Export Process</p> <p>The translator uses the export map defined in the inbound relationship to translate data from one format to another.</p> <p>See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for more information on the following subjects:</p> <ul style="list-style-type: none"> • Setting up an inbound partner relationship, including automatic export options • The manual export process

Pass-Through Break Maps

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 Sterling Gentran:Server group table, you need to create a functional group break 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 break map	<ul style="list-style-type: none"> • Map type = Transaction Break • Input format = XML • Output format = XML
interchange break map	<ul style="list-style-type: none"> • Map type = Interchange Break • Input format = XML • Output format = XML

Use this table to determine where it is valid to use an Update standard rule or extended rule.

If the map is of type...	Then the update function is valid on the...
Export	Input side of the map only.
Import	Input or Output side of the map.
Break (Interchange, Group, or Transaction Set)	Input side of the map only.
Build (Interchange, Group, or Transaction Set)	Input side of the map only.

Modifying the Input Side of Pass-Through Break Maps

For each map you create, you need to modify both the input and output sides.

About this task

Use this procedure to modify each map.

Procedure

1. Right-click the XML File icon on the input side of the map and select **Properties**.
The system displays the XML File Properties dialog box.
2. In the name box, type the name of the Root Element in your XML document.
3. Select the **Tag** tab.
4. In the Tag box, type the tag of the Root Element in your XML document.
5. Click **OK**.
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.

See the *Application Integration User Guide* for information on specific tasks.

7. Add the necessary lookups and database updates.

See the *Application Integration User Guide* for information on specific tasks.

The following are the lookup and update keys that you may need in your map. Your XML map may require more or less, depending on your XML specifications.

- interchange break map - Partner Lookup, Agency Update, Version Update
- transaction break map - Transaction Set ID Update

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.

8. Right-click the appropriate element in the XML file structure and select **Properties**.

The system displays the XML Element Properties dialog box.

9. Select the **Loop Extended Rules** tab.

The system displays the extended rule options.

10. In the extended rule box, do the following:

- Accept the default **On Begin** option.
- Type the readblock/writeblock extended rule.
- Click **OK**.

For 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);
```

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

The system saves your change and closes the XML Element Properties dialog box.

Modifying the Output Side of Pass-Through Break Maps

For each map you create, you need to modify both the input and output sides.

About this task

Use this procedure to modify each map.

Procedure

1. Right-click the XML File icon on the output side of the map and select **Properties**.

The system displays the XML File Properties dialog box.

2. In the name box, type the name of the Root Element in your XML document.
3. Select the **Tag** tab.

The system displays the tag options.

4. In the Tag box, type the tag of the Root Element in your XML document.
5. On the **Output** tab, do the following:
 - In the XML Prolog and Document Type Declaration section, select **No prolog**.
 - In the Document Formatting section, select **No newlines**.

- Click **OK**.

The system saves your changes and closes the XML File Properties dialog box.

6. Right-click the XML File icon and select **Create Sub > Element**.

See About Creating Map Objects for more information on the Create Sub and Insert functions.

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

The system displays the XML Element Properties dialog box.

7. In the name box, type **TEMP**.

8. Click **OK**.

The system saves the temporary element.

9. Right-click the element you created and select **Create Sub > Pcd data**.

The system displays the Pcd data Properties dialog box.

10. Click **OK**.

The system closes the Pcd data Properties dialog box.

Modifying the Map Details for Pass-Through Break Maps

For each map you created, you also need to use the Translation Object Details dialog box to modify details of the maps.

About this task

Use this procedure to modify each map.

Procedure

1. Select **Edit > Details**.

The system displays the Translation Object Details dialog box.

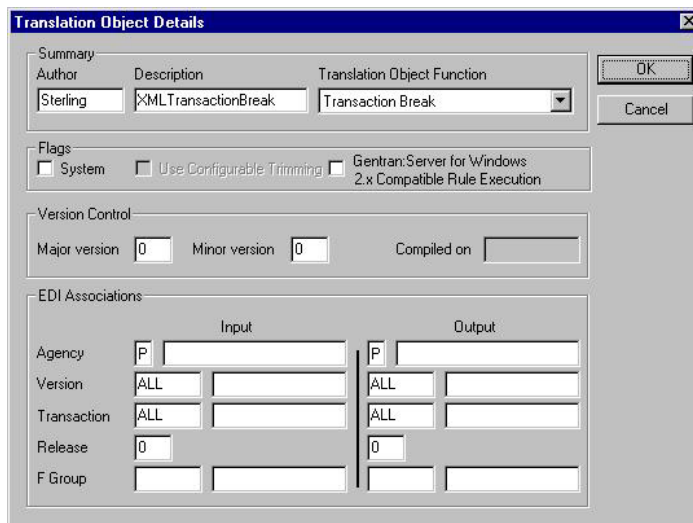
2. In the EDI Associations section, type a letter that is not currently used to designate a standards agency (such as P) in the first Agency box for both Input and Output.

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.

4. In the EDI Associations section, type **ALL** in the first Transaction box for both Input and Output.

This diagram illustrates how the EDI Associations section of the Transaction Object Detail dialog box should look.



5. Click OK.

The system saves your changes and closes the Translation Object Details dialog box.

Non-Pass-Through Break Maps

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.

You can only update the Sterling 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).

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	<ul style="list-style-type: none"> • Map type = Transaction Break • Input format = XML • Output format = XML
interchange break map	<ul style="list-style-type: none"> • Map type = Interchange Break • Input format = XML • Output format = XML

Modifying the Input Side of Non-Pass-Through Break Maps

For each map you create, you need to modify both the input and output sides.

About this task

Use this procedure to modify each map.

Procedure

1. Right-click the XML File icon on the input side of the map and select **Properties**.

The system displays the XML File Properties dialog box.

2. In the name box, type the name of the Root Element in your XML document.
3. Select the **Tag** tab.
4. In the Tag box, type the tag of the Root Element in your XML document.
5. Click **OK**.

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.

See the *Application Integration User Guide* for information on specific tasks.

Note: You may need to create temporary storage fields and use constants to hard-code specific values.

7. Add the necessary lookups and database updates.

See the *Application Integration User Guide* for information on specific tasks.

The following are the lookup and update keys that you may need in your map. Your XML map may require more or less, depending on your XML specifications.

- interchange break map - Partner Lookup, Agency Update, Version Update
- transaction break map - Transaction Set ID Update

8. Right-click the appropriate element in the XML file structure and select **Properties**.

The system displays the XML Element Properties dialog box.

9. Select the **Loop Extended Rules** tab.

10. In the extended rule box, do the following:

- Accept the default On Begin option.
- Type the readblock/writeblock extended rule.
- Click **OK**.

For 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
```

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

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.

The system saves your change and closes the XML Element Properties dialog box.

Modifying the Output Side of Non-Pass-Through Break Maps

For each map you create, you need to modify both the input and output sides.

About this task

Use this procedure to modify each map.

Procedure

1. Right-click the XML File icon on the output side of the map and select **Properties**.

The system displays the XML File Properties dialog box.

2. In the name box, type the name of the Root Element in your XML document.
3. Select the **Tag** tab.
4. In the Tag box, type the tag of the Root Element in your XML document.
5. On the **Output** tab, do the following:
 - In the XML Prolog and Document Type Declaration section, select **No prolog**.
 - In the Document Formatting section, select **No newlines**.
 - Click **OK**.

The system saves your changes and closes the XML File Properties dialog box.

6. Click **OK**.

The system saves your changes and closes the XML File Properties dialog box.

7. Right-click the XML File icon and select **Create Sub > Element**.

See About Creating Map Objects for more information on the Create Sub and Insert functions.

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

The system displays the XML Element Properties dialog box.

8. In the name box, type **TEMP**.
9. Click **OK**.

The system saves the temporary element.

10. Right-click the element you created and select **Create Sub > Pcd data**.

The system displays the Pcd data Properties dialog box (Validation tab).

11. Click **OK**.

The system closes the Pcd data Properties dialog box.

Modifying the Map Details for Non-Pass-Through Break Maps

For each map you created, you also need to use the Translation Object Details dialog box to modify details of the maps.

About this task

Use this procedure to modify each map.

Procedure

1. Select **Edit > Details**.

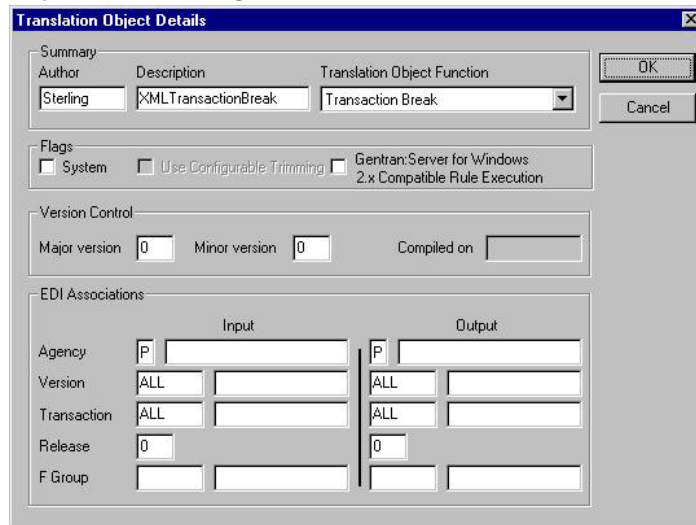
The system displays the Translation Object Details dialog box.

- In the EDI Associations section, type a letter that is not currently used to designate a standards agency (such as P) in the first Agency box for both Input and Output.

The information you enter must match your import specifications.

- In the EDI Associations section, type **ALL** in the first Version box for both Input and Output.
- In the EDI Associations section, type **ALL** in the first Transaction box for both Input and Output.

This diagram illustrates how the EDI Associations section of the Transaction Object Detail dialog box should look.



- Click **OK**.

The system saves your changes and closes the Translation Object Details dialog box.

Additional Tasks

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. See the <i>Application Integration User Guide</i> for information on compiling a translation object.
2	Register the build and break translation objects with the Sterling Gentran:Server system. See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for information on registering a translation object.
3	Create or modify the appropriate partner relationship in Sterling Gentran:Server. See XML Partner Relationships for information on XML-specific partner relationships.

XML Partner Relationships

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.

See the *IBM Sterling Gentran:Server for Microsoft Windows User 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. See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for information on creating an inbound partner relationship.
2	Modify the splitter configuration. See <i>Modifying the Splitter Configuration</i> 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. See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for information on creating an outbound partner relationship.
2	Select the XML build translation object at each appropriate level (transaction, group, interchange). See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for more information about the following subjects: <ul style="list-style-type: none">• Selecting a transaction build translation object• Selecting a group build translation object• Selecting an interchange build translation object
3	Set up the generic envelope segment.

Chapter 4. XML Tutorial

About the XML Tutorial

The following topics contain a tutorial for the Sterling Gentran:Server 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 used when you create your own maps.

To access reference information, see the *Application Integration User Guide*. For additional information on the various tasks described, refer to the *IBM Sterling Gentran:Server for Microsoft Windows User 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).

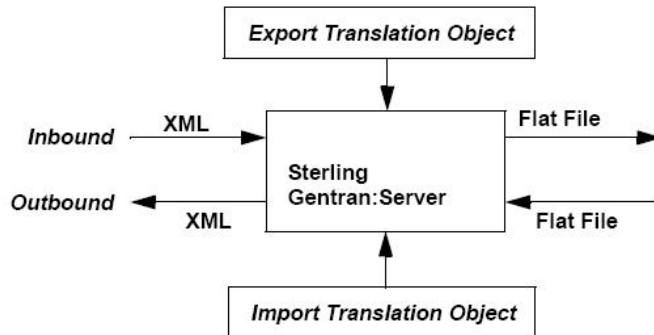
You need to create two maps—one that enables your company to translate the invoices that you receive from Pet Zone and one that allows you to translate your application file to the invoices that you send to Pet Zone.

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 Sterling Gentran:Server 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 Sterling Gentran:Server 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. For this tutorial we have done the analysis for you.

Task	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.

Map Building Process

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

Stage	Description
1	<p>Prepare and Analyze</p> <p>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.</p>
2	<p>Set Global Defaults (first time only)</p> <p>The first time you use the Sterling Gentran:Server Application Integration subsystem, you should establish the default date format, display, and confirmation options that the system uses.</p> <p>See the <i>Application Integration User Guide</i> for more information on setting these defaults.</p>
3	<p>Register the Tutorial XML Build and Break Translation Objects</p> <p>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 Sterling Gentran:Server.</p> <p>See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for more information on registering translation objects.</p>
4	<p>Set Up a Splitter Configuration Entry for XML</p> <p>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.</p> <p>See <i>Modifying the Splitter Configuration</i> for more information on setting up an XML Splitter entry.</p>
5	<p>Create, Save, and Name a New Map</p> <p>See <i>Creating the Inbound Tutorial Map</i> for more information on creating, saving, and naming a map.</p>
6	<p>Define your Flat File Format</p> <p>If you are creating an import or export map, you must define your flat file to the Application Integration subsystem. In Sterling Gentran:Server 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).</p> <p>See <i>Creating the Inbound Tutorial Map</i> for more information about defining your application file format by importing a DTD.</p>

Stage	Description
7	<p>Map the Appropriate Data for Each Application Field</p> <p>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.</p> <p>To map information to a field, you use linking, standard rules, extended rules, or a combination of all three.</p> <p>See <i>Correlating your Flat File and the XML Invoice and Additional Mapping Requirements</i> for more information about mapping your data.</p>
8	<p>Compile the Translation Object</p> <p>See <i>About Finalizing a Map</i> for more information on compiling the translation object and translation object naming conventions.</p>
9	<p>Register the Translation Object</p> <p>See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for more information on registering translation objects.</p>
10	<p>Create or Import the Appropriate Trading Relationship</p> <p>Establish the appropriate trading relationship in Sterling Gentran:Server 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 Sterling Gentran:Server, before you begin this tutorial.</p> <p>See the <i>IBM Sterling Gentran:Server for Microsoft Windows User Guide</i> for more information on creating and importing trading relationships.</p>
11	<p>Testing the Translation Object</p> <p>Obtain test data from your partners and process the data. Verify acknowledgement processing (if applicable). Verify communications with your network.</p> <p>See <i>Testing the Inbound Translation Object</i> for more information on test the translation objects.</p>

Data for the Tutorial

For these tutorials, we have provided building blocks (a DTD and a DDF) to enable you to build the tutorial translation objects more quickly. We have provided test data for the translation objects you create. We have provided you with the system import translation object you will use in the outbound process.

The following table describes how to locate the building blocks for the tutorial:

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

The following table describes how to locate the test data for the tutorial:

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

The following table describes how to locate the translation object for the tutorial:

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

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.

Creating the Inbound Tutorial Map

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 (for example, a DTD).

About this task

Use this procedure to create the inbound tutorial map.

Procedure

1. In the Application Integration subsystem, select **File > New**.
The system displays the New Map Wizard.
2. Enter the following information and click **Next**:
 - a. Select **Export** as the map type.
 - b. Type **Petest Inbound XML Invoice** for the map name. The system adds the .MAP extension.
 - c. Type your name if it differs from the user name prompted by the system.

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 Sterling 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. Select **XML (Extensible Markup Language)** and click **Customize**.
4. Do the following:
 - a. Select **DTD** as the document source type and click **Next**.
 - b. Navigate to the \GENSRVNT\Tutorial\XML directory, select **PET_XML.DTD**, click **Open**, and click **Next**.
 - c. Select **PETTEST_INVOICE** as the doctype.
 - 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 needs to make changes to the DTD to make it compliant with Sterling Gentran:Server, it informs you of the changes. Click **OK**.

f. Click **Next**.

5. For the output side of the map you want to load the data format from a saved definition.

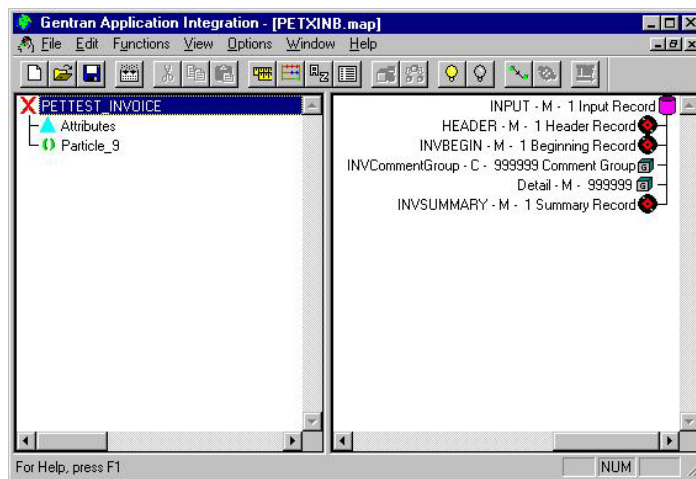
a. Select **Load the data format from a saved definition**.

b. Navigate to the \GENSRVNT\Tutorial\XML directory and select the **PET_INV.DDF** definition file.

c. Click **Next**.

6. Click **Finish** to create the new map from the information you selected (this may take a few seconds).

The system displays the new map in the Application Integration Window.



7. Select **File > Save** and accept the name **Petest Inbound XML Invoice.MAP**.

8. Select **Edit > Details**.

The system displays the Transaction Object Details dialog box.

9. 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.

10. To change the map version, type the appropriate version numbers in the Major and Minor boxes.

11. Click **OK**.

The system saves your changes and exits the Translation Object Details dialog box.

What to do next

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

Correlating your Flat File and the XML Invoice

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.

About this task

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 component from the input side to a component on the output side. The link between two map components is visually represented with a line connecting them.

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

Procedure

1. If the Linking function is not currently active, click **Link** on the Main Toolbar.
2. Link all of the components listed in the tables in "Invoice Map Components to Link."
 - a. Click the input map component.
The cursor changes to a link arrow.
 - b. Click the output map component to which you want to link the input component.

A line is displayed, visually illustrating the link between the input and output map components.

Invoice Map Components to Link

Correlate your flat file and XML invoice by linking the components listed in these tables.

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 pcddata. Link the map components in this table by using the Application Integration Link function.

Link this input pcddata...	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 Pcddata. Link the map components in this table by using the Application Integration Link function.

Link this input pcddata...	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	CUSTPROCEDURE
LINEITEM\UPCCCode	UPCCODE
LINEITEM\ItemDescription	ITEMDESC

Link this input pcddata...	To this INVCOMMENT2 record output field...
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 pcddata. Link the map components in this table by using the Application Integration Link function.

Link this input pcddata...	To this INVSUMMARY record output field...
HEADER\InvoiceNumber	INVNUMSUM
SUMMARY\TotalInvoiceAmount	TOTALAMOUNT
SUMMARY\NumberOfLineItems	TRANSTOTAL

Additional Mapping Requirements

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

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

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.

Record	Field	Mapping to be performed
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.

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

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.

Fields Using Constant	Constant ID	ConstantValue	Fields Qualified
<ul style="list-style-type: none"> • BILLTOACCTBEG • BILLTOACCTCM2 • BILLTOACCTDET • BILLTOACCTSUM 		INV5008	<ul style="list-style-type: none"> • INVNUMBEG • MESSAGETEXT • ITEMNUM • TOTALAMOUNT
<ul style="list-style-type: none"> • DOCTYPEBEG • DOCTYPECM2 • DOCTYPEDET • DOCTYPSUM 		Invoice	<ul style="list-style-type: none"> • INVNUMBEG • MESSAGETEXT • ITEMNUM • TOTALAMOUNT

Performing Additional Mapping 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 components that do not have the same maximum usage specified.

About this task

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

Procedure

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

Mapping the HEADER\STANDARD Field

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

About this task

Use this procedure to create the extended rule.

Procedure

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";`
4. Click **Compile** to validate the syntax of the extended rule.

Note: 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 you can verify the accuracy of the rule after you create it.

5. Click **OK** to add the extended rule to the STANDARD field.

Mapping the HEADER\TEST_PROD Field

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

About this task

Use this procedure to create the extended rule.

Procedure

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";`
4. Click **Compile** to validate the syntax of the extended rule.
5. Click **OK** to add the extended rule to the TEST_PROD field.

Mapping the INVBEGIN\BILLTOACCTBEG Field

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.

About this task

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.

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

Procedure

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 **INV5008** to identify this field as containing Bill To Account information.
5. From the Qualifies list, select **NVNUMBEG** 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.

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.

Mapping the INVBEGIN\DOCTYPEBEG Field

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.

About this task

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.

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

Procedure

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 **Invoice** to identify this field as containing Document Type information.
5. From the Qualifies list, select **INVNUMBEG** to set up a qualifying relationship with the Invoice Number field.

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.

Mapping the INVCOMMENT2\BILLTOACCTCM2 Field

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.

About this task

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.

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

Procedure

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 **INV5008** to identify this field as containing Bill To Account information.
5. From the Qualifies list, select **MESSAGETEXT** to set up a qualifying relationship with the Message Text field.
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.

Mapping the INVCOMMENT2\DOCTYPECM2 Field

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.

About this task

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.

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

Procedure

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 **Invoice** to identify this field as containing Document Type information.
5. From the Qualifies list, select **MESSAGETEXT** to set up a qualifying relationship with the Message Text field.

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.

Mapping the INVCOMMENT2\INVNUMCM2 Field

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.

About this task

Use this procedure to create the extended rule.

Procedure

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;`
4. Click **Compile** to validate the syntax of the extended rule.
5. Click **OK** to add the extended rule to the INVNUMCM2 field.

Mapping the INVDETAIL\BILLTOACCTDET Field

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.

About this task

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.

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

Procedure

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 **INV5008** to identify this field as containing Bill To Account information.
5. From the Qualifies list, select **ITEMNUM** to set up a qualifying relationship with the Item Number field.
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.

Mapping the INVDETAIL\DOCTYPEDET Field

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.

About this task

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.

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

Procedure

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 **Invoice** to identify this field as containing Document Type information.
5. From the Qualifies list, select **ITEMNUM** to set up a qualifying relationship with the Item Number field.
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.

Mapping the INVDETAIL\INVNUMDET Field

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.

About this task

Use this procedure to create the extended rule.

Procedure

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;`
4. Click **Compile** to validate the syntax of the extended rule.
5. Click **OK** to add the extended rule to the INVNUMDET field.

Mapping the INVSUMMARY\BILLTOACCTSUM Field

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.

About this task

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.

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

Procedure

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 **INV5008** to identify this field as containing Bill To Account information.
5. From the Qualifies list, select **TOTALAMOUNT** to set up a qualifying relationship with the Total Invoice Amount field.
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.

Mapping the INVSUMMARY\DOCTYPSUM Field

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.

About this task

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.

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

Procedure

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 **Invoice** to identify this field as containing Document Type information.
5. From the Qualifies list, select **TOTALAMOUNT** to set up a qualifying relationship with the Total Invoice Amount field.
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.

What to do next

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.

Outbound Flat File to XML Invoice

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.

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 Sterling 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.

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

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.

See the *Application Integration User Guide* for more information on how to create a system import map.

Creating the Outbound Tutorial Map

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 (such as a DTD).

About this task

Use this procedure to create the outbound tutorial map.

Procedure

1. In the Application Integration subsystem, select **File > New**.
The system displays the New Map Wizard.
2. Enter the following information and click **Next**:

- a. Select **Import** as the map type.
- b. Type **Petest Outbound XML Invoice** for the map name. The system adds the **.MAP** extension.
- c. Type your name if it differs from the user name prompted by the system.

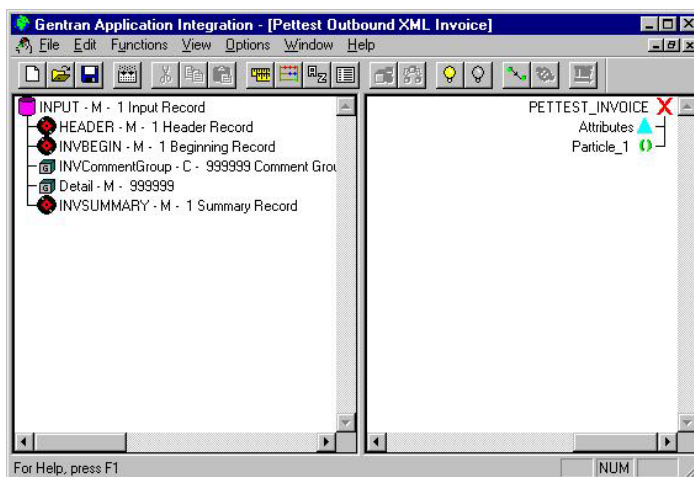
Note: You need to complete the format of the Input side of the map. This is the format of the data that is translated by the Sterling Gentran:Server system.

3. For the input side of the map you want to load your application data format from a saved definition. Do the following:
 - a. Select **Load the data format from a saved definition**.
 - b. Navigate to the **\GENSRVNT\Tutorial\XML** directory and select the **PET_INV.DDF** definition file.
 - c. Click **Next**.
4. For the output side of the map you want to create a new data format using a syntax that you define. Select **XML** and click **Customize**.
5. Do the following:
 - a. Select **DTD** as the document source type and click **Next**.
 - b. Browse to the **\GENSRVNT\Tutorial\XML** directory, select **PET_XML.DTD**, click **Open**, and then click **Next**.
 - c. Select **PETTEST_INVOICE** as the doctype.
 - d. Set the maximum length of data elements to **256** and click **Next**.
 - e. Click **Finish**.

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

- f. Click **Next**.
6. Click **Finish** to create the new map from the information you selected (this may take a few seconds).

The system displays the new map in the Application Integration Window.



7. Select **File > Save** and accept the name **Petest Outbound XML Invoice**. The system adds the **.MAP** extension and saves the map.
8. Select **File > Details**. The system displays the Transaction Object Details dialog box.
9. 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.
10. To change the map version, type the appropriate version numbers in the Major and Minor boxes.
 11. Click **OK**.
The system saves your changes and exits the Translation Object Details dialog box.

What to do next

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

Correlating your Outbound Flat File and the XML Invoice

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.

About this task

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.

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

Procedure

1. If the Linking function is not currently active, click **Link** on the Main Toolbar to turn on the Linking function.
2. Link all of the components listed in the tables in “Outbound Map Components to be Linked.”
 - a. Click the input map component.
The cursor changes to a link arrow.
 - b. Click the output map component to which you want to link the input component.
A line is displayed, visually illustrating the link between the input and output map components.

Outbound Map Components to be Linked

Correlate your flat file and XML invoice by linking the components listed in these tables.

Outbound attributes

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...
INVDATA	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
CUSTPROCEDURE	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

Performing Additional Mapping for the Output Side

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.

About this task

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 components that do not have the same maximum usage specified.

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

Procedure

1. On the output side of the inbound map, right-click the **Comments** element (located within the Header element) and select **Properties**.
The system displays the XML Element Properties dialog box.
2. Select the **Repeating** tab.

3. Select **Can repeat, with a maximum usage**.
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**.
The system displays the XML Element Properties dialog box.
7. Select the **Repeating** tab.
8. Select **Can repeat, with a maximum usage**.
9. In the Maximum Usage box, type 999999.
10. Click **OK** to change the maximum usage of the LineItem element.

About Finalizing a Map

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 Sterling Gentran:Server reports, and test the maps.

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

After you compile the map, print and verify the report, and register the translation object with Sterling Gentran:Server, 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.

Compiling a Map

About this task

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

Procedure

1. Select **File > Save** to save the source map prior to using the Compile function.
2. Select **File > Compile** to display the Run-Time Translation Object Name dialog box.
3. Navigate to the location of the compiled translation objects, if necessary.

Important: 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 Sterling Gentran:Server.

4. To compile the inbound map, type **Petest Inbound XML Invoice.tpl** and click **Save**.
5. To compile the outbound map, type **Petest Outbound XML Invoice.tpl** and click **Save**.

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.

Important: Do not overlay the source map with the compiled translation object. Use the .TPL file extension to distinguish the translation object.

The system compiles the map and generates a translation object. The Compile Error dialog box is displayed.

6. 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.

7. Select **File > Save** to save the source map with the Compiled on date.

Note: You must register this translation object with the Sterling Gentran:Server system before you can use it.

See the *IBM Sterling Gentran:Server for Microsoft Windows User Guide* for more information on registering a translation object.

Printing the Report

The Sterling Gentran:Server 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).

About this task

Use this procedure to print the Sterling Gentran:Server report.

Procedure

1. Select **File > Print**.
2. Accept the defaults on this dialog box (all options checked) so the report will include all sections and only the activated map components. Click **OK**.
3. Set the appropriate options. Click **OK** if you do not need to change Setup options, and the mapping report is printed.
4. Click **Setup** if you need to access printer setup information (to select a specific printer, paper orientation, or size and source of paper).
5. Set the appropriate options in the Print Setup dialog box. 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.
6. Click **Options** if you need to access printer setup options information. The Options dialog box is displayed.
7. Set the appropriate options. Click **OK** to return to the Print Setup dialog box.
8. Click **OK** on the Print Setup dialog box to return to the Print dialog.
9. Click **OK** on the Print dialog box and the mapping report is printed.

Testing the Inbound Translation Object

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

About this task

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

Procedure

1. Register the following translation objects with Sterling Gentran:Server:

- Pettest Inbound XML Invoice.TPL
- XML_INT_BRK.TPL

The default 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>

See the *Administration Guide* for more information about how to define a new splitter entry.

3. Click **Apply** and then **OK** to exit the System Configuration program.
4. Import the PETZONE5.PAR partner relationship into Sterling Gentran:Server.

The default path for the partner file is:

C:\GENSRVNT\TUTORIAL\XML\PETZONE5.PAR

5. Verify (in Partner Editor) that the Pettest Outbound XML Invoice Export translation object is selected for the Inbound Relationship.
6. Use the Process File option in Sterling Gentran:Server 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 v is installed.

The default full path for the data file is:

C:\GENSRVNT\TUTORIAL\XML\Inbound_XML_Invoice.INT

After the document is translated, it is located in the In Documents in Sterling Gentran:Server.

7. Export the document to ensure that it was translated correctly.

Testing the Outbound Translation Object

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

About this task

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

Procedure

1. Register the following translation objects with Sterling Gentran:Server:

- Pettest Outbound XML Invoice.TPL
- XML_INT_BLD.TPL
- XML_TRN_BLD.TPL

The default 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 Sterling Gentran:Server.

The default path for the header translation object is:

C:\GENSRVNT\TUTORIAL\HEADER.TPL

3. If you have not already done so, import the PETZONE5.PAR partner relationship into Sterling Gentran:Server.

The default 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

Click **Apply** and then **OK** to exit the System Configuration program.

See the *IBM Sterling Gentran:Server for Microsoft Windows Administration Guide* for more information about how your system administrator must modify the system configuration program.

6. Use the Import option in Sterling Gentran:Server to process the data file (Outbound_XML_Invoice.TXT) through the translation object.

The default path for the data file is:

c:\GENSRVNT\TUTORIAL\XML\Outbound_XML_Invoice.TXT

After the document is translated, it is located in the Workspace in Sterling Gentran:Server.

7. View the EDI data to ensure that the document was translated correctly.
8. Post the document to the Out Documents browser.
9. In the Out Documents browser, highlight the Tutorial document and click **View**.
10. When you are done, click **Send**.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2012. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2012.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise®, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce®, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Index

A

- attribute properties
 - extended rules 33
 - name 30
 - standard rules 33
 - tags 31
 - types 31
 - validation 32
- attributes
 - creating 33

B

- BILLTOACCTBEG field
 - tutorial 66
- BILLTOACCTCM2 field
 - tutorial 67
- BILLTOACCTDET field
 - tutorial 68
- BILLTOACCTSUM field
 - tutorial 69
- break maps 35
 - inbound process 44
 - modifying non-pass-through maps 49, 51
 - modifying pass-through maps 46, 47, 48
 - non-pass-through 49
 - overview 44
 - pass-through 45
- build maps 35
 - enveloping 40
 - modifying enveloping maps 41, 43
 - modifying pass-through maps 38, 39
 - outbound process 36
 - overview 36
 - pass-through 37
- building maps
 - overview 57

C

- child objects
 - type 26
- compiling maps
 - tutorial 76
- constants
 - tutorial 63
- content particle properties
 - extended rules 26
 - looping 26
 - name 25
 - repeating 26
 - type 26
- content particles
 - creating 27
- Create Sub function 15

D

- DDFs
 - exporting 7
 - importing 6
 - restrictions 5
- decimal point 19
- DOCTYPEBEG field
 - tutorial 66
- DOCTYPECM2 field
 - tutorial 67
- DOCTYPEDET field
 - tutorial 69
- DOCTYPESUM field
 - tutorial 70
- DTD overview 5

E

- element properties
 - extended rules 24
 - key field 22
 - looping 24
 - name 21
 - repeating 23
 - tags 22
- elements
 - creating 24
- encoding options
 - inbound 3
 - outbound 4
- entities 17
 - creating 21
- entity properties
 - entity value 21
 - name 20
 - tags 20
- envelopes
 - extracting data with splitter entries 10
- enveloping build maps 36, 40
 - modifying input side 41
 - modifying map details 43
 - modifying output side 41
- export map 55
- extended rules
 - attributes 33
 - content particles 26
 - overview 63
 - pcdata 29
 - XML elements 24
 - XML files 18

F

- file objects 15
- file properties
 - conditional or mandatory objects 17
 - decimal point 19
 - DTD 17
 - encoding 17

- file properties (*continued*)
 - entities 17
 - extended rules 18
 - file formatting 17
 - modifying 19
 - name 16
 - prolog 17
 - XML tags 16
- flat file to XML invoice
 - tutorial 71
- flat files
 - correlating to invoice 61, 73
- functional changes for version 5.2 9, 10

I

- import map 55
- inbound maps
 - testing 78
 - tutorial 59
- inbound process
 - break maps 44
- inbound processing
 - tutorial 59
- inbound relationships
 - overview 53
- input mapping
 - tutorial - additional steps 64
- Insert function 15
- INVNUMCM2 field
 - tutorial 68
- INVNUMDET field
 - tutorial 69
- invoice map components
 - linking 61

M

- map analysis 55
- map components
 - linking invoice 61
 - linking outbound 74
- maps
 - additional tasks 52
 - break maps 35
 - break maps overview 44
 - build maps 35
 - build maps overview 36
 - building process 57
 - compiling 76
 - creating an XML map 12
 - DDF files 5
 - enveloping build maps 40
 - export 55
 - finalizing 76
 - icons 2
 - import 55
 - map objects 15
 - map types 11
 - modifying enveloping maps 41, 43

- maps (*continued*)
 - modifying for version 5.2 changes 10
 - modifying non-pass-through maps 49, 51
 - modifying pass-through maps 38, 39, 46, 47, 48
 - New Map Wizard 11
 - non-pass-through break maps 49
 - pass-through break maps 45
 - pass-through build maps 37
 - reports 77
 - testing 78
 - transaction break maps 10
 - tutorial 76
 - using DTDs 11
 - XML file objects 15

N

- New Map Wizard
 - features 11
- non-pass-through break maps 44, 49
 - modifying input side 49
 - modifying map details 51
 - modifying output side 51

O

- outbound map components
 - linking 74
- outbound maps
 - testing 78
 - tutorial 71
- outbound process
 - build maps 36
- outbound processing
 - tutorial 71
- outbound relationships
 - overview 53
- output mapping
 - tutorial - additional steps 63, 75

P

- partner relationships
 - overview 53
- pass-through break maps 44, 45
 - modifying input side 46
 - modifying map details 48
 - modifying output side 47
- pass-through build maps 36, 37
 - modifying input side 38
 - modifying map details 39
 - modifying output side 38
- pcdata
 - creating 29
- pcdata properties
 - extended rules 29
 - standard rules 29
 - validation 28
- process overview 9
- properties
 - attribute extended rules 33
 - attribute name 30
 - attribute standard rules 33
 - attribute tag 31

- properties (*continued*)
 - attribute types 31
 - attribute validation 32
 - conditional or mandatory objects 17
 - content particle looping 26
 - content particle name 25
 - content particle repeating 26
 - content particle type 26
 - decimal point 19
 - DTD 17
 - element key field 22
 - element looping 24
 - element name 21
 - element repeating 23
 - element tags 22
 - encoding 17
 - entities 17
 - entity name 20
 - entity tag 20
 - entity value 21
 - extended rules 18, 24, 26
 - formatting 17
 - modifying XML files 19
 - pcdata extended rules 29
 - pcdata standard rules 29
 - pcdata validation 28
 - prolog 17
 - XML file name 16
 - XML tags 16

R

- relationships
 - overview 53
- reports
 - tutorial 77

S

- splitter entries 10
- STANDARD field
 - tutorial 65
- standard rules
 - attribute 33
 - overview 63
 - pcdata 29

T

- tags 16
 - attributes 31
 - elements 22
 - entities 20
- TEST_PROD field
 - tutorial 65
- testing maps
 - inbound 78
 - outbound 78
- translation process
 - inbound 55
 - outbound 55
- tutorial data 58
- types
 - attributes 31

V

- validation
 - attributes 32

W

- wildcards 35

X

- XML functionality
 - changes for version 5.2 9, 10
- XML invoice
 - correlating to flat file 61, 73
- XML invoice to flat file
 - tutorial 59
- XML rule exceptions 1
- XML support specifications 1



Product Number: 5725-D09

Printed in USA