InterSystems® Ensemble

Ensemble X12 Development Guide

Version 2017.2
2020-06-26

Ensemble X12 Development Guide
Ensemble   Version 2017.2   2020-06-26
Copyright © 2020 InterSystems Corporation
All rights reserved.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

About This Book

This book describes how to add X12 routing interfaces to an Ensemble production. It contains the following sections:

- Introduction
- X12 Schemas and Available Tools
- Configuring the Production
- Handling X12 Interchanges
- Reference for Settings

For a detailed outline, see the table of contents.

The following books provide related information:

- *Ensemble Best Practices* describes best practices for organizing and developing Ensemble productions.
- *Developing Ensemble Productions* explains how to perform the development tasks related to creating an Ensemble production.
- *Configuring Ensemble Productions* describes how to configure Ensemble productions, business hosts, and settings. It also provides reference information on settings not discussed in this book.
- *Ensemble Virtual Documents* describes the concept of Ensemble virtual documents and provides generic information on working with them.

For general information, see the *InterSystems Documentation Guide*.

1

Introduction

This chapter briefly introduces the X12 standard and Ensemble support for X12. It contains the following sections:

- Introduction to X12
- X12 and HIPAA.
- Standard Exchange Format (SEF) Files
- Ensemble Support for X12 Documents

1.1 Introduction to X12

The American National Standards Institute (ANSI) founded the Accredited Standards Committee (ASC) X12 as a cross-industry forum to build and support electronic data exchange standards, related documents and products intended for worldwide use. Thus, X12 is the ANSI standard for Electronic Data Interchange (EDI). There are more than 300 document types defined. No X12 document type is excluded from Ensemble support, but most of the focus at InterSystems has been on the documents related to the Health Insurance Portability and Accountability Act (HIPAA).

1.2 X12 and HIPAA

Title II of HIPAA, the Administrative Simplification (AS) provisions, requires the establishment of national standards for electronic healthcare transactions and national identifiers for providers, health insurance plans, and employers. The AS provisions also address the security and privacy of health data. The standards are meant to improve the efficiency and effectiveness of the healthcare system in the United States by encouraging the widespread use of electronic data interchange in the US healthcare system.

Among the X12 documents that specifically support HIPAA are:

- 270_X092: Eligibility, Coverage or Benefit Inquiry
- 271_X092: Eligibility, Coverage or Benefit Information
- 276_X093: Healthcare Claim Status Request
- 277_X093: Healthcare Claim Status Notification
- 278_X094Request: Healthcare Service Review (Request)

- 278_X094Response: Healthcare Service Review (Response)

- 820_X061: Payment Order/Remittance Advice

- 834_X095: Benefit Enrollment and Maintenance

- 835_X091: Healthcare Claim Payment/Advice

- 837_X096: Healthcare Claim: Institutional

- 837_X097: Healthcare Claim: Dental

- 837_X098: Healthcare Claim: Professional

Ensemble comes prepackaged with HIPAA 4010 and HIPAA 5010 Schemas installed. HIPAA 5010 provides additional documents for health insurance exchange including 277CA, 820hix, 837apD, 837apI, 837apP, and 999.

1.3 Standard Exchange Format (SEF) Files

The X12 Standards describe in detail the syntax and semantics of documents. These descriptions are available as PDF files from the official publisher, Washington Publishing Company:

http://www.wpc-edi.com

While informative, the documents are not intended to be machine readable. That is where SEF files come in. The SEF file encodes the syntax and much of the semantics of an EDI document. For example, not only does it specify the size and types of fields and any code values which may be applicable, but it also defines dependencies between fields and dependencies between segments within documents. The SEF standard was developed by Foresight Corporation to encode any EDI specification (not just X12) and put into the public domain. You can download a printable version of the standard from:

http://www.edidev.com/articles/sef16.pdf

This information is worth examining, as it will give you an idea of the scope and complexity of EDI. Many organizations have encoded various EDI standards and made them available to the public. In addition, the United States Department of Defense has published many SEF files for their acquisition systems.

1.4 Ensemble Support for X12 Documents

Ensemble stores X12 documents as type of Caché object called a virtual document to enable faster processing. Specifically, an X12 document is stored as an EnsLib.EDI.X12.Document where each segment of the X12 document is stored as an EnsLib.EDI.X12.Segment. X12 Interchanges are also stored as an EnsLib.EDI.X12.Document with pointers to the other X12 documents contained within. Segments and Elements of a Transaction Set are reached by using a virtual property path. For example, the identifier code of a given Transaction Set would be accessed by virtual property path "ST:TransactionSe-tIdentifierCode". See Parsing the Document for more information.

Ensemble provide tools so that you can access values in virtual documents for use in data transformations, business rules, and searching and filtering messages. For background information, see *Ensemble Virtual Documents*.

2

X12 Schemas and Available Tools

This chapter provides an overview of the Ensemble tools that you can use to work with X12 schemas and documents. It contains the following sections:

2.1 Using the X12 Schema Structures Page

The **X12 Schemas** page enables you to import and view X12 schema specifications.

To display this page, click **Ensemble** > **Interoperate** > **ASC X12** > **ASC X12 Schema Structures**.

For general information on using this page, see "Using the Schema Structures Page" in *Ensemble Virtual Documents*.

To determine the schema structures, Ensemble extracts details from the .SETS, .SEGS, .COMS, .ELMS, and .CODES sections of the SEF file that was imported to define the structure of this X12 document. The description of a document is extracted from the .INI section of the SEF file that defines this X12 schema; an example is "Healthcare Claim Status Request."

2.1.1 Loading X12 Schemas into Ensemble

To load an X12 schema into Ensemble from the Schema Structures page, click **Import** and select your file.

For suggestions on where to find SEF files (X12 schemas) to import, see "Standard Exchange Format (SEF) Files," in the first chapter.

2.1.1.1 Loading SEF Files Programmatically

To load SEF files programmatically:

1. Start a Terminal session.

2. Change to an Ensemble namespace and issue the following command:

```
Do ##class(EnsLib.EDI.SEF.Compiler).Import(filename)
```

Where *filename* is the full pathname of the SEF file.

This command imports the data from the SEF file and make it available as a schema definition within Ensemble.

3. Ensemble creates a name for the new schema category from the first piece of the first line in the .INI section of the SEF file. For example, in 837_X097.SEF you might see this line:

```
.INI 837_X098,,004 010 X098,X,X12-4010,Healthcare Claim: Professional
```

The extracted schema category would have this name:

```
837_X098
```

Due to the schema naming convention, if you want to edit a SEF file to customize it, InterSystems suggests you first change the text in the SEF file that provides its category name, so that you can distinguish your version from any other SEF file that you also import into Ensemble.

4. A SEF file may contain syntax errors. If so, Ensemble issues an error message and identifies the location of the error in the SEF file.

2.1.2 Viewing a Document Structure

To view the internal organization of a document structure, click its name from the **DocType Structures** tab on the **X12 Schemas** page. Ensemble displays the segment structure of the document using the system of visual cues explained below. This is the **X12 Document Structure** page. The following example shows the HIPAA_4010:820 document structure.

Schema Category : **HIPAA_4010**
Document Structure (TransactionSet) : **820**
Implementation Guide : **X061A1**

Payment Order/Remittance Advice

ST	
BPR	
TRN	
CUR	
REF	
DTM	

Key

group
optional
repeating

▸ Show Raw Definition Text
▸ Go to Validation Structure

loop1000A
N1
N2
N3
N4

loop1000B
N1
N2
N3
N4
PER

loop2000A ENT!2=2L
loop2300A RMR
loop2310A IT1
loop2315A SLN
loop2320A ADX

loop2000B ENT
loop2100B NM1
loop2300B RMR
DTM
loop2320B ADX

SE
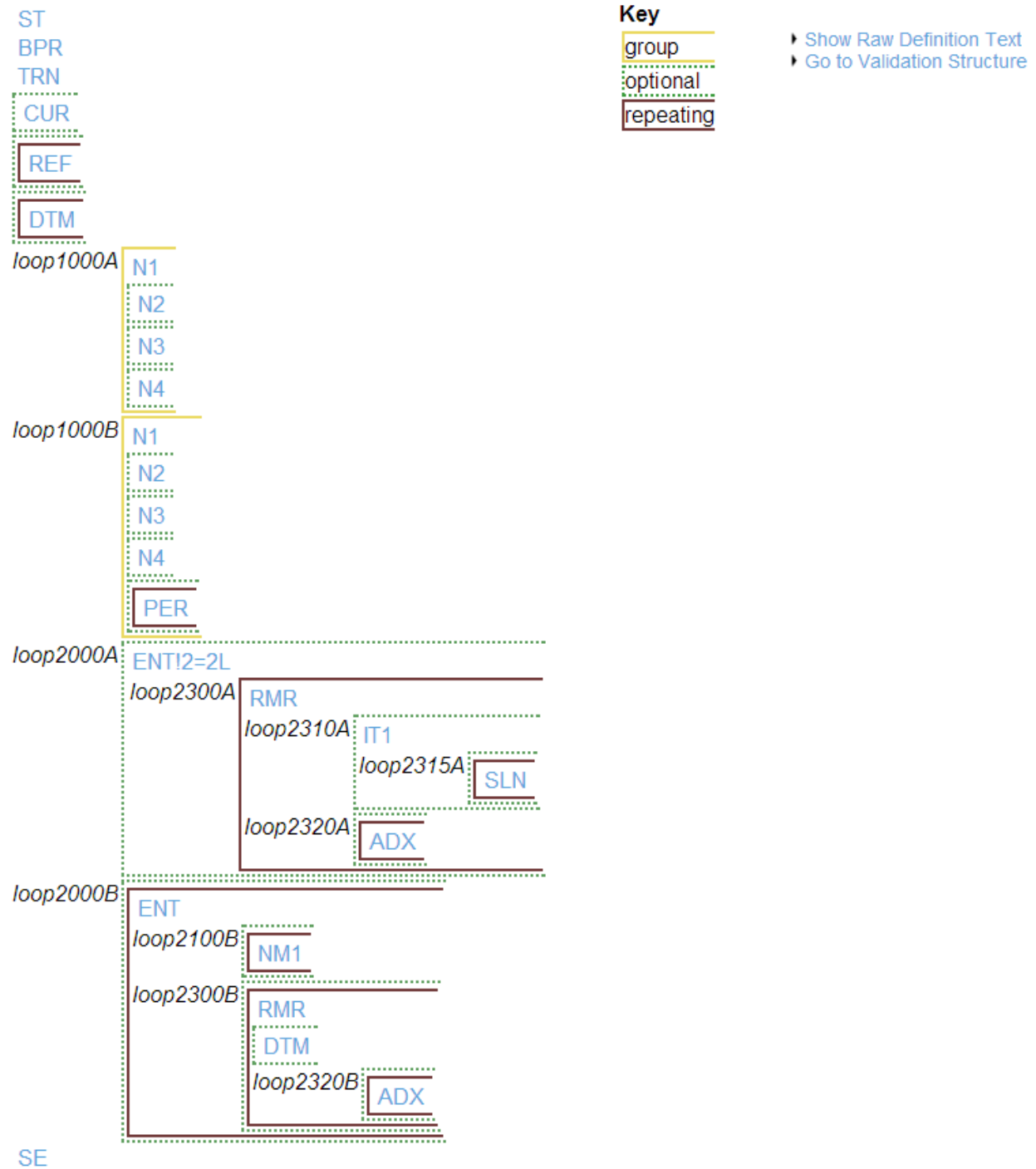
2.1.2.1 Layout

- Segments are listed in blue text and link to the relevant segment structure page.

- Loops are listed in black text.

- The segments that comprise the document structure are listed in sequential order, from left to right and top to bottom.

- The name of each document segment is displayed: BPR, NM1, DTM, etc. This name indicates the type of segment that exists at this location in the X12 document structure.

- Green dotted lines enclose segments that are *optional*.

- Brown solid lines enclose segments that, if present, may *repeat* several times.

- Yellow solid lines enclose segments that are part of a *group*.

- A segment may be both repeating and optional (see segment NM1 above).

- When you are viewing a segment diagram, if you hover the cursor over a three-letter segment name, a tooltip displays the syntax for referring to this segment in a virtual property path.

2.1.2.2 Raw Definition

To see the document structure in a raw text format, click **Show Raw Definition Text**. The raw definition of the HIPAA_4010:820 document structure is as follows:

Raw Definition

```
ST~BPR~TRN~[~CUR~]~[~{~REF~}~]~[~{~DTM~}~]~(1000A~N1~[~N2~]~[~N
[~N3~]~[~N4~]~[~{~PER~}~]~)~[2000A~ENT!2=2L~{2300A~RMR~[2310A~IT
{~ADX~}~]~}~]~[2000B~{~ENT~[2100B~{~NM1~}~]~]~[2300B~{~RMR~[~DTM
```

Note: Loops in X12 can have custom names. Note in the above raw definition sample for HIPAA_4010:820 that loop titles directly follow open bracket characters "[, { , (" unlike segment names, which are separated from bracket characters by a tilde "~". To use a custom loop name, type in your custom name in the place of the default loop name within the schema file prior to importing the schema into Ensemble.

2.1.2.3 Legacy Document Structure

You can view the old document structure viewer for schemas by clicking **Go to Validation Structure**. The legacy document structure also provides some information that is not available in the default document structure viewer. For example, the legacy document structure can provide code tables, the number of loop repetitions allowed, and syntax notes.

2.1.2.4 Implementation Guide

This code identifies the relevant ASC X12 **Implementation Guide** which can be found at http://www.wpc-edi.com/. The identifier is unique across all document structures and schemas.

2.1.3 Viewing a Segment Structure

To view the structure of a document segment, click on its name in any page similar to the example shown in the previous section. Ensemble displays a table that lists all the fields in that segment. This is the **X12 Schema Segment Structure** page.

For example, if you click on the PER segment in the HIPAA_4010:276 document structure, Ensemble displays the following page.

Schema Category: HIPAA_4010
Document Structure (Transaction Set): 276
Segment Structure: **PER**

Path you followed to get to this segment structure: **loop2000A().loop2100A().PER**

HIPAA_4010:PER Administrative Communications Contact

Elem	Description	Property Name	Data Type	Required	Length	Max Repeats	Alternate Description
1	Contact Function Code	ContactFunctionCode	(ID)	O	2	1	
2	Name	Name	(AN)	O	1-60	1	
3	Communication Number Qualifier	CommunicationNumberQualifier	(ID)	O	2	1	
4	Communication Number	CommunicationNumber	(AN)	O	1-80	1	
5	Communication Number Qualifier	CommunicationNumberQualifier2	(ID)	O	2	1	
6	Communication Number	CommunicationNumber2	(AN)	O	1-80	1	
7	Communication Number Qualifier	CommunicationNumberQualifier3	(ID)	O	2	1	
8	Communication Number	CommunicationNumber3	(AN)	O	1-80	1	
9	Contact Inquiry Reference	ContactInquiryReference	(AN)	O	1-20	1	

The columns are as follows:

- **Elem** — the number to use to access the element within the segment (if you prefer numbers).

- **Description** — a short description of the element.

- **Property Name** — the name no use to access the element within the segment (if you prefer names).

- **Data Type** — a one to two letter symbol representing the element data type. See the table below for details.

SYMBOL	TYPE
Nn	Numeric
R	Decimal
ID	Identifier
AN	String
DT	Date
TM	Time
B	Binary

- **Required** — displays R for required, O for optional.

- **Length** — displays the number of characters that can be in the element. If only one number is present, it represents the maximum number of characters. If two numbers are separated by a hyphen, it is the range of characters that can be in the associated element (minimum-maximum).

- **Max Repeats** — the maximum number of times the element can repeat.

- **Alternate Description** — a second, longer description of the element.

You can use this information, particularly the **Property Name** column, to build virtual property paths for Ensemble in the format *segment:elem*. The following are examples of virtual property paths involving simple *elem* values from the **PER**

segment in the HIPAA_4010:276 document structure. The () shortcut syntax indicates all available instances of a repeating field, whereas (1) indicates the first instance:

```
loop2000A().loop2100A().PER:ContactFunctionCode
loop2000A().loop2100A(1).PER:ContactFunctionCode
loop2000A().loop2100A(2).PER:ContactFunctionCode
loop2000A().loop2100A(x).PER:ContactFunctionCode
loop2000A().loop2100A().PER:Name
```

2.1.4 Viewing a Composite Structure

When you click on a name in the **Composite Structure** column, Ensemble displays all the elements in that data structure. This is the **X12 Composite Structure** page. The column values are identical to those of the previous section.

2.1.5 Choosing a Different Category

It is a feature of the X12 standard that a document structure can differ by X12 version, even when the structure has the same name and number. For example, both X12 HIPAA_4010 and X12 HIPAA_5010 define a document structure called 277, but these definitions contain different segments. Ensemble provides the document structure definitions HIPAA_4010:277 and HIPAA_5010:277. The **X12 Document Structure** page makes it easy to see the differences between the two definitions, as the following two figures show.

2.2 Using the X12 Document Viewer Page

Ensemble provides a Document Viewer page for X12. You can use this page to display, transform, and export X12 documents (either external files or documents from the Ensemble message archives).

To access this page:

1. Click **Ensemble**.

2. Click **Interoperate**.

3. Click **ASC X12**

4. Click the **ASC X12 Document Viewer** and then click **Go**.

2.2.1 Selecting Options

To specify the document to display:

1. For **Document Source**, select **File**, **Message Header ID**, or **Message Body ID**.

2. Specify the document to display:

   • If you selected **File**, use **Browse** to choose a file. For **Document Number in File**, type the number of the document to display.

   • If you selected **Message Header ID** or **Message Body ID**, type the ID of the message header or message body to display.

3. Specify how to parse the document. To do so, select one of the following options for **Document Structure or Schema**:

   • **As received by a Business Service** — Use the schema as assigned by a business service. If you select this, select a business service from the drop-down list.

     This option enables you to determine the DocType to which a particular business service would assign this document.

   • **Use a specific Schema Category/Version** — Choose a document category from the drop-down list.

   • **Use a specific DocType** — Enter the name of an document structure (<MessageStructure>) in the format *category*:*structure*. The parser uses this document structure.

   • **Use content-declared Version:Name** — Use the document structure associated with the document type declared in the document.

   • **Use object's stored DocType** — Use the DocType as declared in the document body object. (This option does not apply to stored documents loaded from a file.)

   • **None** — Do not use any DocType to parse the document. Instead, display the raw segments without transforming any of them into links.

   This option enables you to try interpreting documents from a particular data source as different schema category types to determine which DocType is the right one to use when handling documents from that source. There are a variety of reasons why you might need to do this. For example, you might find when you update an external application that it has changed the actual version of the documents it sends, but has neglected to update the type declaration that it sends in these documents. It is also useful in determining which of the built-in categories to use as a schema base, when a document uses a custom document structure.

4. Optionally click **Transform Document?** and specify the transformation details. See Testing a Transformation.

5. Click **OK**.

2.2.2 Parsing the Document

To parse the document, set the options described above and click **OK**. The Document Viewer displays the following on the right side of the screen:

- A summary of the document which contains following basic information:

    – The Data Transformation applied, if applicable

    – The Document ID

    – The DocType

    – The DocType description, if available

    – The number of segments

    – The Document ID of parent documents, if applicable

- The document data, which has one row for each segment in the document structure. Each row contains:

    – Segment number

    – Segment name, such as ISA or DN1

    – Element contents and separators, as contained in the document

If the document matches the schema you have selected, segments and elements will appear as links to the relevant structure page.

Note: Due to the multiplicity of X12 DocType structures that use the same transaction set identifier code (ST:1), Ensemble uses the reference identification number (REF:2) — see Implementation Guide — to uniquely identify document types. For example, examine HIPAA_4010:837P and HIPAA_4010:837D. Both documents have a transaction set identifier of 837, however each has a distinct reference identification number.

2.2.2.1 Displaying the Segment Address

To display the segment address, hover the cursor over a segment name in the shaded column. The tooltip displays the following:

- Segment address to use in a virtual property path

- Descriptive name of this segment
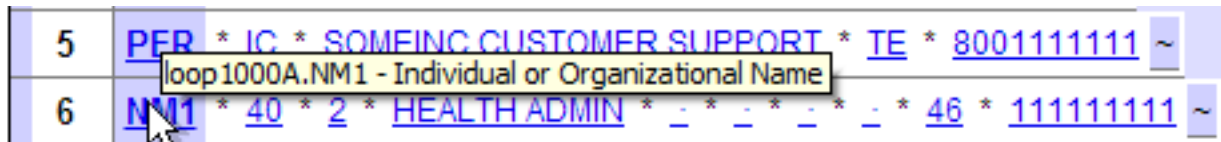
2.2.2.2 Displaying the Element Address

To display the Element address, hover the cursor over a field within the document structure. The tooltip displays the following:

- The *element* address to use in a virtual property path (as a number)

- The *element* address to use in a virtual property path (as a name)

2.2.2.3 Viewing X12 Interchanges

When viewing Transaction Sets nested in Groups and Interchanges the **X12 Document** allows you to walk through the document structure one level at a time.

The following display is the result of using the X12 document viewer to view a 4010:Interchange document.

The Document Viewer assigns the group document the identifier <38>.

When you click on a group document link in an X12 Interchange document display, a new browser window opens to display the group document. The document Viewer window, with the top-level parent, remains open in the original browser window.

The next display is the result of clicking the link to Group document <38>.

Note: If there are more than 10 Groups in an Interchange or more than 10 Transaction Sets in a Group, the Document Viewer displays links to the first five and last five documents. Between the lists is a text field, into which you can enter any identifier number between the first and last numbers. After you enter a number, click **Other**. A new browser window opens to display the document.

X12 **Group** Document - Id = 38,DocType = '**HIPAA_4010:Group**'
'*Group batch document containing TransactionSet documents of a given type*', **3** Segments, **10** child documents, parent document : <37>

1	**GS**	* HC * 133052274 * 121160001 * 20070112 * 2254 * 000000519 * X * 004010X097A1 ~
2	**TransactionSetDocsRef**	* 2 * 10	**TransactionSet documents :**	<39> <40> <41> <42> <43> <44> <45> <46> <47> <48> ~
3	**GE**	* 10 * 000000519 ~

X12 TransactionSet Document Id = 39
Type Name = '837'

The next display is the result of clicking the Transaction Set <39>. You can return to either the Group or the Interchange by clicking their respective Document ID number links.

X12 837 Document - Id = 39,DocType = 'HIPAA_4010:837D'
'Health Care Claim: Dental', 32 Segments, parent documents : <38>: <37>

1	ST	* 837 * 000000001 ~
2	BHT	* 0019 * 00 * 000000001 * 20070112 * 2254 * CH ~
3	REF	* 87 * 004010X097A1 ~
4	NM1	* 41 * 2 * SOMEINC * . * . * . * . * 46 * 111111111 ~
5	PER	* IC * SOMEINC CUSTOMER SUPPORT * TE * 8001111111 ~
6	NM1	* 40 * 2 * HEALTH ADMIN * . * . * . * . * 46 * 111111111 ~
7	HL	* 1 * . * 20 * 1 ~
8	PRV	* BI * ZZ * 204E00000X ~
9	NM1	* 85 * 1 * Xxxxx * Xxxxx * . * . * . * XX * 1111111111 ~
10	N3	* 1111 SOME ST ~
11	N4	* SOME CITY * CA * 11111 ~
12	REF	* EI * 111111111 ~
13	REF	* G5 * 0001 ~
14	REF	* 1E * 111111111 ~
15	REF	* LU * 1111111111 ~
16	HL	* 2 * 1 * 22 * 0 ~
17	SBR	* P * 18 * GRGRGRGRG * . * . * 6 * . * . * CI ~
18	NM1	* IL * 1 * Xxxxxx * Xxxxx * X * . * . * MI * 111111 ~
19	N3	* 111 SOME AVE ~
20	N4	* SOME CITY * CA * 11111 ~
21	DMG	* D8 * 11111111 * M ~
22	NM1	* PR * 2 * HEALTH ADMIN * . * . * . * . * PI * 11111 ~
23	REF	* FY * NOCD ~
24	CLM	* 11111111 * 55 * . * . * 11 : . : 1 * Y * C * Y * Y ~
25	REF	* D9 * 111111111 ~
26	NM1	* 82 * 1 * Xxxxx * Xxxx * . * . * . * XX * 1111111111 ~
27	PRV	* PE * ZZ * 204E00000X ~
28	REF	* EI * 111111111 ~
29	LX	* 1 ~
30	SV3	* AD : D0150 * 55 * 11 * . * . * 1 ~
31	DTP	* 472 * D8 * 17760704 ~
32	SE	* 32 * 000000001 ~

When you are done viewing documents in the batch document hierarchy, you can close all the pop-up browser windows until the top-level parent document remains in the original document Viewer window. From here, you may return to other Management Portal activities.

2.2.3 Testing a Transformation

To test a transformation:

1. Check **Transform Document?**.

2. For **Choose Data Transformation**, select a data transformation.

3. For **Choose Display Option**, select one of the following:

   - **Transformation Result Only** — Display only the transformed document.

   - **Original Message and Result Together** — Display both the original document and the transformed document.

4. Now do either or both of the following:

   - Click **OK** to display the transformed document.

   - Click **Save Result To File?** to save the transformed document to a file. In this case, also specify a path and filename.

     The default directory is the management directory for the active namespace. For example, if you installed Ensemble into the directory C:\MyCache and your current namespace is ENSDEMO, the file is saved as C:\MyCache\Mgr\ENSDEMO*filename*

2.3 X12 Classes

For reference, this section lists the classes that Ensemble provides to enable you to work with X12 documents.

Item	Classes	Notes
X12 business services[*]	- EnsLib.EDI.X12.Service.FTPService - EnsLib.EDI.X12.Service.FileService - EnsLib.EDI.X12.Service.TCPService	Each of these X12 business service classes uses a different adapter, as indicated by the class name.
X12 business process[*]	EnsLib.MsgRouter.VDocRoutingEngine	This class is the standard virtual document business process.
X12 business operations[*]	- EnsLib.EDI.X12.Operation.FTPOperation - EnsLib.EDI.X12.Operation.FileOperation - EnsLib.EDI.X12.Operation.TCPOperation	Each of these X12 business operation classes uses a different adapter, as indicated by the class name.
Messages	EnsLib.EDI.Document (automatically used by the business host classes)	This is a specialized message class to carry X12 documents as Ensemble virtual documents.
Search table	EnsLib.EDI.X12.SearchTable	This is a specialized search table class for X12 documents.

[*]When you configure a production to work with X12 documents, the Management Portal automatically selects the appropriate business host class.

You can also create and use subclasses of these classes.

The business host classes include configurable targets. The following diagram shows some of them:
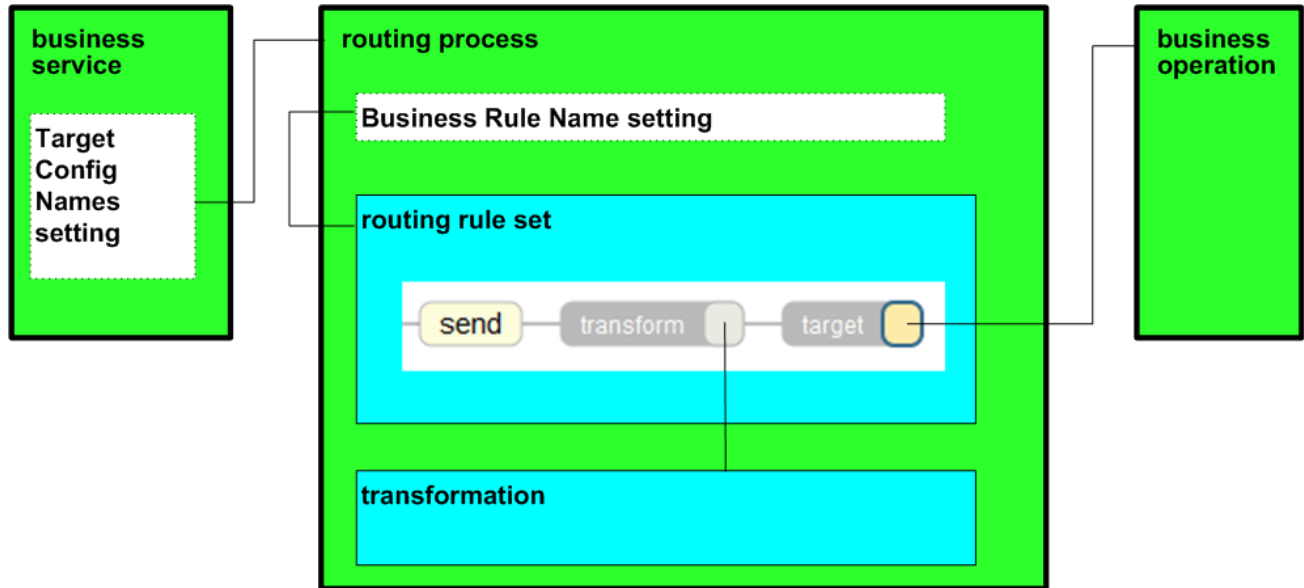
For information on other configurable targets, see "Reference for Settings."

2.4 Creating Custom X12 Schemas

To create custom X12 schemas, do the following in Studio:

1. From the **File** menu, choose **New**.

2. The New dialog displays. Select the **Custom** tab.

3. Select the **HL7 Schema** icon and click **OK**.

    Note: Ensemble uses the same syntax to represent schema category definitions for ASTM, EDIFACT, HL7, and X12. Click **HL7 Schema** to create a custom schema category for any of these formats.

4. Enter a **New Schema Name**. This is the <Category> name of the new schema.

5. Enter a **Base Schema Name**. Specify the <Category> name of the schema base.

6. Click **OK**.

You are now ready to edit your new file to add and remove XML statements from the <Category> block. The basic editing steps are as follows:

1. Define custom segments using <SegmentStructure> elements.

2. Define custom <MessageStructure> elements that contain the custom segments.

3. Define custom <MessageType> elements that contain the custom message structures.

4. From the **File** menu, choose **Save**. This action both saves and compiles the new category definition.

5. Try viewing the new category definition from the Management Portal. See Portal Tools.

For information on creating custom schema categories, see "Creating Custom Schema Categories" in *Ensemble Virtual Documents*.

2.5 Defining X12 Search Tables

The X12 search table class, EnsLib.EDI.X12.SearchTable, automatically indexes the X12 document ID, which it gives the name Identifier.

If you need more items to search, you can create a subclass. The subclass inherits the Identifier property, plus the infrastructure that makes search tables work. For details, see "Defining a Search Table Class" in *Ensemble Virtual Documents*.

Be sure to perform all tasks in the same namespace that contains your production. Do not use reserved package names; see "Reserved Package Names" in *Developing Ensemble Productions*. Also see "Overriding the Validation Logic" in *Ensemble Virtual Documents*.

3

Configuring the Production

This chapter describes the process of creating and configuring an X12 Production. It contains the following sections:

- Creating a new X12 Production

- Adding an X12 Business Services

- Adding an X12 Business Processes

- Adding an X12 Routing Rule

- Adding an X12 Data Transformation

- Adding an X12 Business Operations

Be sure to perform all tasks in the same namespace that contains your production. When you create rule sets and transformations do not use reserved package names; see "Reserved Package Names" in *Developing Ensemble Productions*.

Also see "Overriding the Validation Logic" in *Ensemble Virtual Documents*.

3.1 Creating a new X12 Production

You can add X12 components to an already existing production. However, if you want to create a new Ensemble production explicitly for handling X12, follow the steps below.

1. In the Management Portal, switch to the appropriate namespace.

   To do so, click **Switch** in the title bar, click the namespace, and click **OK**.

2. Click **Ensemble**.

3. Click **Configure**.

4. Click **Production** and then click **Go**.

   Ensemble then displays the last production you accessed, within the **Production Configuration** page.

5. Click the **Actions** tab on the **Production Settings** menu.

6. Click **New** to invoke the Production Wizard.

7. Enter a **Package Name**, **Production Name**, and **Description**.

8. Choose the **Generic** production type and click **OK**.

Ensemble creates a blank production from which you can add components such as business services, business processes, and business operations. See the sections below for more details.

Note: As you build your production, it frequently happens that while configuring one component you must enter the name of another component that you have not yet created. *A clear naming convention is essential to avoid confusion.* For suggestions, see "Naming Conventions" in *Ensemble Best Practices*. For rules, see "Configuration Names," in *Configuring Ensemble Productions*.

3.2 Adding an X12 Business Service

Add one X12 business service for each application or source from which the production will receive X12 documents.

To add an X12 business service to a production:

1. Access the Business Service Wizard as usual; see *Configuring Ensemble Productions*.

2. Click the **X12 Input** tab.

3. Click one of the following from the **Input type** list:

    • **TCP**

    • **File**

    • **FTP**

4. For **X12 Service Name**, type the name of this business service.

5. For **X12 Service Target**, select one of the following:

    • **Create Target Automatically** — Ensemble adds a business process to the production and configures the business service to use it as a target. You can edit the business process details later.

    • **None for Now** — Do not specify a target for this business service. If you make this selection, ensure that you specify a target later.

    • **Choose an Existing Production Item as Target** — In this case, also select an existing business host from the drop-down list.

6. Click **OK**.

3.3 Adding an X12 Business Process

To add an X12 business process to a production:

1. Access the Business Process Wizard as usual; see *Configuring Ensemble Productions*.

2. Click the **X12 Router** tab; the router class defaults to EnsLib.MsgRouter.VDocRoutingEngine.

3. For **Routing Rule Name**, do one of the following:

    • Select an existing routing rule from the **Routing Rule Name** drop-down list.

    • Select **Auto-Create Rule** and type a rule name into **Routing Rule Name**. In this case, the wizard creates the routing rule class in the same package as the production.

Later you must edit the routing rule and add your logic to it.

4. For **X12 business process Name**, type the name of this business process.

5. Click **OK**.

6. Ensure that your X12 business service is connected to the new X12 Business Process. To connect the process:

   • Select your X12 business service.

   • Click the **Settings** tab and open the **Basic Settings** menu in the menu to the right of the screen.

   • Enter the name of the new X12 business process in the **Target Config Names** field.

7. Configure additional settings of the business process, as needed. For details, see "Settings for X12 Business Processes".

3.4 Adding an X12 Routing Rule

For general information on defining business rules, see *Developing Business Rules*.

When you create an X12 routing rule:

• On the **general** tab, **Rule Type** should be **Virtual Document Message Routing Rule**. This choice sets the following options:

   – **Rule Assist Class** should be EnsLib.MsgRouter.VDocRuleAssist

   – **Context Class** should be EnsLib.MsgRouter.VDocRouting Engine

• In the **constraint** for a rule, specify **Message Class** as EnsLib.EDI.X12.Document.

In all other respects, the structure and syntax for both types of rule set are the same.

3.5 Adding an X12 Data Transformation

Your routing rule may need one or more data transformations.

For general information on defining DTL data transformations, see *Developing DTL Transformations*.

When you create a DTL data transformation for X12 documents:

• On the **Transform** tab, **Source Class** and **Target Class** should both be EnsLib.EDI.X12.Document.

• **Source Doc Type** should match the schema category name assigned by the business service.

• **Target Doc Type** should be the name of the target schema category. This must match a schema category name that you have loaded into Ensemble.

Use the X12 Document Viewer Page to test your transformations, as described in the chapter "Available Tools."

To integrate the DTL data transformation in the production, enter its full package and class name in the **Transform** field of a routing rule.

Note: Be sure your data transformation reflects your Batch Handling settings.

3.6 Adding an X12 Business Operation

To send X12 messages from a production to a file or application, you must add an X12 business operation. Add an X12 business operation for each output destination.

You might also want to add business operations to handle bad messages (for background, see "Business Processes for Virtual Documents" in *Ensemble Virtual Documents*).

To add an X12 business operation to a production:

1. Access the Business Operation Wizard as usual; see *Configuring Ensemble Productions*.

2. Click the **X12 Output** tab.

3. Click one of the following from the **Output type** list:

   - **TCP**

   - **File**

   - **FTP**

4. For **X12 Operation Name**, type the name of this business operation.

5. Click **OK**.

6. Ensure that the business operation is connected to the relevant business services or business process

   - For a routing rule, enter the name of your X12 business operation in the **Target** field of the routing rule set.

   - If your design uses a pass-through interface that simply relays messages from the incoming business service to the outgoing business operation, enter the name of your X12 business operation in the **Target Config Names** field of the X12 business service.

7. Configure additional settings of the business operation, as needed. For details, see "Settings for X12 Business Operations".

If you want the production to send data that is not an X12 message, see "Defining Business Operations" in *Developing Ensemble Productions*. Also see "Connectivity Options" in *Introducing Ensemble*.

4

Handling X12 Interchanges

As X12 documents are sent within Interchanges you need to choose how Ensemble should handle incoming documents. Especially if you need to access documents at the Group or Transaction Set level. This chapter describes different scenarios for handling X12 Interchanges

- X12 Batch Handling

- Configuring Business Processes for Whole Batch

- Configuring Business Processes for Other Batch Settings

4.1 X12 Batch Handling

Batch Handling settings determine how Ensemble sends and receives documents. The choices you make here affect how your Business Processes should be configured.

4.1.1 Receiving Batch Documents

X12 business services have the Batch Handling configuration setting, which determines how to process incoming batch documents. The options are:

- `Whole Batch` — Do not process child documents individually; accumulate and send the whole batch as one composite document.

- `Single-Session Batch` — Forward all documents in the Interchange as part of a single session, including final parent document objects containing batch and group header and trailer segments.

- `Multi-Session Batch` — Forward each document in the Interchange in its own session, followed by final parent document objects containing the batch and group header and trailer segments.

- `Individual` — Forward each child document in the batch in its own session; do not forward parent batch document objects.

Note: If you select **Whole Batch** any transformations of individual Transaction Sets will have to account for the Interchange and Group levels. See Configuring Business Processes for Whole Batch below.

4.1.2 Sending Batch Documents

On the outgoing side, X12 File and FTP business operations have the Auto Batch Parent Segs configuration setting. When **Auto Batch Parent Segs** is False (the default) the business operation outputs child documents, but does not output the batch headers and trailers. When **Auto Batch Parent Segs** is True, while outputting a message that has a batch parent, the business operation outputs the batch headers first, then the child documents, then follows up with the batch trailers when triggered by the final batch header message or by a file name change.

4.1.3 Batch Modes

The combination of **Batch Handling** and **Auto Batch Parent Segs** enables the following modes of operation for X12 batch documents:

Batch Handling	Auto Batch Parent Segs	Results
Whole Batch	(any)	Business service sends only the parent document; all child documents are referenced to it but not sent individually. Operation outputs entire batch at one time when it receives the parent document.
Single-Session or Multi-Session	True	Service sends each child document as it receives and parses it, followed by the parent document when all children have been sent. The business operation outputs parent headers when it receives the first child document, then finishes up with trailers when it receives the parent document object. Trailer segments automatically contain the correct child count values.
Single-Session or Multi-Session	False	This results in double output: the business operation sends out each child document individually, followed by the parent document containing each child document (again).
Individual	False	Business service forwards each child document in the batch in its own session and does not forward objects representing the batch headers and trailers. On the outgoing side, the business operation does the same.

4.2 Configuring Business Processes for Whole Batch

If you choose to have the business service process incoming X12 documents in Whole Batch mode you will need to configure the business process accordingly.

For examples of X12 Productions that use the Whole Batch setting please view the Demo.X12.SorterDTL.ByGroup.Production and Demo.X12SorterDTL.ByInterchange.Production demo productions. To view them, ensure that you are within the ENSDEMO namespace and navigate to the Production List page. Select the production you wish to view and click open. These demos incorporate all of the configuration settings listed below.

4.2.1 Routing Rules for Whole Batch

To properly route documents in Whole Batch mode, set the routing rule to accept Interchanges. Send to the relevant data transformation if necessary.

4.2.1.1 Create a new routing rule

1. Ensure you are in the proper namespace and navigate to the Ensemble Rule Editor page.

2. Click **New**.

3. Follow the instructions in Adding an X12 Routing Rule.

4.2.1.2 Edit the constraint

1. Double click on the **constraint** rule item.

2. Set the **Source** to your business service.

3. Set the **Message Class** to EnsLib.EDI.X12.Document.

4. Set the **Schema Category** as appropriate for your production.

5. Choose Interchange from the **Document Name** selection.

For more information, see Using the Rule Constraint Editor.

4.2.1.3 Define the condition

Choose any conditional statement relevant to your production. If you always want the action to be applied, click on the **condition** rule item and type "1".

4.2.1.4 Add a Send action.

1. Click on the **when** rule item.

2. Click **Send** to add a send action.

3. Double click on the **transform** rule item and select the appropriate data transformation.

4. Double click on the **target** rule item and select the appropriate business operation from your production.

For more information, see Selecting the Transformation and Target of a Send Action.

4.2.2 Data Transformations for Whole Batch

To transform whole batch documents at the Group or Transaction Set level, you must iterate through each group and each transaction set within each group. To do so, follow the steps below.

4.2.2.1 Create a new Data Transformation

1. Create a new set action by clicking **Add Action** and selecting **set**.

2. Input a variable name into the **Property** field, such as ArrayOfGroups.

3. Input **##class(%Library.VariableName).%New()** into the **Value** field, where VariableName is the value you provided in the **Property** field.

4. Input " " into the **Key** field.

4.2.2.2 Create an Array of Groups

1. Create a new `set` action by clicking **Add Action** and selecting **set**.

2. Input a variable name into the **Property** field, such as `ArrayOfGroups`.

3. Input **##class(%Library.VariableName).%New()** into the **Value** field, where `VariableName` is the value you provided in the **Property** field.

4. Input `" "` into the **Key** field.

4.2.2.3 Loop over the Groups in each Interchange

1. Create a new `for each` action by clicking **Add Action** and selecting **for each**.

2. Input **source.{GroupDocsRef}** into the **Property** field.

3. Input `Group` into the **Key** field.

4.2.2.4 Loop over the Transaction Sets in each Group

1. Create a new `for each` action by clicking **Add Action** and selecting **for each**.

2. Be sure the for each is between the **for each** and **endeach** blocks of the previous action.

3. Input **Group.{TransactionSetDocsRef}** into the **Property** field.

4. Input `TSIn` into the **Key** field.

Add any additional actions necessary. For more information, see Developing DTL Transformations.

4.3 Configuring Business Processes for Other Batch Settings

If you choose to have the business service process incoming X12 documents in Whole Batch mode you will need to configure the business process accordingly.

For an example of an X12 production that use Single Batch Setting please view the Demo.X12.SorterProcess demo production. To view it, ensure that you are within the ENSDEMO namespace and navigate to the Production List page. Select the production you wish to view and click open. This demo incorporates all of the configuration settings listed below.

4.3.1 Routing Rules for Other Batch Settings

To properly route documents in Single Session, Multi Session, or Individual mode, create a routing rule set with a unique routing rule for each type of X12 document the production will encounter.

To create new rules within the rule set:

1. Click on **ruleSet**

2. Click **rule** to add a rule

3. Follow the instructions set in Routing Rules for Whole Batch

4. When editing the constraint, be sure to set the **Document Name** to the appropriate X12 document type.

5.   When adding a data transformation, be sure to use a transformation written for the appropriate X12 document type.

4.3.2 Data Transformations for Other Batch Settings

To transform X12 documents within productions with Single Session, Multi Session, or Individual mode you must create a separate data transformation for each type of X12 document the production will encounter. For more information, see Developing DTL Transformations.

Reference for Settings

This section provides the following reference information:

- Settings for X12 Business Services

- Settings for X12 Business Processes

- Settings for X12 Business Operations

For information on settings for the business process (EnsLib.MsgRouter.VDocRoutingEngine), see "Settings of a Virtual Document Routing Process" in *Ensemble Virtual Documents*.

Settings for X12 Business Services

Provides reference information for settings of an X12 business service.

Summary

X12 business services have the following settings:

Group	Settings	See
Basic Settings	Target Config Names, Doc Schema Category	"Settings for Business Services" in *Ensemble Virtual Documents*
Additional Settings	Search Table Class, Validation	"Settings for Business Services" in *Ensemble Virtual Documents*
	Reply Target Config Names, Reply Mode, Batch Handling, Batch Error Action, Batch Reply Type, Local Application ID, Tolerate Newlines, Default Char Encoding	*sections in this topic*

The remaining settings are either common to all business services or are determined by the type of adapter. For information, see the section "Reference for Settings" in each of the following books:

- "Settings for All Business Services" in *Configuring Ensemble Productions*

- "Settings for the File Inbound Adapter" in *Using File Adapters with Ensemble*

- "Settings for the FTP Inbound Adapter" in *Using FTP Adapters with Ensemble*

  EnsLib.EDI.X12.Adapter.TCPInboundAdapter has Job Per Connection set to False, which is usually appropriate for X12.

Batch Error Action

Specifies what to do when a validation error is detected in a batch Interchange document. Options include:

- `Reject With All Errors` — Reject the whole batch if any error is found in any document within it. Enumerate all errors found in the batch if Batch Reply Type allows for reporting them. This prevents forwarding any documents in a batch until all have been read and validated.

- `Reject On First Error` — Reject the whole batch when the first error is found in any document within it. Do not bother checking for more errors or parsing any further contents of the Interchange. This prevents forwarding any documents in a batch until all have been read and validated.

- `Reject Individual Errors` — Reject only those documents within the Interchange that have errors. Forward each acceptable child document to its target(s) as soon as it has been read and validated. This is the default.

If Reply Mode is `Application` and Batch Error Action is not `Individual`, it could happen that Ensemble forward some of the documents in a batch before rejecting the whole batch upon encountering an error.

Batch Handling

X12 Transaction Set documents are often packaged in a batch document called an *Interchange* which contains nested sub-batches called *Functional Groups*. The **Batch Handling** setting specifies how Ensemble treats received document batches. The options are:

- `Whole Batch` — Do not process child documents individually; accumulate and send the whole batch as one composite document.

- `Single-Session Batch` — Forward all documents in the batch together in one session; the session includes objects representing the parent document header and trailer segments. This is the default.

- `Multi-Session Batch` — Forward each document in the batch in its own session, including the objects representing the batch header and trailer segments.

- `Individual` — Forward each child document in the batch in its own session; do not forward objects representing the parent batch document's header and trailer segments.

Batch Reply Type

Specifies the type of batch reply to create for an Interchange batch that has been received. The following table lists the possible choices:

Value	Meaning
None	Do not generate a batch reply. If an error occurs, do not create any immediate notification reply to the sender.
All	Generate a reply Interchange containing a reply notification for every TransactionSet received in the Interchange.
All+TA1	Generate a reply Interchange containing a TA1 segment that indicates acceptance or error status for the entire Interchange, and a reply notification for every TransactionSet received in the Interchange.
Errors	Whether or not errors are found, generate a reply Interchange. If no errors are found, generate an empty reply Interchange. If errors are found, generate an Interchange that contains reply notifications only for TransactionSets in which errors are detected. This is the default setting if no choice is specified.
OnlyIfErrors	If errors are found, generate a reply Interchange that contains reply notifications only for TransactionSets in which errors are detected.
Successes	Whether or not errors are found, generate a reply Interchange. If errors are found for every TransactionSet, generate an empty reply Interchange. Otherwise, generate a reply Interchange that contains reply notifications only for TransactionSets in which no errors are detected (successes).
TA1	Generate a reply Interchange containing only a TA1 segment that indicates acceptance or error status for the whole Interchange received.
OnlyIfErrorTA1	If errors are found, generate a reply Interchange that contains only a TA1 segment that indicates error status for the whole Interchange received.
ISA14-TA1	If field ISA:14 of the incoming ISA header segment is set to 1, generate a reply Interchange containing only a TA1 segment; otherwise return nothing.
ISA14-OnlyIfErrorTA1	If errors are found and field ISA:14 of the incoming ISA header segment is set to 1, generate a reply Interchange containing only an error TA1 segment; otherwise return nothing.
Byte	Generate a reply consisting of a single character code: 'A' if the entire Interchange is accepted, 'R' if it is rejected due to one or more errors.

All of the options that relate to TA1 segments are used to force a TA1 segment to be generated, often as the only body segment of the reply interchange. This convention is used to represent the presence or absence of errors in the entire inbound Interchange. However, if an error is found in the incoming ISA or IEA that can only be reported in a TA1 segment, then a TA1 is generated even if the configured setting does not force a TA1 to appear.

Default Char Encoding

Specifies the character set of the input data. Ensemble automatically translates the characters from this character encoding. Supported values are UTF-8 or any member of the Latin*n* family. The value Native means to use the native encoding of the Ensemble server.

Placing a @ (at sign) character at the beginning of this field means that the field identifies an internal NLS Translation Table instead of a logical character encoding.

The default depends on the adapter.

For background information on character translation in Caché, see "Localization Support" in the *Caché Programming Orientation Guide*.

Local Application ID

Colon-separated *LocalID*:*Qualifier* code that represents the facility and application that receive X12 documents via this business service. These are used to create reply document headers. The @ (at sign) character represents using the corresponding field from the incoming document. If your ID must contain a literal @ symbol, escape it with back slash: \@

The default value is:

EnsembleX12Service:03

Reply Mode

Specifies how to issue X12 reply documents (such as TA1 and 997). Options include:

- Never — Do not send back any reply.

- Immediate — Send a reply from the business service immediately upon receipt of an Interchange. This is the default.

- Application — Wait for a response from the target configuration item. When it arrives, relay the reply back to the sender. If validation fails or some other error occurs, generate an immediate reply according to the option selected for **BatchReplyType**.

Reply Target Config Names

(**File** and **FTP** only) Specifies a comma-separated list of configuration items within the production to which the business service should relay any X12 *reply* documents that it receives (such as TA1 and 997). Usually the list contains one item, but it can be longer. The list can include business processes or business operations, or a combination of both.

Compare to Target Config Names.

Tolerate Newlines

True or False. If True, the business service processes an incoming X12 file without error, even if new lines have been inserted into the file after (or in place of) segment terminators to enhance readability. If False, these extra new lines trigger an error in parsing the file. The default is True.

Settings for X12 Business Processes

Provides reference information for settings of an X12 business process.

Summary

X12 business processes have the following settings:

Group	Settings	See
Basic Settings	Validation	*section in this topic*
	Business Rule Name	"Settings for Routing Processes" in *Ensemble Virtual Documents*
Additional Settings	Response Target Config Names	*sections in this topic*
	Act On Transform Errors, Act On Validation Errors, Alert On Bad Message, Bad Message Handler, Response From, Response Timeout, Force Sync Send	"Settings for Routing Processes" in *Ensemble Virtual Documents*
Development and Debugging	Rule Logging	"Settings for Routing Processes" in *Ensemble Virtual Documents*

The remaining settings are common to all business processes. See "Settings for All Business Processes" in *Configuring Ensemble Productions*.

Response Target Config Names

A comma-separated list of configured items within the production. If specified, this list identifies the destinations, in addition to the caller, to which responses will be forwarded. If empty, responses are only returned to the caller. This setting takes effect only if the **Response From** field has a value.

Validation

The **Validation** setting of an X12 Document Router controls how the router validates the incoming message. If the incoming message fails the specified validation, Ensemble reports the failure in the event log, and the X12 routing process passes the message to its bad message handler only; see the Bad Message Handler setting. If the message fails the specified validation but there is no bad message handler, an error is logged but the message is not sent to any target. If the message passes validation, the X12 routing process sends the message to the targets specified by the routing rules.

Ideally, you can use routing rules and data transformations to ensure each message is acceptable to the target system, and can, consequently, avoid using validation. This ensures that all messages are processed by the appropriate target. If you enable validation, Ensemble applies the validation tests before the routing rules. Any message that fails validation will not be sent to a target based on the routing rules; it will be sent only to the Bad Message Handler. However, there are some environments where X12 message validation is the preferred way to filter messages. For example, in the following situations, using X12 validation is a good choice:

- You are developing or debugging an interface and want to determine the kind of message variants that your system needs to handle.

- The target application cannot handle messages that have variances from the specification, and the routing rules and transformations cannot resolve those variances.

- There is a regulatory or other business requirement that the messages conform to the specification.

X12 validation does add overhead to the routing process. This overhead can be significant and can reduce the maximum load of messages that your production can handle.

The Validation property allows you to specify flags that control the following:

- Whether the message has a valid document type.

- Whether the message structure is validated.

If you specify an empty string as the Validation property value, the message router skips validation and routes all messages. When you create a new X12 routing process in the Management Portal, the **Validation** setting is initialized to an empty string.

The message validation sequence is as follows:

1. Validate that the message object has a DocType property value.

2. Validate the segment order.

Note: A message can pass validation and not conform exactly to the schema definition depending on the **Validation** flags specified.

The following table lists the X12 validation flags and describes how the routing process validates the message when each is specified.

Flag	Routing Process
d	Validation examines the DocType property of the document to see if it has a value.
m	Validation verifies that the document segment structure is well formed, and that it can be parsed using the schema identified in the DocType property of the document. This ensures that all required segments in the schema definition are included in the message and that the message does not contain any misplaced segments that are not allowed by the schema.
dm	Both d and m are active.
1	Same as dm, for backward compatibility with previous releases.
(empty string)	Skips validation and routes all messages. This is the default.

Settings for X12 Business Operations

Provides reference information for settings of an X12 business operation.

Summary

X12 business operations have the following settings:

Group	Settings	See
Basic Settings	File Name	*section in this topic*
Additional Settings	Search Table Class, Default Char Encoding	"Settings for Business Operations" in *Ensemble Virtual Documents*
	Auto Batch Parent Segs, No Fail While Disconnected, Separators, Validation, Reply Code Actions, Failure Timeout	*sections in this topic*

The remaining settings are either common to all business operations or are determined by the type of adapter. For information, see:

- "Settings for All Business Operations" in *Configuring Ensemble Productions*

- "Settings for the File Outbound Adapter" in *Using File Adapters with Ensemble*

- "Settings for the FTP Outbound Adapter" in *Using FTP Adapters with Ensemble*

- "Settings for the TCP Outbound Adapter" in *Using TCP Adapters with Ensemble*

    EnsLib.X12.Adapter.TCPOutboundAdapter has the following settings configured appropriately for X12:

    – **Connect Timeout** has its usual default of 5 seconds, but has a maximum limit of 30,000 seconds.

    – **Get Reply** is set to False. This means the adapter will wait to read a reply message back from the socket before returning.

    – **Response Timeout** has a default of 30 instead of its usual 15, and has a maximum limit of 30,000 seconds.

Auto Batch Parent Segs

(**File** and **FTP** only) If True, when writing a document that has a batch parent, output the batch header segments first, then child documents, then follow up with the batch trailer segments when triggered by the final batch header document object or by a file name change. If False, omit headers and trailers and output child documents only. The default for X12 is True.

Default Char Encoding

Specifies the desired character set of output data. Ensemble automatically translates the characters to this character encoding. For X12 output, the default is `Latin1`. See "Default Char Encoding" in "Settings for X12 Business Services."

Failure Timeout

The number of seconds during which to continue retry attempts. After this number of seconds has elapsed, the business operation gives up and returns an error code. X12 business operations automatically set this value to –1 for *never time out* to ensure that no X12 document is skipped.

File Name

(**File** and **FTP** only) Output file name. This setting can include Ensemble time stamp specifiers. If you leave **File Name** blank, the default value is `%f_%Q` where:

- `%f` is the name of the data source, in this case the input filename

- `_` is the literal underscore character, which will appear in the output filename

- `%Q` indicates ODBC format date and time

In substituting a value for the format code `%f`, Ensemble strips out any of the characters |,?,\,/,:,[,],<,>,&,,,;,NUL,BEL,TAB,CR,LF, replacing spaces with underscores (_), slashes (/) with hyphens (-), and colons (:) with dots (.).

For full details about time stamp conventions, including a variety of codes you can use instead of the default `%f_%Q`, see "Time Stamp Specifications for Filenames" in *Configuring Ensemble Productions*.

No Fail While Disconnected

(**TCP** only) If True, suspend counting seconds toward the **Failure Timeout** while disconnected from the TCP server. This setting does not apply if **Failure Timeout** is –1 or if **Stay Connected** is 0.

Reply Code Actions

(**TCP** only) When the adapter setting **Get Reply** is True, this setting allows you to supply a comma-separated list of code-action pairs, specifying which action the business operation will take on receipt of various types of acknowledgment documents. The format of the list is:

code=action , code=action , ... code=action

Where *code* represents a literal value found in field TA1:4, AK5:1, or AK9:1 of the acknowledgment document. The following table lists the expected values for *code*.

Code	Meaning
A	Accepted
E	Accepted, But Errors Were Noted
R	Rejected
M	Rejected; Message Authentication Code (MAC) Failed
W	Rejected; Failed Validity Tests
X	Rejected; Content Decryption Failed
~	The tilde character matches replies that do not contain a TA1, AK5 or AK9 segment
_	The underscore character matches replies with an empty value in the field. An empty or whitespace code value is the same as _
*	The asterisk character matches any value not matched otherwise (default=S)
I?	Matches the case in which the reply ControlId does not match the ControlId of the original document

The following values for *action* may be used alone or combined to form strings. S is the default *action* if no other is given, except for A whose default action is C:

Action	Meaning
C	Treat the document as Completed OK. Code A has a default action of C.

Action	Meaning
W	Log a warning but treat the document as Completed OK.
R	Retry the document according to the configured RetryInterval and FailureTimeout; finally Fail unless a different action is also specified
S	Suspend the document, log an error, and move on to try the next document. S is the default action for all codes except for code A, which has a default action of C.
D	Disable the business operation, log an error and restore the outbound document to the front of the business operation's queue
F	Fail with an error and move on to try the next document

The default value for this setting string is:

A=C,*=S,~=S,I?=W

This means:

- A=C — When the action is accepted, treat the document as Completed OK.

- I?=W — When the reply ControlId does not match the ControlId of the original document, log a warning but treat the document as Completed OK.

- *=S,~=S — In all other cases, including when replies that do not contain a TA1, AK5 or AK9 segment, suspend the document, log an error, and move on to try the next document.

Separators

A string of separator characters which Ensemble assigns to X12 separators in left to right order as described below.

If the **Separators** string is empty, the default is to use the current default separators and segment terminators for X12, plus a carriage return (ASCII 13) and line feed (ASCII 10).

*:\a~\r\n

An X12 document uses special characters to organize its raw contents. These characters may vary from one clinical application to another. For non-empty values of **Separators**, positions 1 through 3 (left to right) are interpreted as follows:

1. Data Element Separator (ES)

2. Component Separator (CS)

3. Data Element Repeat Separator (RS)

The default values for positions 1 through 3 are:

1. * (asterisk)

2. : (colon)

3. \a (record separator)

For **Separators**, you must supply a string of three characters which Ensemble assigns to X12 separators in left to right order: ES, CS, RS, as described in the previous list.

Any characters in positions 4 through 6 override the default segment terminator character, which is ~ (tilde). You may specify from 0 to 3 characters in positions 4 through 6 using the following:

- \r for the carriage return (ASCII 13)

- \n for the line feed (ASCII 10)

- \a for the array record separator (ASCII 30)

You can use \x in positions 1 through 3 if you need to specify segment terminators in positions 4 and higher but want your output documents to use fewer than 3 separators. Separators designated by \x in positions 1 through 3 are not used. The purpose of \x is simply to extend the length of the list of separators so that position 4 is interpreted correctly as the first segment terminator.

Validation

Any non-empty string triggers basic validation of the outgoing document. If the **Validation** field is left empty, no validation of the outgoing document is performed.