



webMethods EDI Module

User's Guide

VERSION 6.5.2

webMethods, Inc.
South Tower
3877 Fairfax Ridge Road
Fairfax, VA 22030
USA
703.460.2500
<http://www.webmethods.com>

webMethods Access, webMethods Administrator, webMethods Broker, webMethods Dashboard, webMethods Developer, webMethods Fabric, webMethods Glue, webMethods Installer, webMethods Integration Server, webMethods Mainframe, webMethods Manager, webMethods Modeler, webMethods Monitor, webMethods Optimize, webMethods Portal, webMethods Servicenet, webMethods Trading Networks, and webMethods Workflow are trademarks of webMethods, Inc. webMethods and the webMethods logo are registered trademarks of webMethods, Inc.

Acrobat and Adobe are registered trademarks, and Reader is a trademark of Adobe Systems Incorporated. Amdocs is a registered trademark, and ClarifyCRM is a trademark of Amdocs. Ariba is a registered trademark of Ariba, Inc. BEA, BEA WebLogic Server, Jolt, and Tuxedo are registered trademarks, and BEA WebLogic Platform is a trademark of BEA Systems, Inc. Action Request System, BMC Software, PATROL, and Remedy are registered trademarks of BMC Software, Inc. BroadVision is a registered trademark of BroadVision, Inc. ChemeStandards and CIDX are trademarks of Chemical Industry Data Exchange. Unicenter is a registered trademark of Computer Associates International, Inc. PopChart is a registered trademark of CORDA Technologies, Inc. Kenan and Arbor are registered trademarks of CSG Systems, Inc. Data Connection and SNAP-IX are registered trademarks of Data Connection Corporation. DataDirect, DataDirect Connect, and SequelLink are registered trademarks of DataDirect Technologies. D&B and D-U-N-S are registered trademarks of Dun & Bradstreet Corporation. Entrust is a registered trademark of Entrust, Inc. papiNet is a registered trademark of the European Union and the United States. Financial Information eXchange, F.I.X., and F.I.X Protocol are trademarks of FIX Protocol Ltd. UCCnet and eBusinessReady are registered trademarks, and 1SYNC and Transora are trademarks of GS1 US. Hewlett-Packard, HP, HP-UX, OpenView, PA-RISC, and SNAplus2 are trademarks of Hewlett-Packard Company. i2 is a registered trademark of i2 Technologies, Inc. AIX, AS/400, CICS, DB2, Domino, IBM, Informix, Infoprint, Lotus, Lotus Notes, MQSeries, OS/390, OS/400, RACF, RS/6000, SQL/400, S/390, System/390, VTAM, z/OS, and WebSphere are registered trademarks; and Communications System for Windows NT, DB2 Universal Database, IMS, MVS, and SQL/DS are trademarks of IBM Corporation. InnoDB is a trademark of Innobase Oy. Itanium is a registered trademark of Intel Corporation. JBoss is a registered trademark, and JBoss Group is a trademark of Jboss, Inc. Linux is a registered trademark of Linus Torvalds. W3C is a registered trademark, and X Window System is a trademark of the Massachusetts Institute of Technology. MetaSolv is a registered trademark of Metasolv Software, Inc. ActiveX, Microsoft, Outlook, Visual Basic, Windows, and Windows NT are registered trademarks; and Windows Server is a trademark of Microsoft Corporation. Six Sigma is a registered trademark of Motorola, Inc. Firefox is a registered trademark, and Mozilla is a trademark of the Mozilla Foundation. MySQL is a registered trademark of MySQL AB. nCipher is a trademark of nCipher Corporation Ltd. Teradata is a registered trademark of NCR International, Inc. Netscape is a registered trademark of Netscape Communications Corporation. SUSE is a registered trademark of Novell, Inc. ServletExec is a registered trademark, and New Atlanta is a trademark of New Atlanta Communications, LLC. CORBA is a registered trademark of Object Management Group, Inc. JD Edwards, OneWorld, Oracle, PeopleSoft, Siebel, and Vantive are registered trademarks, and PeopleSoft Pure Internet Architecture and WorldSoftware are trademarks of Oracle Corporation. Infranet and Portal are trademarks of Portal Software, Inc. Red Hat is a registered trademark of Red Hat, Inc. PIP and RosettaNet are trademarks of RosettaNet, a non-profit organization. SAP and R/3 are registered trademarks of SAP AG. SWIFT and SWIFTNet are registered trademarks of Society for Worldwide Interbank Financial Telecommunication SCRL. SPARC and SPARCSStation are registered trademarks of SPARC International, Inc. SSA is a registered trademark, and Baan and SSA Global are trademarks of SSA Global Technologies, Inc. EJB, Enterprise JavaBeans, Java, JavaServer, JDBC, JSP, J2EE, Solaris, Sun, and Sun Microsystems are registered trademarks; and Java Naming and Directory Interface, SOAP with Attachments API for Java, JavaServer Pages, and SunSoft are trademarks of Sun Microsystems, Inc. Sybase is a registered trademark of Sybase, Inc. VERITAS is a registered trademark, and VERITAS Cluster Server is a trademark of Symantec Corporation. UNIX is a registered trademark of The Open Group. Unicode is a trademark of Unicode, Inc. VeriSign is a registered trademark of Verisign, Inc.

All other marks are the property of their respective owners.

Copyright © 2003-2006 by webMethods, Inc. All rights reserved, including the right of reproduction in whole or in part in any form.

Contents

About This Guide	15
Document Conventions	15
Additional Information	16
Part I. Using EDI Module without Other webMethods Components	17
 Chapter 1. Before You Can Process EDI Documents	19
Overview	20
Creating Flat File Schemas for EDI Documents	20
Creating a Flat File Schema from a SEF File	21
Creating a Flat File Schema from an IDOC	23
Migrating 4.x or Earlier Templates to Flat File Schemas	23
Creating Flat File Schemas for Internal-Format Documents	24
Configuring How the Format Services Convert Field Values	25
Associating the EDI Format Services to EDI Data Types	29
 Chapter 2. Creating Clients that Send EDI Documents to the Integration Server	31
Overview	32
Content Type to Use	32
Service to Invoke to Handle the EDI Document	33
Sending EDI Documents to the Integration Server via HTTP	33
Identifying the Service to Invoke	33
Logic to Include in the HTTP Client	33
Example of Input Variables for the pub.client:http Service	34
Sending EDI Documents to the Integration Server via FTP	35
Identifying the Service to Invoke	35
Logic to Include in the FTP Client	35
Example of Input Variables for the pub.client:ftp Service	36
Sending EDI Documents to the Integration Server via File Polling	37
Setting Up File Polling	37
Identifying the Service to Invoke	37
 Chapter 3. Receiving and Processing Inbound EDI Documents	39
Overview	40
Before Creating the Service to Process EDI Documents	40
Inputs to Your Service	41
Logic to Include in the Service to Process EDI Documents	41

Determining the Method to Use to Process	42
Processing the Entire Document at One Time	43
What Does the Envelope Validation and Compliance Check Entail?	47
Processing the Document Iteratively Segment by Segment	47
Customizing the EDI Flat File Schema	47
Logic for the Service to Process the Document Iteratively	50
Interchange (or Transmission) Envelope Validation and Compliance Check	52
Examples	53
Chapter 4. Generating Functional Acknowledgments	55
Overview	56
Before You Can Generate a Functional Acknowledgment	57
Configuring the Maximum Number of Transaction Errors	58
How the generateFA Service Reports the FA Status	59
Child Transaction that is Not Allowed in its Group	59
Transaction or Group is for an FA	60
Syntax Error Status	60
Logical Error Status	61
Child Transaction Rejected Status	62
How the EDI Module Determines Which FA Status to Use	63
Transaction FA Status	63
Group FA Status	64
UN/EDIFACT Interchange FA Status	65
Adding the Call to generateFA to Your Service	66
When You Process the Document at One Time	66
Invoking generateFA Before Invoking envelopeProcess	66
Invoking generateFA Immediately after Invoking envelopeProcess	67
When You Process the Document Iteratively Segment by Segment	68
Specifying Inputs for the generateFA Service	69
Output from the generateFA Service	69
Chapter 5. Mapping Data to Form New Documents	71
Overview	72
Before Mapping Data	72
Mapping EDI Documents to Other Formats	73
Examples	73
Creating Services that Map Data from a Segment of the EDI Document	73
Transforming Source Data Before Mapping it to the Target	77
Converting Field Values	77
Concatenating Strings	78
Best Practices for Creating Mapping Services	79

Building the Final Mapping Service From Segment-Mapping Services	80
Invoking the Final Service from the Service that Processes Your EDI Document	83
Mapping XML Documents to EDI Format	84
Creating Services that Map Data into a Segment of the EDI Document	84
Building the Final Mapping Service from Segment-Mapping Services	85
Best Practices When Mapping From XML to EDI	85
Chapter 6. Forming EDI Documents and Sending Them Outbound	87
Overview	88
Before Creating the Service to Form an EDI Documents	88
Inputs to Your Service	88
Logic to Include in the Service to Form an EDI Document	89
Validating the Input Internal-Format Document When it is an IData Object	91
Adding UN/EDIFACT Envelopes	91
Checking Your Work	92
Example	92
Chapter 7. Handling Large Documents	93
Overview	94
Configuring Large Document Handling	94
When is the EDI Module Configuration File Checked?	97
Creating Clients that Send Documents as InputStreams	97
Converting Documents to IData Objects Iteratively	98
Part II. Using EDI Module with webMethods Trading Networks	99
Chapter 8. Before You Can Use Trading Networks to Process EDI Documents	101
Overview	102
Setting Up Items Required for Structural Validation and Conversion	103
Installing TN Document Types and Creating Flat File Schemas	104
Flat File Schema Namespace Conventions	105
Flat File Schema Namespace Conventions (ANSI X12 and UN/EDIFACT)	105
Flat File Schema Namespace Conventions (TRADACOMS)	105
Attributes Stored for TRADACOMS Document Types	106
After Creating the Flat File Schemas	106
Configuring the EDI Module To Support New Versions of EDI Standards	107
Chapter 9. Defining Partner Information (ANSI X12 and UN/EDIFACT)	109
Overview	110
Using Standard or Non-Standard Processing	110
Advantages and Disadvantages of Standard vs Non-Standard Processing	111
Partner Information You Need to Define	112

Defining Trading Networks Profiles	113
Adding External ID Types for EDI ID Qualifiers	114
Adding and Overriding EDI ID Qualifiers	115
Defining EDI Trading Partner Agreements	116
Defining the Default EDITPA	116
Defining a Partner-Specific EDITPA	117
wm.b2b.edtn.TPA:EDITPA IS Document Type	118
Chapter 10. Defining Partner Information (TRADACOMS)	145
Overview	146
Partner Information You Need to Define	146
Defining Trading Networks Profiles	147
External ID Types for TRADACOMS	148
Optionally Configuring the External ID Types	149
Defining EDI Trading Partner Agreements	150
Defining the Default EDITPA	150
Defining a Partner-Specific EDITPA	151
wm.b2b.edtn.TPA:EDITPA IS Document Type	152
Storage Options For File Document Types	159
Errors That Prevent the Splitting of Transmissions	161
Chapter 11. Defining Control Number Information for Partners	163
Overview	164
Processing Inbound EDI Documents	166
Information You Can Set for Inbound Control Number Validation	167
Processing Outbound EDI Documents	168
Turning Inbound Control Number Validation On and Off	168
Defining Actions for Invalid Control Numbers	170
For ANSI X12 or UN/EDIFACT Documents	171
For TRADACOMS Documents	171
Control Number Cap, Minimum, Increment, and Window	172
Using the Settings to Determine Whether a Control Number is Valid	174
Using the Settings to Determine the Next Expected Control Number	175
Using the Settings to Determine the Type of Invalid Control Number	175
Duplicate Control Numbers	175
Out-of-Sequence Control Numbers	176
Defining Control Number Settings	177
Searching for Existing Control Number Settings	181

Chapter 12. Creating Clients that Send EDI Documents to Trading Networks	185
Overview	186
Content Type to Use	186
Content Type for EDI Documents	187
Content Types for EDI Documents with Mainframe Data	187
Service the Client Invokes	188
Submitting EDI Documents to Trading Networks via HTTP/S	188
Example of Input Variables for the pub.client:http Service	188
Submitting EDI Documents to Trading Networks via FTP	189
Example of Input Variables for the pub.client:ftp Service	190
Submitting EDI Documents to Trading Networks via File Polling	191
Using a Service to Send Multiple EDI Documents to Trading Networks	191
Chapter 13. Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks	193
Overview	194
Before You Can Process Inbound EDI Documents	195
Specifying EDITPA Variables that Affect Inbound Processing	196
Trading Networks Attributes and EDI Documents	196
System Attributes that the EDI Module Sets	197
DocumentID	197
GroupID	197
ConversationID	198
Enabling the Extraction of <i>ConversationIDs</i> From EDI Documents	198
Disabling the Extraction of <i>ConversationIDs</i> From EDI Documents	199
Custom Attributes that the EDI Module Sets	200
Defining Your Own Custom Attributes for EDI Documents	204
At Design Time Creating Attributes and Associating Them with TN Document Types	204
Setting the Values of Attributes at Run Time	204
Example	205
Defining a Processing Rule that Uses Your Custom Attribute as Criteria	205
Defining Processing Rules to Process Inbound EDI Documents	206
Specifying Processing Rule Criteria	207
Specifying Pre-Processing Actions	208
Specifying Processing Actions	209
Coding Services to Process Transaction and Group Documents	210
Information in the Pipeline that Your Service Can Access	211
Logic to Process a Transaction Document	213
Logic to Process a Group Document when <i>splitOption</i> is Group	215

Chapter 14. Processing Inbound TRADACOMS Documents using Trading Networks	219
Overview	220
Before You Can Process Inbound EDI Documents	221
Specifying EDITPA Variables that Affect Inbound Processing	222
Trading Networks Attributes and EDI Documents	222
System Attributes that the EDI Module Sets	223
DocumentID	223
GroupID	223
Custom Attributes that the EDI Module Sets	224
Defining Your Own Custom Attributes for EDI Documents	225
At Design Time Creating Attributes and Associating Them with TN Document Types	225
Setting the Values of Attributes at Run Time	225
Example	226
Defining a Processing Rule that Uses Your Custom Attribute as Criteria	226
Defining Processing Rules to Process Inbound EDI Documents	227
Specifying Processing Rule Criteria	228
Specifying Pre-Processing Actions	229
Specifying Processing Actions	230
Coding Services to Process File and Batch Documents	231
Information in the Pipeline that Your Service Can Access	232
Logic to Process a File Document	234
Logic to Map a File Document To an Internal-Format Document	236
Chapter 15. Optional Inbound Processing When Using Trading Networks	237
Overview	238
Performing Inbound Control Number Validation	238
How the EDI Module Validates Inbound Control Numbers	239
Actions the EDI Module Can Take for Invalid Control Numbers	240
Error & Continue	241
Process Normally	241
Reject	242
Setting Up to Perform Inbound Control Number Validation	244
For ANSI X12 or UN/EDIFACT	244
For TRADACOMS	246
Defining Processing Rules to Handle Documents with Invalid Control Numbers	247
Custom Attribute Set for Control Number Validation	247
Defining the Processing Rule	248
Reprocessing EDI Documents with Invalid Control Numbers	250
Reprocessing Documents with Duplicate Control Numbers	250
Reprocessing Documents with Out-of-Sequence Control Numbers	251
Creating a Service to Reprocess Documents with Out-of-Sequence Control Numbers	252

Automatically Generating Functional Acknowledgements	258
Setting Up to Use Automatic FA Generation	258
Turning Automatic FA Generation On or Off	259
Variables that Affect How the EDI Module Generates the FA	260
Contents of the Generated Functional Acknowledgment	262
How the EDI Module Reports the FA Status	263
Child Transaction that is Not Allowed in its Group	264
Transaction or Group is for an FA	264
Syntax Error Status	264
Logical Error Status	265
Child Transaction Rejected Status	267
How the EDI Module Determines Which FA Status to Use	268
Transaction FA Status	268
Group FA Status	270
UN/EDIFACT Interchange FA Status	271
Actions the EDI Module Takes Based on FA Status	272
Defining the FA Statuses that Are Acceptable and Unacceptable	274
Steps the EDI Module Takes to Automatically Generate FAs	276
Defining Processing Rules to Process FAs	278
Defining Processing Rules for Documents With Unacceptable FA Statuses	279
Chapter 16. Forming EDI Documents to Send Outbound When Using Trading Networks 283	
Overview	284
Creating the Service to Form the EDI Document	284
Before Creating the Service to Form an EDI Document	285
Creating the Service	286
Obtaining Control Numbers for Outbound Processing (ANSI X12 and UN/EDIFACT)	290
Adding TRADACOMS Envelopes and Obtaining Control Numbers	290
Setting up Trading Networks to Process Internal-Format Document	291
Delivering the EDI Document	291
Deliver the EDI Document Directly from the Service that Forms It	292
Submitting the EDI Document to Trading Networks	293
Specifying Variables that Affect Outbound Processing (ANSI X12 and UN/EDIFACT)	293
Specifying Variables that Affect Outbound Processing (TRADACOMS)	295
Setting up Trading Networks to Deliver the EDI Document	296
Adding Logic to Service that Forms the EDI Document to Submit It to Trading Networks	298
Preparing the Pipeline to Invoke the <code>wm.tn.doc.xml:routeXml</code> Service	298
Routing the Outbound EDI Document to Trading Networks	301
Setting up Trading Networks to Deliver the EDI Document	302
Adding Logic to Service that Forms the EDI Document to Route It through Trading Networks	302

Creating a BizDocEnvelope for the Outbound EDI Document	303
Input Variables for the createEDIBizDoc Service	303
Output Variable for the createEDIBizDoc Service	304
Logic for the Service to Create a BizDocEnvelope for the Outbound EDI Document ..	304
Chapter 17. Batching EDI Documents	309
Overview	310
Indicating How Many Batch EDI Documents to Create	310
How Documents are Combined to Create the Batch EDI Document	311
Extracting Transactions and Sorting the Batched Documents	311
Sorting Transactions in the Queued Documents	312
Collection Areas (which represent an interchange in the final batch EDI document)	312
Subcollections (which represent a group in the final batch EDI document)	312
Delimiters Associated with Collection Areas	313
Example of Sorting Documents	314
When EDI Documents Have No Interchange or Transmission Header	315
Sorting Transactions When Queued Documents Have No Interchange Headers	315
Default Collection Area (which will be an interchange in the final batch EDI document)	316
Subcollections When There are No Headers (which will be groups in the final batch EDI document)	316
Delimiters Associated with the Default Collection Area	317
Delimiters Used for the Batch EDI Document	318
Updating the Task Status and Publishing Documents in the Case of Failure	319
Recombining the Transactions into the Batch EDI Document	320
Creating the Batch EDI Document when using SINGLEOUTPUT	322
Creating the Batch EDI Document when using MULTIPLEOUTPUTS	324
Before You Can Batch EDI Documents	326
Controlling How the batchProcess Service Forms the Batch Document	326
EDITPA Variables that the batchProcess Service Uses	327
EDITPA Variables used When Sorting Transaction	327
EDITPA Variables used When Recombining Transactions	328
Setting Up to Batch EDI Documents	329
Determining the Number of Queues You Need to Define	329
Defining the Scheduled Delivery Queues	329
Defining Processing Rules to Batch EDI Documents	331
Delivering the Batch EDI Document	332

Chapter 18. Retrieving and Delivering EDI Documents from and to VANs	335
Overview	336
Setting Up to Retrieve Documents from a VAN	336
Setting Up to Deliver Documents to VANs	337
Defining the Scheduled Delivery Queue	337
Defining a Processing Rule to Place Documents in the Queue	340
Handling Conventions Required by Specific VANs	340
Customizing the Built-in Services to Connect To Another VAN	341
Chapter 19. Handling Large Documents When Using Trading Networks	343
Configuring Large Document Handling when Using webMethods Trading Networks	344
Converting Documents to IData Objects Iteratively	344
Chapter 20. Including Documents in a Business Process	345
Overview	346
Designing the Process Model	346
Sample Process Models for ANSI X12	347
Sample Process Models for UN/EDIFACT	350
How to Assign Conversation IDs to EDI Documents	352
Instance IDs for Interchange, Group, and Transaction Documents	353
Defining an Instance ID Query for a Transaction	354
Creating an IS Document Type for a Transaction	355
Instance IDs for Functional Acknowledgments	356
Sample Business Process	357
Monitoring Your Business Processes	357
Part III. Managing and Analyzing Information about EDI Documents	359
Chapter 21. Reconciling Functional Acknowledgments	361
Overview	362
FA Reconciliation	362
FA Reports	363
Turning FA Reconciliation On	363
Information the EDI Module Records to Reconcile FAs	363
FA Statuses	365
Controlling FA Status for Documents Submitted Multiple Times	366
Creating FA Reconciliation Reports	368
Chapter 22. Viewing Log Information for EDI Documents	373
Overview	374
Viewing Server Log	374
Viewing the Trading Networks Activity Log	374

Chapter 23. Viewing Information About EDI Documents	375
Overview	376
EDI Documents You Can View	376
Viewing EDI Documents Using the Trading Networks Console	377
Trading Networks Processing Status and EDI Documents	378
Viewing Information about EDI Documents in Queues	379
Viewing Related Documents With Relationship Labels	380
Appendix A. Non-Standard Processing	385
Overview	386
Defining Partner Information	386
Defining Trading Networks Profiles	387
Defining a Partner-Specific EDITPA	388
wm.b2b.editn.TPA:EDITPA IS Document Type	388
Defining Interchange-Level Sender/Receiver Pair Information	390
Editing and Deleting Interchange-Level Sender/Receiver Pair Information	403
Defining Group-Level Sender/Receiver Pair Associations	404
Querying for Group-Level Sender/Receiver Pair Associations	407
Defining Control Number Information for Partners	408
Turning Inbound Control Number Validation On or Off	408
Defining Actions for Invalid Control Numbers	408
Defining the Control Number Settings	409
Processing Inbound EDI Documents using Trading Networks	410
Before You Can Process Inbound EDI Documents	410
Variables that Affect Inbound Processing	411
Forming EDI Documents to Send Outbound When using Trading Networks	412
Before Creating the Service to Form an EDI Document	412
Creating the Service When Using Non-Standard Processing	413
Appendix B. Using the 6.0.1 Version of the Batching Feature	417
Overview	418
The EDI Module Version 6.0.1 Batching Feature	418
Batching the Documents in a Single Pass Through Trading Networks	419
How the Documents are Combined When Using a Single Pass	420
Batching the Documents in Two Passes Through Trading Networks	421
How the Documents are Combined When Using Two Passes	424
Delivering the Batch Document	426
Before You Can Batch EDI Documents	426
Controlling How the batchProcess Service Forms the Batch Document	427
Input Variables to the batchProcess Service	427
EDITPA Variables that the batchProcess Service Uses	429

Setting Up to Batch EDI Documents	430
Determining the Number of Queues You Need to Define	431
Defining the Scheduled Delivery Queues	432
Defining Processing Rules to Batch EDI Documents	433
The EDI Batch Attribute and Processing Rules	436
Delivering the Batch EDI Document	437
Index	439

About This Guide

This guide is for users of the webMethods EDI Module. It describes how to use the webMethods EDI Module to process EDI documents. It describes: 1) how to use the EDI Module when you use it without other webMethods components. This includes how to configure the EDI Module and create services to process inbound and outbound EDI documents. And, 2) how to use the EDI Module with Trading Networks and other webMethods components. This includes how to install TN document types, define processing rules, and create services to process EDI documents.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
<i>Italic</i>	Identifies variable information that you must supply or change based on your specific situation or environment. Identifies terms the first time they are defined in text. Also identifies service input and output variables.
Narrow font	Identifies storage locations for services on the webMethods Integration Server using the convention <i>folder.subfolder:service</i> .
Typewriter font	Identifies characters and values that you must type exactly or messages that the system displays on the console.
UPPERCASE	Identifies keyboard keys. Keys that you must press simultaneously are joined with the "+" symbol.
\	Directory paths use the "\ " directory delimiter unless the subject is UNIX-specific.
[]	Optional keywords or values are enclosed in []. Do not type the [] symbols in your own code.

Additional Information

The webMethods Advantage Web site at <http://advantage.webmethods.com> provides you with important sources of information about webMethods components:

- **Troubleshooting Information.** webMethods provides troubleshooting information for many webMethods components in the [webMethods Knowledge Base](#).
- **Documentation Feedback.** To provide documentation feedback to webMethods, go to the [Documentation Feedback Form](#) on the [webMethods Bookshelf](#).
- **Additional Documentation.** All webMethods documentation is available on the [webMethods Bookshelf](#).

Using EDI Module without Other webMethods Components

■ Before You Can Process EDI Documents	19
■ Creating Clients that Send EDI Documents to the Integration Server	31
■ Receiving and Processing Inbound EDI Documents	39
■ Generating Functional Acknowledgments	55
■ Mapping Data to Form New Documents	71
■ Forming EDI Documents and Sending Them Outbound	87
■ Handling Large Documents	93

Before You Can Process EDI Documents

■ Overview	20
■ Creating Flat File Schemas for EDI Documents	20
■ Creating Flat File Schemas for Internal-Format Documents	24
■ Configuring How the Format Services Convert Field Values	25
■ Associating the EDI Format Services to EDI Data Types	29

Overview

Before you can process EDI documents, ensure you have set up the items required for parsing and validating EDI documents and converting documents from EDI format to the format required by your internal applications and vice versa.

To perform these functions, the EDI Module uses flat file schemas. You create flat file schemas that define the structure of EDI documents, and you can also create flat file schemas that define the structure of your internal format documents. You associate format services with your flat file schemas that are invoked to convert field values in a document from EDI format to the internal format, and vice versa.

- Create flat file schemas for EDI documents, as described in “[Creating Flat File Schemas for EDI Documents](#)” on page 20
- If you will want to use EDI built-in services to convert internal-format documents (e.g., documents from a back-end system) from IData objects to Strings and vice versa, create a flat file schema that defines the structure of the internal-format documents. For more information, see “[Creating Flat File Schemas for Internal-Format Documents](#)” on page 24.
- Configure how you want format services to convert field values in documents from EDI format to internal format, and vice versa. For more information, see “[Configuring How the Format Services Convert Field Values](#)” on page 25.
- Specify how to associate format service to fields defined in a flat file schema for an EDI document, as described in “[Associating the EDI Format Services to EDI Data Types](#)” on page 29.

Creating Flat File Schemas for EDI Documents

You may need to create a flat file schema for each type of EDI transaction you want to process. You can create an EDI flat file schema from either:

- Standard Exchange Format (SEF) file
- IDOC (for SAP system users)

You should create flat file schemas manually *only* if:

- You are *not* using the webMethods EDI Module in conjunction with webMethods Trading Networks
 - AND—
- And if you are processing ANSI X12 or UN/EDIFACT documents (or documents of any supported sub-standard)

You do *not* have to manually create flat file schemas from a SEF file or from an IDOC if either of the following cases applies to you:

- If you are using the webMethods EDI Module in conjunction with webMethods Trading Networks. The flat file schemas are created automatically for you when you use the procedure described in “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104 in Chapter 8, “Before You Can Use Trading Networks to Process EDI Documents”.



Important! If you are processing documents of the TRADACOMS EDI standard, you must use the webMethods EDI Module in conjunction with webMethods Trading Networks.

- If you are processing non-standard EDI formats, or if you are processing a user-defined format (such as a non-EDI flat file), you *must* create flat file schemas from the webMethods Developer using the functionality in the WmFlatFile package instead of using the one of the methods described in “[Creating Flat File Schemas for EDI Documents](#)” on page 20. For more information, see the *Flat File Schema Developer’s Guide*.

Creating a Flat File Schema from a SEF File

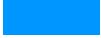
The EDI Module provides the `wm.b2b.edi:SEFParse` service to create flat file schemas from a SEF file. This service extracts information about the EDI document’s structure, separators, and segments from the SEF file and transfers the information to a flat file schema.

If you want to have the `wm.b2b.edi:SEFParse` service automatically assign format services to fields in the flat file template, configure the EDI format services that you want to associate with the different data types before you create a flat file schema. For instructions, see “[Associating the EDI Format Services to EDI Data Types](#)” on page 29.

For a description of this service, see the *webMethods EDI Module Built-In Services Reference*.



Note: If you are using the webMethods EDI Module in conjunction with webMethods Trading Networks, you do *not* need to use the `wm.b2b.edi:SEFParse` service to create the flat file schema. The flat file schemas are created automatically for you when you use the procedure described in “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104 in Chapter 8, “Before You Can Use Trading Networks to Process EDI Documents”.

 To create a flat file schema from a SEF file

- 1 Determine where the SEF file you want to use is located, either on your local file system or on the Web.

Many SEF files are provided with the EDI Module. When you install the EDI Module, the installer stores each SEF file as:

`WmEDI/pub/SEFs/standard/version.sef`, where:

- *standard* represents the EDI standard (for example, `x12`)
- *version* represents the EDI standard version (for example, `4010`).

For example: `WmEDI/pub/SEFs/X12/4010.sef`

- 2 Create the flat file dictionary for the EDI standard and version that you are using.
- 3 Invoke the `wm.b2b.edi:SEFParse` service from the webMethods Developer. The inputs to the service include the location of the SEF file you want to use, the EDI transaction set (or TRADACOMS File) name, and the package name and namespace name in which you want the service to create the schema.

The input variable *targetSchema* specifies the name you want to give the flat file schema being created.

- For all supported EDI standards except TRADACOMS, use the following naming convention for *targetSchema* if you plan to use Trading Networks:

`EDIFFSchema.standard.Vversion:Tname`, where:

- *standard* represents the EDI standard (for example, `x12`)
- *version* represents the EDI standard version (for example, `4010`)
- *name* represents the EDI transaction (for example, `850`)

For example: `EDIFFSchema.X12.V4010:T850`

- For the TRADACOMS EDI standard, the `wm.b2b.edi:SEFParse` service creates a temporary flat file schema. Use the following naming convention for *targetSchema*:

`EDIFFSchema.TRADACOMS.Vversion.Tname:TEMP_SCHEMA`, where:

- *version* represents the version of the TRADACOMS File document type (for example, `v2`)
- *name* represents the name of the TRADACOMS File document type (for example, `ORDHDR`)

This temporary flat file schema contains all the messages contained in the TRADACOMS file. You should then execute the `wm.b2b.edi.tradacoms.ui:modifyTradacomsSchema` service to split the flat file schema into one flat file schema per MHD segment in the TRADACOMS file.

This temporary flat file schema will be deleted upon successful execution of the `wm.b2b.edi.tradacoms.ui:modifyTradacomsSchema` service.

For descriptions of all the input variables of these services, see the *webMethods EDI Module Built-In Services Reference*.

- 4 From the webMethods Developer, make changes to the flat file schema to meet your site's needs. For example, you might want to change some of the format services that are associate with fields or if you do not want a format service executed for a field, remove the format service for that field.

Creating a Flat File Schema from an IDOC

The EDI Module provides the `wm.b2b.edi:createIDOCtemplate` service to create flat file schemas from an SAP IDOC. This service queries the SAP system directly for the IDOC that you want to use to create the schema.

For a description of this service, see the *webMethods EDI Module Built-In Services Reference*.

To create a flat file schema from an IDOC

- 1 Invoke the `wm.b2b.edi:createIDOCtemplate` service (e.g., from the webMethods Developer). The inputs to the service include the SAP server name, the IDOC name, the IDOC version, and the package name and namespace name in which you want the service to create the schema.

For descriptions of all the input variables, see the descriptions of the `wm.b2b.edi:createIDOCtemplate` service in the *webMethods EDI Module Built-In Services Reference*.

- 2 From the webMethods Developer, make changes to the flat file schema to meet your site's needs. For example, you might want to assign format services to fields in the flat file schema.

Migrating 4.x or Earlier Templates to Flat File Schemas

The templates that were used for webMethods EDI Module 4.x or earlier will not work in webMethods 6.0 or higher. Users of 4.x or earlier versions can either create flat file schemas as described above, or migrate the existing templates using the Migration utility on the WmEDI Home Page. webMethods recommends that you create the flat file schemas for EDI transactions rather than use migration.

Migrated flat file schemas do *not* provide the same functionality of those created using the above methods; however, they do provide the same level of functionality that was available in the EDI Module 4.x or earlier.

- EDI flat file schemas created from SEF files and IDOCs allow for parsing as well as validation of document structure and content based on a given flat file schema.
- EDI flat file schemas created via migration from 4.x templates do not contain sufficient information to validate the structure or content of the document.

You can migrate your templates and then modify the flat file schemas to perform validation, or you can use your existing namespace records to perform validation, but webMethods recommends that you create new flat file schemas.

Additionally, you must alter all places in which the template was referenced to point to the namespace name of the generated flat file schema. All 4.x or earlier flow services that invoke the `wm.b2b.edi.migration:getTemplate` service and then the `wm.b2b.edi:convertToValues` service should continue to function successfully after running the migration utility. However, 4.x or earlier flow services that have templates built into the flow-templates stored in the pipeline (with blue arrow) will no longer work. The same is true of the `wm.b2b.edi:convertToString` service. Data flows from old templates that reside on the file system under the package/pub directory to the new EDI flat file schemas that will live in the server namespace (also on the file system, under the package/ns directory).

To migrate 4.x or Earlier Templates to Flat File Schemas

For steps to migrate existing templates, see the EDI Module upgrade procedures, available on Bookshelf area of the webMethods Advantage Web site at <http://advantage.webmethods.com>.

Creating Flat File Schemas for Internal-Format Documents

If you will want to use EDI built-in services to convert internal-format documents (e.g., documents from a back-end system) from `IData` objects to Strings and vice versa, create a flat file schema that defines the structure of the internal-format documents.

To create flat file schemas for internal-format documents

Create the flat file schemas from the webMethods Developer using the functionality in the `WmFlatFile` package. For more information, see the *Flat File Schema Developer's Guide*.



Note: For backward compatibility, you can also use an IS document type to define the structure of internal-format documents. However, it is recommended that you use flat file schemas.

Configuring How the Format Services Convert Field Values

You associate format services with specific fields in a flat file schema. A format service formats the value for a field and optionally ensures that the value meets the restrictions specified in the format service. The EDI Module provides format services to convert field values from the format required by an EDI standard to the format required by your internal application (e.g., a back-end system), and vice versa. For example, a format service can alter the format of a date field from YYYYMMDD to MMDDYYYY.

The EDI format services are automatically invoked for fields within a document when you invoke one of the following services and specify a flat file schema as input:

- The `wm.b2b.edi:convertToValues` service (or the `wm.b2b.edi.tradacoms:convertToValues` service) to convert an EDI transaction set document to an IS document (`IData` object), which contains the internal document format.
- The `wm.b2b.edi:convertToString` service (or the `wm.b2b.edi.tradacoms:convertToString` service) to convert an IS document (`IData` object), which contains the internal document format, to an EDI document.



Important! If a field does not have a value (that is, a value is not returned in the IS document (`IData` object) for the `convertToValues` service or is not present in the input data for the `convertToString` service) the format service assigned to that field will not be executed.

For more information about the `convertToValues` and `convertToString` services, see the *webMethods EDI Module Built-In Services Reference*. For more information about how to use the `convertToValues` and `convertToString` services when processing EDI documents, see “[Logic to Include in the Service to Process EDI Documents](#)” on page 41 and “[Logic to Include in the Service to Form an EDI Document](#)” on page 89.

To determine how to do the field conversions, the format services use the settings in the following configuration file:

`webMethods6\IntegrationServer\packages\WmEDI\config\format.xml`

The `format.xml` configuration file specifies the format required by the EDI standard (external format) and the format required by your internal application (internal format). As installed, the external formats in the `format.xml` configuration file match the EDI ANSI standard format. You need to update the configuration so that it reflects the correct formats for the standard of EDI that you are using and the correct formats for the internal application format that you will want to convert field values to and from.

To configure how to convert field values in documents, perform the following procedure to update the configuration in memory (so the changes take effect immediately) and in the `format.xml` file.

To configure how to convert field values in documents

- 1 Open the Server Administrator if it is not already open.
- 2 In the Solutions menu of the navigation panel, click EDI. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the Configuration menu of the navigation panel, click Configure Field Formats.
- 4 Click Edit Format Settings.
- 5 Update each entry in the `format.xml` configuration file to meet your site's needs. Do not add any additional `FormatService` entries.

Each entry has the following format:

Format	Example
<pre><FormatService name="name_of_format_service" isEnabled="true_or_false" internalFormatString="format" externalFormatString="format" applyFormat="true_or_false"/></pre>	<pre><FormatService name="formatDate6" isEnabled="true" internalFormatString="MM/dd/yy" externalFormatString="yyMMdd" applyFormat="true"/></pre>

The following describes the meaning of each setting in an entry:

Setting	Description	
	Format Service	Description
name		The name of the format service that the entry governs. Do not change <code>name</code> . The following lists the EDI format services. For more information about these services, see the <i>webMethods EDI Module Built-In Services Reference</i> .
	formatDate6	Formats fields with EDI data type <code>DT</code> that have a minimum length of 6 and a maximum length of 6.
	formatDate8	Formats fields with EDI data type <code>DT</code> that have a minimum length of 8 and a maximum length of 8.
	formatDecimal	Formats fields with EDI data type <code>D</code> .
	formatImplied Decimal	This entry affects all implied decimal services (<code>formatN0-formatN9</code>). These services format fields with EDI data types <code>N0 - N9</code> .

Setting	Description
formatTime4_4	Formats fields with EDI data type TM with a minimum length of 4 and a maximum length of 4.
formatTime4_6	Formats fields with EDI data type TM with a minimum length of 4 and a maximum length of 6.
formatTime4_8	Formats fields with EDI data type TM with a minimum length of 4 and a maximum length of 8.
formatTime6_6	Formats fields with EDI data type TM with a minimum length of 6 and a maximum length of 6.
isEnabled	<p>Whether the format service is enabled. Specify either true or false.</p> <ul style="list-style-type: none"> ■ If you specify true, format service performs its processing to convert and/or validate field values. ■ If you specify false, the format service immediately returns without performing any conversion or validation.

Note: You can also configure whether to enable format services in the configuration of the WmFlatFile.

Note: If the configuration for the WmFlatFile package specifies not to use format services, the convertToValues and convertToString services will not invoke format service, even if the isEnabled configuration setting is set to true. For more information about how to configure the WmFlatFile package, see the *Flat File Schema Developer's Guide*.

<u>Setting</u>	<u>Description</u>
internalFormatString	<p>The format required by your internal application, e.g., your back-end system.</p> <p>Specify a String that follows the conventions described in the java class <code>java.text.DecimalFormat</code> (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html) and the java class <code>java.text.SimpleDateFormat</code> (http://java.sun.com/products/jdk/1.2/docs/api/java/text/SimpleDateFormat.html).</p>
externalFormatString	<p>The format required by the EDI standard you are using. If you are using the EDI ANSI standard format, you should not change this setting.</p> <p>Specify a String that follows the conventions described in the java class <code>java.text.DecimalFormat</code> (http://java.sun.com/products/jdk/1.2/docs/api/java/text/DecimalFormat.html) and the java class <code>java.text.SimpleDateFormat</code> (http://java.sun.com/products/jdk/1.2/docs/api/java/text/SimpleDateFormat.html).</p>
applyFormat	<p>Whether the <code>convertToValues</code> or <code>convertToString</code> services should apply the converted value. Specify either <code>true</code> or <code>false</code>.</p> <ul style="list-style-type: none">■ If you specify <code>true</code>, if you want to validate and update the document to reflect the converted value.■ If you specify <code>false</code>, if you want to validate only and do <i>not</i> want the document to reflect the converted value.

6 Click Save Changes.

For information about format services and the `WmFlatFile` package, including error information and how to specify format services for fields in flat file schemas and dictionaries, see “Format Services” in the “Creating and Editing Flat File Schemas and Dictionaries” chapter of the *Flat File Schema Developer’s Guide*.

Associating the EDI Format Services to EDI Data Types

When you create a flat file schema from a SEF file, you can have the `wm.b2b.edi:SEFParse` service automatically assign format services to the fields in the flat file schema. For the `wm.b2b.edi:SEFParse` service to be able to assign the format services, you must associate EDI format services with EDI data types. The `wm.b2b.edi:SEFParse` service assigns the format services to fields based on the fields' data types. For example, if a field has data type "DT", the `wm.b2b.edi:SEFParse` service assigns the format service that you associate with the DT data type to that field. For more information about how to create flat files from SEF files, see "[Creating a Flat File Schema from a SEF File](#)" on page 21.

To determine the data type of a field, the `wm.b2b.edi:SEFParse` service uses the element definition contained within the SEF file. The element definition has the following structure:

```
<element name>=<data type>,<min length>,<max length>
```

To determine the format service to associate with a specific data type, the `wm.b2b.edi:SEFParse` service uses information that you configure in the following configuration file:

```
webMethods6\IntegrationServer\packages\WmEDI\config\format.xml
```

In the `format.xml` configuration file, you configure the format service you want the `wm.b2b.edi:SEFParse` service to use for each EDI data type.

To associate EDI format services with EDI data types, perform the following procedure to update the configuration in memory (so the changes take effect immediately) and in the `format.xml` file.

To associate EDI format services with EDI data types

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **Configuration** menu of the navigation panel, click **Configure Field Formats**.
- 4 Click **Edit Format Settings**.
- 5 Update each entry in the `format.xml` configuration file to meet your site's needs. Each entry has the following format structure:

```
<DatatypeFormat  
datatype="data_type"  
service="format_service_to_associate_with_datatype"/>
```

For more information about the format services that are provided with the EDI Module, see the description of the `wm.b2b.edi.util.formatServices` in the *webMethods EDI Module Built-In Services Reference*.

Examples:

- The following entry associates the data type N1 with the formatN1 format service. This entry matches all elements with data type N1, regardless of their length.

```
<DatatypeFormat  
datatype="N1"  
service="wm.b2b.edi.util.formatServices:formatN1"/>
```

- The following entry associates the data type TM that has a minimum length of 4 and a maximum length of 8 with the formatTime4_8 format service:

```
<DatatypeFormat  
datatype="TM, 4, 8"  
service="wm.b2b.edi.util.formatServices:formatTime4_8"/>
```

6 Re-order of the DatatypeFormat entries, if necessary.

The `wm.b2b.edi:SEFParse` service searches the `DatatypeFormat` entries in the order they appear in the `format.xml` configuration file and uses the first match. As a result, ensure your entries are listed from the more specific to the less specific.

Example

To use format service “example:service1” for data type N1 where the minimum length is 1 and maximum length is 5 and to use format service “example:service2” for all other N1 data types, the `format.xml` file would contain `DatatypeFormat` entries in the following order:

```
<DatatypeFormat  
datatype="N1, 1, 5"  
service="example:service1"/>  
<DatatypeFormat  
datatype="N1"  
service="example:service2"/>
```

7 Click Save Changes.

Creating Clients that Send EDI Documents to the Integration Server

■ Overview	32
■ Content Type to Use	32
■ Service to Invoke to Handle the EDI Document	33
■ Sending EDI Documents to the Integration Server via HTTP	33
■ Sending EDI Documents to the Integration Server via FTP	35
■ Sending EDI Documents to the Integration Server via File Polling	37

Overview

To send EDI documents to the Integration Server for processing, create a client to the Integration Server. For more information about creating clients, see the *webMethods Developer User's Guide*. The client you create must:

- Send the document to the Integration Server using a content type that the EDI Module handles.
- Invoke a service on the Integration Server that you created to process the EDI document.
- Use HTTP, HTTPS, FTP, or File Polling to transport the document to the Integration Server. If you want to use EDIINT to transport the document, you must use webMethods EDIINT Module and webMethods Trading Networks. For more information, see the *webMethods EDIINT Module User's Guide*.

To learn more about the use of clients for inbound EDI document processing, see Chapter 2, "Using the EDI Module without Other webMethods Components", of the *webMethods EDI Module Concepts Guide*.

Content Type to Use

The client you create should send the EDI document to the Integration Server using the content type, `application/EDIstream`.

When you use `application/EDIstream`, the Integration Server passes the document to the EDI Module content handler as an `InputStream`. The EDI Module content handler forms the pipeline with the variable `edidata` and assigns this variable the pointer of the `InputStream`. The content handler then invokes the service that the client specifies.



Note: For backward compatibility, the EDI Module also has content handlers to accept documents with the content types `application/EDI`, `application/X12`, and `application/UNEDIFACT`. With these content types, the EDI Module content handler must convert the document to a String that it places in the pipeline. This can potentially consume a lot of pipeline space and use a significant amount of memory. As a result, it is recommended that you use the content type, `application/EDIstream`, because it conserves system memory.

Service to Invoke to Handle the EDI Document

You create a service that processes the EDI document. For more information about creating this service, see [Chapter 3, “Receiving and Processing Inbound EDI Documents”](#).

How the client identifies the service to invoke depends on the transport you want to use to send the EDI document to the Integration Server. See the sections below for how the client identifies the service to invoke.

Sending EDI Documents to the Integration Server via HTTP

You can create a client that uses HTTP or HTTPS to post an EDI document to the Integration Server. The following table lists requirements for the HTTP client.

Requirements
<ul style="list-style-type: none">■ The client must be able to send the EDI document as a String of data to the Integration Server using the HTTP POST method.■ The client must be able to set the value of the content type (e.g., <code>application/EDISTream</code>) in the request-header field.

Because most browsers do not allow you to modify the content type header field, they are not suitable clients for this type of submission. Clients that you might use to submit an EDI document in this manner include PERL scripts (which allow you to build and issue HTTP requests) and the webMethods pub.client:http service.

Identifying the Service to Invoke

When using HTTP, the client identifies the service to invoke by specifying the URL of that service as the HTTP request URL. For example, if the fully-qualified name of the service you want to invoke is `myEDIServices.v1:processEDIDoc`, use the following URL:

`http://rubicon:5555/Invoke/myEDIServices.v1/processEDIDoc`

Logic to Include in the HTTP Client

When using HTTP, the client must include the following logic:

- Submit a POST request to the Integration Server.
- Address the request to the URL of the service that is to process the EDI document (e.g., `http://rubicon:5555/Invoke/myEDIServices.v1/processEDIDoc`).

- Set the content type you want to `application/EDISTream` in the HTTP request header.
- Put the EDI document to process in the body of the message. The document must be the only text that appears in the body of the request.



Important! Do not include manual line breaks at the end of the EDI document, as you might with an XML document.

Example of Input Variables for the pub.client:http Service

The following example describes the values that you would set if you used the `webMethods pub.client:http` service to POST an EDI document to a service with the fully-qualified name `myEDIServices.v1:processEDIDoc`. For a complete description of this service, see the *webMethods Integration Server Built-In Services Reference Guide*.

<u>Set this Variable...</u>	<u>Data type</u>	<u>To...</u>						
<code>url</code>	String	The URL of the service that you want to invoke to process the EDI document. The following value would invoke the <code>processEDIDoc</code> service located in the <code>myEDIServices.v1</code> folder on the “rubicon” server with port number “5555.” Example: <code>http://rubicon:5555/invoke/myEDIServices.v1/processEDIDoc</code>						
<code>method</code>	String	<code>post</code>						
<code>loadAs</code>	String	The data type of the input data source. Specify either <code>bytes</code> or <code>stream</code> . <ul style="list-style-type: none">■ <code>bytes</code> if the document data source is a <code>byte[]</code>.■ <code>stream</code> if the document data source is an <code>InputStream</code>						
<code>data/string</code>	String	The EDI document that you want to post.						
<code>headers</code>	Document	An <code>IData</code> object that contains the following: <table border="1"><thead><tr><th><u>Variable</u></th><th><u>Value</u></th></tr></thead><tbody><tr><td><code>Name</code></td><td><code>Content-type</code></td></tr><tr><td><code>Value</code></td><td>The content type for the document, for example, <code>application/EDISTream</code></td></tr></tbody></table>	<u>Variable</u>	<u>Value</u>	<code>Name</code>	<code>Content-type</code>	<code>Value</code>	The content type for the document, for example, <code>application/EDISTream</code>
<u>Variable</u>	<u>Value</u>							
<code>Name</code>	<code>Content-type</code>							
<code>Value</code>	The content type for the document, for example, <code>application/EDISTream</code>							

Also, the client can set any optional HTTP variables, such as authorization information, that are required by your application.

Sending EDI Documents to the Integration Server via FTP

You can create a client that FTPs an EDI document to the Integration Server's FTP listening port. By default the FTP port is assigned to port 8021. However, this assignment is configurable, so you should check with your server administrator to see which port is used for FTP communications on your Integration Server.

Identifying the Service to Invoke

When using FTP, the client identifies the service to invoke by changing to the directory that represents the service you want to invoke. For example, if the fully-qualified name of the service you want to invoke is myEDIServices.v1:processEDIDoc, the client would issue the `cd` command to change to the following directory:

```
\ns\myEDIServices\v1\processEDIDoc
```

Logic to Include in the FTP Client

When using FTP, the client must include the following logic:

- 1 Initiate an FTP connection to the Integration Server's FTP listening port (for example, 8021).
- 2 Change to the directory that represents the service you want to invoke using the `cd` command.

Example: `cd \ns\myEDIServices\v1\processEDIDoc`



Note: The root directory for this operation is your Integration Server's *namespace* directory (`ns`), not the root directory of the target machine. Therefore, if you want to FTP a file to a service in the `myEDIServices.v1` folder, you use `\ns\myEDIServices\v1\ServiceName` as the path to that service, not `\webMethods\Server\packages\myEDIServices\v1\ServiceName`.

- 3 Send the EDI document to this directory using the following `put` command:

`put localFileName filename;content type:content sub-type`

Example: `put x12_850 x12_850;application:EDIstream`

In this example, the local EDI document's name is `x12_850`, the document's content type is `application`, and the document's content sub-type is `EDIstream`.

Example of Input Variables for the pub.client:ftp Service

The following example describes the values that you would set if you used the webMethods pub.client:ftp service to FTP an EDI document to a service with the fully-qualified name myEDIServices.v1:processEDIDoc. For a complete description of this service, see the *webMethods Integration Server Built-In Services Reference Guide*.

Set this Variable...	Data type	To...
<i>serverhost</i>	String	Name of the machine running the Integration Server.
<i>serverport</i>	String	Port on which the Integration Server listens for FTP requests.
<i>username</i>	String	A valid user name of an Integration Server user account.
<i>password</i>	String	The password for the user name.
<i>command</i>	String	put
<i>dirpath</i>	String	The path representing the service to invoke, e.g., \ns\myEDIServices\v1\processEDIDoc
<i>localfile</i>	String	Name of the source file containing the EDI document, e.g., x12_850
<i>remotefile</i>	String	Name to assign the file on the Integration Server and the content type. Use the following format: <i>filename;content type:content sub-type</i> For example: x12_850;application:EDIstream
<i>secure</i>	Document	Indicates whether the FTP session is with a secure FTP server. The variable <i>auth</i> specifies the kind of authentication mechanism to use (SSL, TLS, or TLS-P), and the variable <i>securedata</i> specifies whether the client is sending PROT C (Data Channel Protection Level Clear) or PROT P (Data Channel Protection Level Private).

Sending EDI Documents to the Integration Server via File Polling

You can send an EDI document to the Integration Server using File Polling. When you use File Polling, you define:

- Directory that the Integration Server monitors for files
- EDI content type to associate with the files placed in that directory
- Service to execute to handle files placed in the directory

When files (e.g., EDI documents) are placed in this directory (e.g., sent via FTP into the directory), the Integration Server uses the content type you specify. Because the content type is an EDI content type, when the Integration Server receives the EDI document in the monitored directory, it passes the document to the appropriate EDI Module content handler. The content handler parses the document body and passes it in the *edidata* pipeline variable to the specified service.

Setting Up File Polling

For general steps to configure a File Polling listener port, see the “Configuring Ports” section of the “Configuring the Server” chapter in the *webMethods Integration Server Administrator’s Guide*.

When you add the File Polling listener port, set it up to poll specifically for EDI documents. To do so, specify the following:

In this field...	Specify...
Monitoring Directory	The directory on the Integration Server that the listener monitors for incoming EDI documents.
Content type	The content type that you want the Integration Server to use for documents that are placed in the Monitoring Directory. Specify a content type that the EDI Module recognizes. For more information about these content types, see “ Content Type to Use ” on page 32 in this chapter.
Processing Service	Service that is to process the EDI document.

Identifying the Service to Invoke

When you use File Polling, you identify the service that you want to process the EDI document when you add the File Polling listening port. Specifically, you identify the service in the Processing Service field.

Receiving and Processing Inbound EDI Documents

■ Overview	40
■ Before Creating the Service to Process EDI Documents	40
■ Inputs to Your Service	41
■ Logic to Include in the Service to Process EDI Documents	41
■ Examples	53

Overview

The basic EDI engine of the EDI Module (i.e., the WmEDI package) functions as a toolkit that you can use to build your own EDI solution. To create your EDI solution, you create a service that processes EDI documents to meet your site's needs. The client that sends the EDI document to the Integration Server invokes this service. For more information about how to create the client, see [Chapter 2, “Creating Clients that Send EDI Documents to the Integration Server”](#).

The WmEDI package of the EDI Module provides built-in services that provide the functionality to support the EDI standard. When you create the service that processes your EDI documents, you add logic to invoke the provided built-in services. This chapter provides information about how to create the service to process an EDI document.

To learn more about the basics about how to process an inbound EDI document, see Chapter 2, "Using the EDI Module without Other webMethods Components" in the *webMethods EDI Module Concepts Guide*.



Note: Support for the TRADACOMS standard is provided when you use the EDI Module in conjunction with webMethods Trading Networks, as described starting with [Chapter 8, “Before You Can Use Trading Networks to Process EDI Documents”](#).

Before Creating the Service to Process EDI Documents

Before you create the service to process EDI documents that are sent to your Integration Server, create:

- The flat file schema that defines the structure of the EDI document to process. The EDI Module uses the flat file schema for parsing, converting, and validating an inbound EDI document. For instructions on how to create the flat file schemas, see [“Creating Flat File Schemas for EDI Documents” on page 20](#).
- Optionally, the flat file schema that defines the structure of the internal-format document. This is the document that your service will send to your internal application (e.g., a back-end system). The EDI Module provides a service that you can have your service invoke to create your internal-format document based on the flat file schema. Use the webMethods Developer to create the flat file schema. For more information, see the *Flat File Schema Developer’s Guide*.

Inputs to Your Service

The service you create should accept as input the *edidata* variable that the EDI Module content handler placed in the pipeline. The data type of the *edidata* variable depends on the content type you use:

If you use this content type...	The data type of <i>edidata</i> is...
application/EDIstream	InputStream
application/EDI, application/X12, or application/UNEDIFACT	String

 Note: It is recommended that your client use the content type application/EDIstream. The other content types (application/EDI, application/X12, and application/UNEDIFACT) are provided for backward compatibility. For more information, see “Content Type to Use” on page 32.

Logic to Include in the Service to Process EDI Documents

You can perform any processing on the EDI document that you want. Some typical ways to process an EDI document might be to:

- Map information from the EDI document into one or more internal-format documents. The internal-format is the format that an internal application (e.g., a back-end system) requires. Then send the internal-format document to the internal application.
- Map information from the EDI document to the inputs of a service, and then invoke the service.

This chapter focuses on the first case, mapping data to the format an internal-format document. However, you can apply the principles that are presented to other processing you might want to accomplish.

Additionally, you can also generate and send a functional acknowledgments (FAs) for the EDI document. For more information, see [Chapter 4, “Generating Functional Acknowledgments”](#).

Determining the Method to Use to Process

There are two basic ways to process the document:

- Process the entire document at one time. With this method, you invoke services that consumes the entire EDI document at one time, then you can process it. Use it to process EDI documents that have smaller transaction sets.
- Process the document iteratively, segment by segment. With this method, you process the document by working on one segment at a time (or groups of segments at a time). When you finish one segment or a group of segments, you work on the next. This method is useful when working with larger transaction sets that can easily be parsed into smaller units (for example, a large 810 document containing many line item segments).

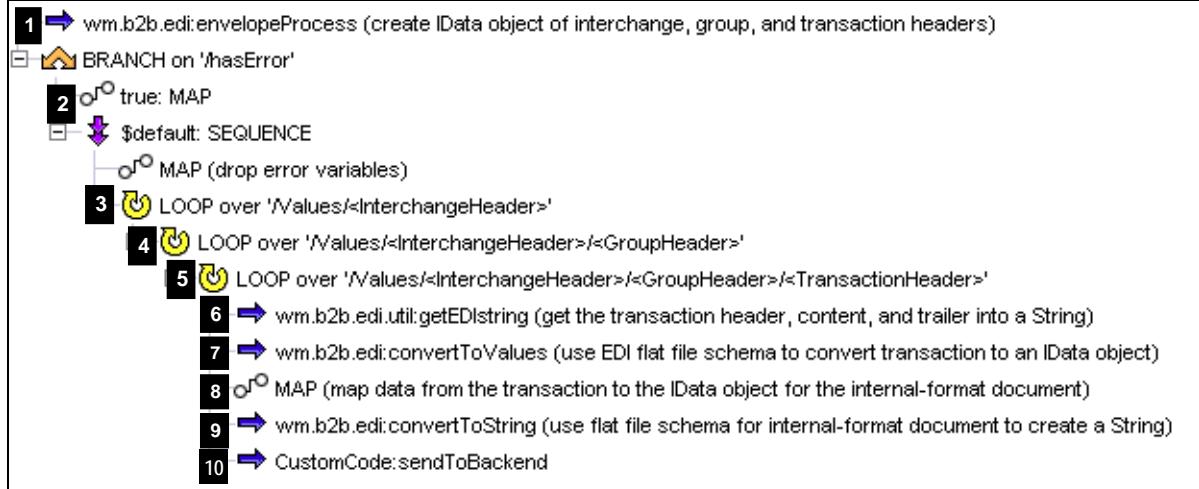
The table below lists the advantages and disadvantages of each method:

Process entire document at one time	Process document segment by segment
Advantages	Advantages
<ul style="list-style-type: none"> ■ This method is easier to implement because most of the processing and set up you need to do is provided with the EDI Module. ■ You can use built-in services to perform envelope validation and compliance checks. For more information about the validation and compliance checks, see "What Does the Envelope Validation and Compliance Check Entail?" on page 47. 	<ul style="list-style-type: none"> ■ Requires less memory because an IData object is created for only the segment or group of segments that represent a repeating content (e.g., line items) as the segments are being processed and the memory is reused when working on subsequent segments.
Disadvantages	Disadvantages
<ul style="list-style-type: none"> ■ Requires more memory because an IData object is created for the entire document. 	<ul style="list-style-type: none"> ■ If you want to perform the envelope validation and compliance checks, you must add your own logic to do so. ■ You most likely will need to modify the flat file schema for the EDI document you are processing and possibly create additional flat file schemas for segments within the EDI document.

Processing the Entire Document at One Time

The following shows sample code that includes the basic logic you would include to process an inbound EDI document. This processing shows how to map information from the EDI document into an internal-format document and send the document to an internal application, e.g., a back-end system. For details about all of the built-in services that the sample uses, see the *webMethods EDI Module Built-In Services Reference*.

Sample code for processing the entire inbound EDI document at one time



Flow operation	Description
1	<p>Invoke the <code>wm.b2b.edi:envelopeProcess</code> service to process the envelopes in the inbound EDI document. This service consumes the entire EDI document and converts all of its envelope header segments, including transaction set header segments, into an <code>IData</code> object named <i>Values</i>. That is,</p> <ul style="list-style-type: none"> ■ For an ANSI X12 document, it creates an <code>IData</code> object of the ISA/IEA, GS/GE and ST/SE headers/trailers. ■ For a UN/EDIFACT document, it creates an <code>IData</code> object of the UNB/UNZ, UNG/UNE, and UNH/UNT headers/trailers. <p>After the header segments are converted to the <i>Values</i> <code>IData</code> object, your service can access and act on data elements within the header segments of the EDI document.</p>

Flow operation	Description
	<p>The contents of the transaction sets remain unparsed. How the service handles the transaction set contents is based on whether the document is considered large. For information about how the EDI Module considers documents large or not, see Chapter 7, “Handling Large Documents”.</p> <ul style="list-style-type: none"> ■ For documents <i>not</i> considered large, the service leaves the transaction set contents in String format. The service assigns the transaction set body to the variable <i>unDefData</i> that is in the <i>Values</i> IData object under the transaction set header. For example, for an ANSI X12 document, the content would be in the elements <i>Values/ISA/GS/ST/unDefData</i>. The variable name <i>unDefData</i> refers to the fact that the content has not yet been processed. ■ For documents considered large, the service writes the transaction set contents to hard disk drive storage. When data is written to the hard disk drive storage, it is assigned a reference ID that is used to access the data. The service assigns the reference ID for the transaction set contents to the variable <i>_RID_</i>, which is in the <i>Values</i> IData object under the transaction set header. For example, for a UN/EDIFACT document that contains group envelopes, the reference ID would be in the elements <i>Values/UINB/UNG/UNH/_RID_</i>. For more information about how the EDI Module handles large documents, see Chapter 7. <p>Additionally, the envelopeProcess service can perform an optional check for basic compliance on the interchange envelope. You specify whether to perform the compliance check by setting two optional input variable for the envelopeProcess service. For more information about the compliance check, see “What Does the Envelope Validation and Compliance Check Entail?” on page 47.</p>
2	Add your own logic to handle errors that might result from executing the <code>wm.b2b.edi:envelopeProcess</code> service, for example errors with the compliance check.
3	<p>Loop through the interchange envelope headers. The data for the interchange headers is within the <i>Values</i> IData object. The sample code above shows the loop over <i>Values/<InterchangeHeader></i>. If you are processing:</p> <ul style="list-style-type: none"> ■ ANSI X12 document, the interchange headers are in <i>Values/ISA</i> ■ UN/EDIFACT document, the interchange headers are in <i>Values/UINB</i>

Flow operation	Description
4	<p>Loop through the group envelope headers. The data for the group headers is within the <i>Values</i> IData object. The sample code above shows the loop over <i>Values/<InterchangeHeader>/GroupHeader</i>. If you are processing:</p> <ul style="list-style-type: none"> ■ ANSI X12 document, the group headers are in <i>Values/ISA/GS</i> ■ UN/EDIFACT document and the document has group headers, the group headers are in <i>Values/UNB/UNG</i> <p>If you are processing an UN/EDIFACT document that might not contain group headers, your processing will need to be a little different. For sample code, see the <code>sampleServices:UNEDIFACTToValues</code> in the <code>WmEDIsamples</code> package, which is located in the Knowledge Base Samples area on the Advantage Web site at http://advantage.webmethods.com. The samples in this folder have been certified, meaning that they have been tested by webMethods.</p>
5	<p>Loop through the transaction set headers. The data for the transaction set headers is within the <i>Values</i> IData object. The sample code above shows the loop over <i>Values/<InterchangeHeader>/GroupHeader/<Transaction></i>. If you are processing:</p> <ul style="list-style-type: none"> ■ ANSI X12 document, the transaction set headers are in <i>Values/ISA/GS/ST</i> ■ UN/EDIFACT document, the transaction set headers are in <i>Values/UNB/UNG/UNH</i>
The remaining steps specify processing to perform for the content of each transaction.	
6	<p>Invoke the <code>wm.b2b.edi.util:getEDIstring</code> service to convert the transaction set header and trailer back to a String and concatenate them with the transaction set contents. The resulting transaction set with header and trailer can be either a String or InputStream.</p> <p>You need to have an element that contains the entire transaction set, including header and trailer before you can invoke the next service, <code>wm.b2b.edi:convertToValues</code>. This is because the <code>wm.b2b.edi:convertToValues</code> service uses a flat file schema for the EDI transaction set that includes the transaction set header and trailers. If you input data without the header and trailer, the <code>wm.b2b.edi:convertToValues</code> service will return errors.</p>

Flow operation	Description
7	<p>Invoke the <code>wm.b2b.edi:convertToValues</code> service to:</p> <ul style="list-style-type: none">■ Convert the EDI transaction set that is in either String or InputStream format to an IData object■ Validate the transaction EDI structure <p>The inputs to the <code>convertToValues</code> service include the output from the <code>wm.b2b.edi.util:getEDIstring</code> service (the String or InputStream) and the flat file schema for the EDI document. The <code>convertToValues</code> service uses the flat file schema to both determine how to parse the transaction set into an IData object and to validate its structure.</p>
8	<p>Map data from the EDI transaction sets into the internal-format document.</p> <p>Now that the transaction set contents is an IData object, you can access the data in the transaction set content to map it to an IData object for the internal-format document. Depending on the complexity of your mapping requirements, you might need to add more logic than a MAP flow operation, or create a separate service to perform the mapping. For more information about how to map, see Chapter 5, “Mapping Data to Form New Documents”.</p>
9	<p>Invoke the <code>wm.b2b.edi:convertToString</code> service to convert the internal-format document from an IData object to String format.</p> <p>The inputs to the <code>convertToString</code> service include the IData object that contains the data for your internal-format document and the flat file schema for the internal-format document. The <code>convertToString</code> service uses the flat file schema to determine how to form the internal-format document. Alternatively, rather than use a flat file schema, the <code>convertToString</code> service also accepts an IS document type to define the structure of the internal-format document.</p>
10	<p>Add your own logic or invoke a service that you create to send the internal-format document to your internal application, e.g., a back-end system.</p>

What Does the Envelope Validation and Compliance Check Entail?

When you invoke the `wm.b2b.edi:envelopeProcess` service, you can set the following input variables to `true` to have the service perform the associated validation or compliance check:

Set this input variable to true	To have the <code>wm.b2b.edi:envelopeProcess</code> service...
<code>validate</code>	<p>Validate the interchange envelope. For ANSI X12 and UN/EDIFACT standards this includes validating field lengths, code lists, ranges, and partitions. If the service finds any errors, it records them in the output variable <code>errorArray</code>.</p> <p>The interchange envelopes are validated against the EDI flat file schema for the EDI document. The EDI Module ships with EDI flat file schemas for ANSI X12 and UN/EDIFACT envelope validation and compliance checks. They are located in the <code>wm.b2b.edi.EDIFFSchema</code> folder.</p>
<code>complianceCheck</code>	<p>Check for matching interchange control numbers, matching group control numbers, matching transaction control numbers, segment counts, transaction counts, and group counts. If the service detects an error during the compliance check, the error service stops executing after it detects the first error.</p>

If you set both the `complianceCheck` and `validate` variables are set to `true`, the `envelopeProcess` service first performs the interchange envelope validation, then the compliance check.

Processing the Document Iteratively Segment by Segment

You can process a document segment by segment instead of all at once. You can process a single segment or a group of segments at a time. To process segment by segment, when you invoke the `wm.b2b.edi:convertToValues` service to convert data from String to an `IData` object, you specify the input variable `iterator` as `true`. Setting `iterator` to `true` causes the `convertToValues` service to process just a segment or a group of segments of the document. The `convertToValues` service determines how many segments to process based on the flat file structure information in the flat file schema for the EDI document. Because the `convertToValues` service uses the flat file schema, you will need to customize the flat file schema for the EDI document.

Customizing the EDI Flat File Schema

To customize the EDI flat file schema, edit it using the webMethods Developer. The following describes the changes you should make. For information about editing flat file schemas, see the *Flat File Schema Developer's Guide*.

 To customize the EDI flat file schema for iterative processing

- 1 If you created the EDI flat file schema from a SEF file, add the elements for the interchange envelopes and group envelopes into the flat file structure.

SEF files do not describe these elements. Your service will execute the `wm.b2b.edi:convertToValues` service against the entire EDI document rather than just a transaction set or file; as a result, the flat file schema needs to reflect the envelope structure.

- 2 Update the flat file structure to reflect how you want the `convertToValues` service to return segments.

When you execute the `convertToValues` service with the input variable `iterator` as `true`, the `convertToValues` service uses the flat file structure to determine the number of segments to process into an `IData` object. It processes starting at a top-level element and processes all its children into an `IData` object in one invocation. For example, if you want to the first invocation to return only the ISA header information, update the structure as follows:

```
ISA
  ISA01
  ISA02
  ISA03
  ISA04
  ISA05
  ISA06
  ISA07
  ISA08
  ISA09
  ISA10
  ISA11
  ISA12
  ISA13
  ISA14
  ISA15
  ISA15
GS
.
.
.
```

Because GS is not a child under ISA, the `convertToValues` service would return only the ISA entry as an `IData` object. A subsequent call would act on the GS entry.

- 3 Identify the repeating sections of the EDI document that you want to loop over to process and remove these sections from the flat file schema structure. For example, in an ANSI X12 810 document, you might want to loop over the `IT1` (line item) segments because you are interested in the line items included in the invoice.

Before Omitting Section	After Omitting Section
ST	ST
.	.
.	.
FA1	FA1
.	.
.	.
FA2	FA2
FA201	FA201
FA202	FA202
IT1	TDS
.	.
.	.
TDS	
.	
.	
.	

When you execute the convertToValues service with the updated flat file schema, when the service encounters sections of the document that are not defined in the flat file schema structure, it places those sections as unparsed in the returned IData object. The unparsed sections will be in either an element named *unDefData* or *_RID_*, depending on whether the document is considered large. For more information about large documents, see [Chapter 7, “Handling Large Documents”](#).

Sample Returned IData object

```
EDIVValues
  ST
    FA1
    FA2
  unDefData
  TDS
```

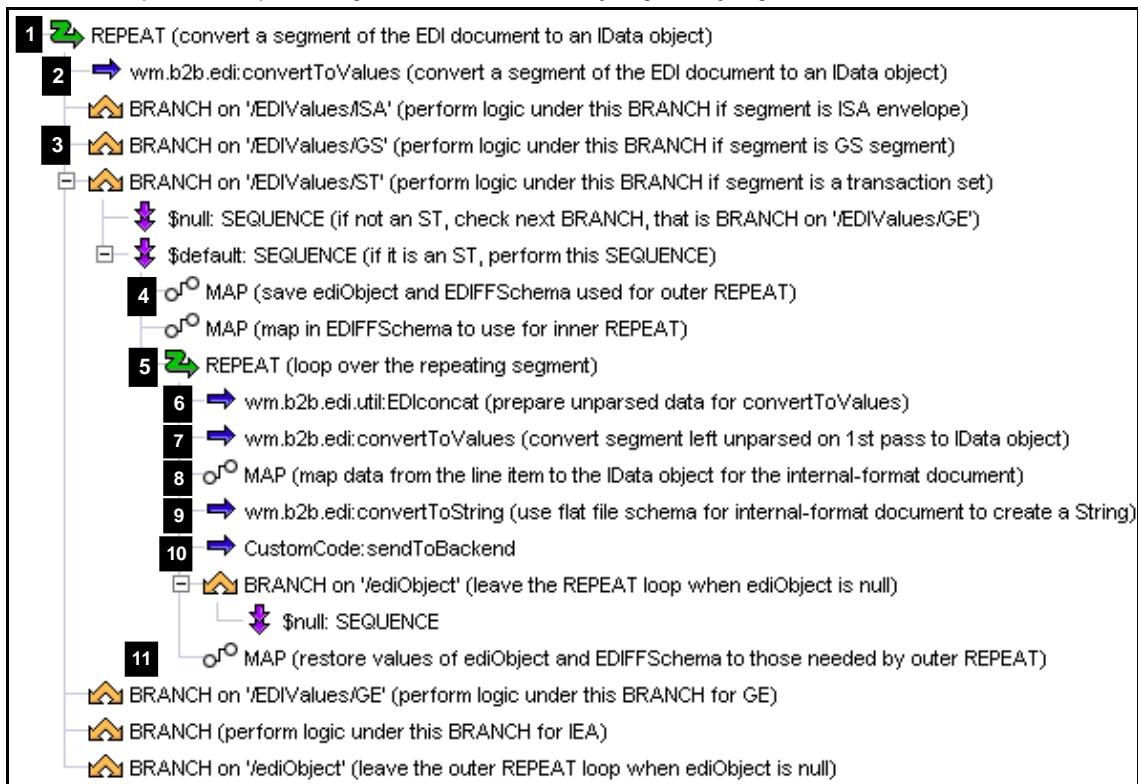
- 4 Create new flat file schemas for the sections that you removed from the EDI flat file schema.

In your service, you will perform a second pass of the convertToValues service for the unparsed sections of the document. When you invoke the convertToValues service for the second pass, the input flat file schema is this new flat file schema that you are creating that defines the structure of the data left unparsed; that is, a repeating section of the document that you want to process iteratively.

Logic for the Service to Process the Document Iteratively

This following shows sample code for processing an inbound EDI document iteratively. This processing shows how to map information from the line item segments of an ANSI X12 810 document into an internal-format document and send the document to an internal application, e.g., a back-end system. For details about all of the built-in services that the sample uses, see the *webMethods EDI Module Built-In Services Reference*.

Sample code for processing an inbound EDI iteratively, segment by segment



Flow operation	Description
1	Repeatedly invoke the <code>wm.b2b.edi:convertToValues</code> service until your service has processed the entire EDI document. When you pass the input variable <code>iterator</code> as <code>true</code> to the <code>convertToValues</code> service, the output contains the object <code>ediObject</code> that is used to keep track of the input data segments. When the entire EDI document has been processed, <code>EDIObject</code> becomes null.

Flow operation	Description
2	<p>In the outer REPEAT loop, invoke the convertToValues service to:</p> <ul style="list-style-type: none"> ■ Convert a segment or group of segments to an IData object ■ Validate the EDI structure
	<p>The inputs to the convertToValues service include the String or InputStream that represents the EDI document and the customized flat file schema for the EDI document. The convertToValues service uses the flat file schema to determine how many segments to read. It returns the object <i>ediObject</i> that keeps track of where it is in the EDI document. This object is input to subsequent invocations of the convertToValues service to ensure the service continues where it left off at the last invocation.</p>
3	<p>Add BRANCH flow operations for each type of segment you expect the convertToValues service to return.</p> <p>For example, the first BRANCH under the call to the convertToValues service is checking for an <i>ISA</i> element in the returned <i>IData</i> object, <i>EDIVValues</i>. If <i>EDIVValues/ISA</i> exists and is not null, the flow service executes the logic under the BRANCH on '<i>EDIVValues/ISA</i>' flow operation.</p>
4	<p>Prepare for invoking the convertToValues service against the unparsed sections of the document. An unparsed section is a section for which the EDI flat file schema for the EDI document did not have flat file structure and for which you created a separate flat file schema.</p> <ul style="list-style-type: none"> ■ Save information about where you are in processing the EDI document. The MAP flow operation saves the values of the <i>ediObject</i> and <i>EDIFFSchema</i> that the convertToValues service in the outer REPEAT loop is using. ■ Map the fully-qualified name of the flat file schema you created for the repeating section of the document into the <i>EDIFFSchema</i> variable. Make sure the value of <i>ediObject</i> is null.
5	<p>Repeatedly invoke the convertToValues service in this inner REPEAT loop to process an unparsed section of the document.</p>
6	<p>Invoke the <code>wm.b2b.edi.util:EDIconcat</code> service to prepare the unparsed section for the call to the convertToValues service.</p> <p>The EDIconcat service automatically gets its input from either the <code>unDefData</code> or <code>_RID_</code> variables that the first pass of the convertToValues service left in the pipeline for the unparsed sections. You can use the EDIconcat service to add a header and/or a trailer to the unparsed data, if needed.</p>

Flow operation	Description
7	<p>In the inner REPEAT loop, invoke the convertToValues service to:</p> <ul style="list-style-type: none"> ■ Convert the unparsed section to an IData object ■ Validate the EDI structure <p>Pass the output from the EDIconcat service (variable <i>output</i>) to the input variable <i>edidata</i> of the convertToValues service. In a previous flow operation, you set the value of the <i>EDIFFSchema</i> input variable to the fully-qualified name of the flat file schema to use for this unparsed section of the EDI document.</p>
8	<p>Map data from the unparsed section into the internal-format document. Depending on the complexity of your mapping requirements, you might need to add more logic than a MAP flow operation, or create a separate service to perform the mapping. For more information about how to map, see Chapter 5, “Mapping Data to Form New Documents”.</p>
9	<p>Invoke the convertToString service to convert the internal-format document from an IData object to String format.</p> <p>The inputs to the convertToString service include the IData object that contains the data for your internal-format document and the flat file schema for the internal-format document. The convertToString service uses the flat file schema to determine how to form the internal-format document. Alternatively, rather than use a flat file schema, the convertToString service also accepts an IS document type to define the structure of the internal-format document.</p>
10	<p>Add your own logic or invoke a service that you create to send the internal-format document to your internal application, e.g., a back-end system.</p>
11	<p>Restore the values of the variables <i>ediObject</i> and <i>EDIFFSchema</i> that you saved in a previous flow operation in preparation for invoking the convertToValues service in the outer REPEAT loop is using.</p>

Interchange (or Transmission) Envelope Validation and Compliance Check

If you want to perform the interchange envelope validation and the compliance check, you need to add logic to your service to do so.

Examples

For examples, see the flow services in the sampleServices folder of the WmEDIsamples package, which is located in the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>. The samples in this folder have been certified, meaning that they have been tested by webMethods.

- The sampleServices:X12toValues service illustrates how to use the `wm.b2b.edi:convertToValues` service to convert an ANSI X12 document to an `IData` object at one time and process it.
- The sampleServices:UNEDIFACTtoValues service illustrates how to use the `wm.b2b.edi:convertToValues` service to convert a UN/EDIFACT document to an `IData` object at one time and process it.
- The sampleServices:Iterator810 service illustrates how to process an ANSI X12 810 document interatively, segment by segment.

Additionally, the `Tutorial.EDItoXML:processEDI850_4101` service shows a complete example of converting an ANSI X12 850 document to an `IData` object and mapping the data from the EDI document to an XML document. For more information about this service, see [“Invoking the Final Service from the Service that Processes Your EDI Document” on page 83](#).



Important! If you downloaded the WmEDIsamples package from the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>, you should delete the WmEDIsamples package before going into production.

Generating Functional Acknowledgments

■ Overview	56
■ Before You Can Generate a Functional Acknowledgment	57
■ How the generateFA Service Reports the FA Status	59
■ Adding the Call to generateFA to Your Service	66
■ Specifying Inputs for the generateFA Service	69
■ Output from the generateFA Service	69

Overview

The EDI Module provides the `wm.b2b.edi.util:generateFA` service to generate functional acknowledgments (FAs).



Note: Functional acknowledgments (FAs) are not applicable to TRADACOMS.

To generate an FA, invoke the `generateFA` service from the service that you created to process an inbound EDI document. The `generateFA` service automatically determines the EDI standard of the document, performs validation using a flat file schema, and creates as output the FA.

For the ANSI X12 standard, the functional acknowledgment is the 997 document. In the UN/EDIFACT standard, the functional acknowledgment is the CONTRL message.

According to the EDI standard, an ANSI X12 997 acknowledges an ANSI X12 group envelope and all of its contents, and a UN/EDIFACT CONTRL message acknowledges an interchange envelope and all of its contents.



Note: FAs validate and acknowledge only the syntax of the document, not that the document has been processed or understood by the receiver.

The generated FA reports the FA status for a transaction, group, or UN/EDIFACT interchange as one of the following:

ANSI X12 997	UN/EDIFACT CONTRL	Description
N	N	Not Allowed
R	R	Rejected
P	R	Partially Accepted (for groups only)
E	R	Accepted, But Errors Were Noted
A	A	Accepted
FA	FA	EDI document is an FA (997 or CONTRL)
NR	NR	Not Required



Note: The rest of this chapter uses the FA statuses reported in an ANSI X12 997. Refer to this table to see the related FA status that is reported in a UN/EDIFACT CONTRL.

The value that the `generateFA` service reports for the FA status is dependent on whether a child transaction is allowed in its envelope -and- how the value you specify the `syntaxErrorStatus`, `logicalErrorStatus`, and `childTransactionRejectedStatus` input variables, for more information, see “[How the generateFA Service Reports the FA Status](#)” on page 59.

The generateFA service does not specify what to do with the FA that it creates. Therefore, in addition to invoking the generateFA service to generate the FA, your service should also invoke a service to deliver the FA to the sender of the original document.

To learn more about:

- Functional acknowledgments and how they fit in with processing an inbound EDI document, see Chapter 2, "Using the EDI Module without Other webMethods Components", in the *webMethods EDI Module Built-In Services Reference*.
- How to have the EDI Module automatically generate FAs when you are using Trading Networks, see [Chapter 15, "Optional Inbound Processing When Using Trading Networks"](#), specifically ["Automatically Generating Functional Acknowledgements" on page 258](#).
- How to generate a report for FA reconciliation, see [Chapter 21, "Reconciling Functional Acknowledgments"](#).
- The `wm.b2b.edi.util:generateFA` service, see the *webMethods EDI Module Built-In Services Reference*.

Before You Can Generate a Functional Acknowledgment

Before you use the `wm.b2b.edi.util:generateFA` service to generate FAs:

- Determine the level of an EDI document that you want the FA to acknowledge. When you invoke the generateFA service, you specify the level of the document you want to acknowledge using the `FALevel` input variable, which can be one of the following:
 - Default (ANSI X12 group or UN/EDIFACT interchange)
 - Transaction set (ANSI X12) or message (UN/EDIFACT)
 - Segment
 - Element
- Create or make sure you already have the flat file schema that defines the structure of the EDI document for which you are generating an FA. The generateFA service uses the flat file schema for validation of the EDI document. The flat file schema must be at least as detailed as the FA you want to create. That is, if you want to create an FA based on the transaction set level, the schema must parse down to that level. For instructions on how to create the flat file schemas, see ["Creating Flat File Schemas for EDI Documents" on page 20](#).
- Create or make sure you already have the flat file schema for the FA. Be sure you have defined the flat file schema for the appropriate standard and version of either the ANSI X12 997 document or the UN/EDIFACT CONTRL message.

- Configure the maximum number of errors to report per FA transaction. This is only necessary if you are generating FAs at the segment and element level. For instructions on how to configure, see “[Configuring the Maximum Number of Transaction Errors](#)” below.

Configuring the Maximum Number of Transaction Errors

When you want to generate FAs at the element level, configure the maximum number of errors to report per FA transaction. To configure the maximum number of errors to report, perform the following procedure to update the configuration in memory (so the changes take effect immediately) and in the WmEDI/config/properties.cnf file.

To configure the maximum number of transaction errors

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **Configuration** menu of the navigation panel, click **Configure Properties**.
- 4 Click **Edit Properties Settings**.
- 5 Add or update the *EDIMaxFATransactionErrors* property to specify one of the following:

<u>Specify this value</u>	<u>To indicate...</u>
n (a positive whole number)	The maximum number of errors that can be reported for any one FA transaction.
-1	Any number of errors can be reported for any one FA transaction.

The default value is 100, which indicates that a maximum of 100 errors can be reported for any one FA transaction.

- 6 Click **Save Changes**.

How the generateFA Service Reports the FA Status

To determine the FA status for a transaction, group, or UN/EDIFACT interchange, the `wm.b2b.edi.util:generateFA` service uses the information described in the following table. See the sections below the table for more information about each item listed in the table.

Section	Information the generateFA service uses to determine the FA status
Transaction	<ul style="list-style-type: none"> ■ Whether the transaction is allowed in the group in which it resides ■ Whether the transaction is an FA (997 or CONTRL) ■ <i>syntaxErrorStatus</i> input variable ■ <i>logicalErrorStatus</i> input variable
Group (The generateFA service will report FA status for UN/EDIFACT groups if a UN/EDIFACT document contains groups.)	<ul style="list-style-type: none"> ■ Whether the group is “FA” or “CONTRL” ■ <i>syntaxErrorStatus</i> input variable ■ <i>logicalErrorStatus</i> input variable ■ <i>childTransactionRejectedStatus</i> input variable
Interchange (UN/EDIFACT only)	<ul style="list-style-type: none"> ■ <i>syntaxErrorStatus</i> input variable ■ <i>logicalErrorStatus</i> input variable ■ <i>childTransactionRejectedStatus</i> input variable

Child Transaction that is Not Allowed in its Group

When setting the FA status for a child transaction, the first check that the `generateFA` service makes is to determine whether the child transaction is allowed in its group. For example, it is not valid for an ANSI X12 997 transaction to be in an ANSI X12 PO group. Similarly, it is not valid for a UN/EDIFACT CONTRL to be in an ORDERS group.

If the child transaction is:

- **Allowed**, the `generateFA` service determines the rest of the statuses (e.g., syntax error status, logical error status, child transaction rejected status) for the child element.
- **Not allowed**, the `generateFA` service sets the FA status to “Not Allowed” and does not determine the rest of the statuses.

Transaction or Group is for an FA

The generateFA service reports the FA status “FA” when a transaction *is* an FA or a group that contains an FA. In other words, in the following circumstances:

- A transaction is an FA, that is either an ANSI X12 997 or UN/EDIFACT CONTRL
- A group is either an ANSI X12 “FA” or UN/EDIFACT “CONTRL” group.

Syntax Error Status

The *syntaxErrorStatus* input variable to the generateFA service indicates how you want the service to report the syntax error status. The syntax error status indicates whether there are syntax errors in the transaction, group, or UN/EDIFACT interchange, for example, missing mandatory elements, violation of syntax rules, invalid field lengths, code list violations, or segment repeat counts exceeded.

The following table describes the settings of the *syntaxErrorStatus* input variable and their meanings:

Set <i>syntaxErrorStatus</i> to:	To have the generateFA service:
Rejected	<p>Report the syntax error status:</p> <ul style="list-style-type: none">■ “Accepted” if there are no syntax errors■ “Rejected” if there are syntax errors <p>Use this setting if you want to reject the element (e.g., transaction) because of the syntax errors.</p>
Accepted, But Errors Were Noted	<p>Report the syntax error status:</p> <ul style="list-style-type: none">■ “Accepted” if there are no syntax errors■ “Accepted, But Errors Were Noted” if there are syntax errors <p>Use this setting if you want to know whether there were syntax errors, but do not want to reject the element (e.g., transaction) because of the syntax errors.</p>
Accepted	<p>Always reports the syntax error status as “Accepted” regardless of whether there are any syntax errors.</p> <p>Use this setting if you do not want to check for syntax errors.</p>

Logical Error Status

The *logicalErrorStatus* input variable to the generateFA service indicates how you want the service to report the logical error status. The logical error status indicates whether the transaction, group, or UN/EDIFACT interchange is malformed. For example:

- The control number in a header does not match the control number in the corresponding trailer.
- The segment count in a trailer does not have an accurate group, transaction, or segment count.

The following table describes the settings of the *logicalErrorStatus* input variable and their meanings:

Set <i>logicalErrorStatus</i> to:	To have the generateFA service:
Rejected	<p>Report the logical error status:</p> <ul style="list-style-type: none"> ■ “Accepted” if there are no logical errors ■ “Rejected” if there are logical errors <p>Use this setting if you want to reject the element (e.g., transaction) because of the logical errors.</p>
Accepted, But Errors Were Noted	<p>Report the logical error status:</p> <ul style="list-style-type: none"> ■ “Accepted” if there are no logical errors ■ “Accepted, But Errors Were Noted” if there are logical errors <p>Use this setting if you want to know whether there were logical errors, but do not want to reject the element (e.g., transaction) because of the logical errors.</p>
Accepted	<p>Always reports the logical error status as “Accepted” regardless of whether there are any logical errors.</p> <p>Use this setting if you do not want to check for logical errors.</p>

Child Transaction Rejected Status

The *childTransactionRejectedStatus* input variable to the generateFA service indicates how you want the service to report the child transaction rejected status. The child transaction rejected status indicates whether child elements of a group or UN/EDIFACT interchange have an FA status of "Rejected". That is:

- For a group, whether any transaction within a group has an FA status of "Rejected"
- For a UN/EDIFACT interchange, whether any transaction or group within the interchange has an FA status of "Rejected"



Note: For information about how the EDI Module sets the FA status for a transaction or group based on the syntax error status and the logical error status, see "["Transaction FA Status"](#) on page 63 and "["Group FA Status"](#) on page 64.

The following table describes how the generateFA service sets the child transaction rejected status based on the setting of the *childTransactionRejectedStatus* input variable and the FA statuses of the child transactions:

Setting of <i>childTransactionRejected Status</i>	When the FA statuses of the child transactions are:			
	All "Accepted"	"Rejected" -or- "Accepted, But Errors Were Noted" -and- no "Accepted"	"Rejected" -or- "Accepted, But Errors Were Noted" -and- at least one "Accepted"	All "Rejected"
The child transaction rejected status is set to:				
Rejected	Accepted	Rejected	Rejected	Rejected
Partially Accepted	Accepted	Accepted, But Errors Were Noted	Partially Accepted	Rejected
Accepted, But Errors Were Noted	Accepted	Accepted, But Errors Were Noted	Accepted, But Errors Were Noted	Rejected

How the EDI Module Determines Which FA Status to Use

After determining the syntax error status, logical error status, and child transaction rejected status (if applicable), the generateFA service can determine the FA status for a transaction, group, or UN/EDIFACT interchange.

Transaction FA Status

The generateFA service uses the most restrictive value in the following list when setting the FA status for a transaction:

- Whether the transaction is an FA (997 or CTRL); for more information about transactions that are not allowed, see [“Transaction or Group is for an FA” on page 60](#).
- Syntax error status; for more information about how the EDI Module sets the syntax error status, see [“Syntax Error Status” on page 60](#).
- Logical error status; for more information about how the EDI Module sets the logical error status, see [“Logical Error Status” on page 61](#).

The following table shows the possible combinations of values and how the generateFA service sets the FA status for a transaction based on these values.

The EDI Module reports this FA status for a transaction...	When the transaction has the following statuses...			
	Is transaction allowed?	Is the transaction an FA?	Syntax Error Status	Logical Error Status
Not Allowed	Not Allowed	Is an FA	Any syntax error status	Any logical error status
FA	Allowed	Is an FA	Any syntax error status	Any logical error status
Rejected	Allowed	<i>Not</i> an FA	■ At least one status is “Rejected”.	
Accepted, But Errors Were Noted	Allowed	<i>Not</i> an FA	■ Statuses are either “Accepted, But Errors Were Noted” -or- “Accepted”. ■ At least one status is “Accepted, But Errors Were Noted”.	
Accepted	Allowed	<i>Not</i> an FA	Accepted	Accepted

Group FA Status

The generateFA service uses the most restrictive value in the following list when setting the FA status for a group:

- Whether the group is “FA” or “CONTRL”, see “[Transaction or Group is for an FA](#)” on [page 60](#).
- Syntax error status; for more information about how the EDI Module sets the syntax error status, see “[Syntax Error Status](#)” on [page 60](#).
- Logical error status; for more information about how the EDI Module sets the logical error status, see “[Logical Error Status](#)” on [page 61](#).
- Child transaction rejected status; for more information about how the EDI Module sets the child transaction rejected error status, see “[Child Transaction Rejected Status](#)” on [page 62](#).

The following table shows the possible combinations of values and how the generateFA service sets the FA status for a group based on these values.

The EDI Module reports this FA status for a group...	When the group has the following statuses...			
	Is group “FA” or “CONTRL”	Syntax Error Status	Logical Error Status	Child Transaction Rejected Status
FA	Is an “FA” or “CONTRL”	Any syntax error status	Any logical error status	Any child transaction rejected status
Rejected	Is <i>not</i> an “FA” or “CONTRL”	■ At least one status is “Rejected”.		
Partially Accepted	Is <i>not</i> an “FA” or “CONTRL”	Accepted, But Errors Were Noted Accepted	Accepted, But Errors Were Noted Accepted	Partially Accepted
Accepted, But Errors Were Noted	Is <i>not</i> an “FA” or “CONTRL”	<ul style="list-style-type: none"> ■ Statuses are either “Accepted, But Errors Were Noted” -or- “Accepted”. ■ At least one status is “Accepted, But Errors Were Noted”. 		
Accepted	Is <i>not</i> an “FA” or “CONTRL”	Accepted	Accepted	Accepted

UN/EDIFACT Interchange FA Status

The generateFA service uses the most restrictive value of the following when setting the FA status for a UN/EDIFACT interchange:

- Syntax error status; for more information about how the EDI Module sets the syntax error status, see “[Syntax Error Status](#)” on page 60.
- Logical error status; for more information about how the EDI Module sets the logical error status, see “[Logical Error Status](#)” on page 61.
- Child transaction rejected status; for more information about how the EDI Module sets the child transaction rejected error status, see “[Child Transaction Rejected Status](#)” on page 62.

The following table shows the possible combinations of values and how the generateFA service sets the FA status for a UN/EDIFACT interchange based on these values.

The EDI Module reports this FA status for an interchange...	When the interchange has the following statuses...		
	Syntax Error Status	Logical Error Status	Child Transaction Rejected Status
Rejected	■ At least one status is “Rejected”.		
Partially Accepted	Accepted, But Errors Were Noted Accepted	Accepted, But Errors Were Noted Accepted	Partially Accepted
Accepted, But Errors Were Noted	■ Statuses are either “Accepted, But Errors Were Noted” -or- “Accepted”. ■ At least one status is “Accepted, But Errors Were Noted”.		
Accepted	Accepted	Accepted	Accepted
Not Required	Any syntax error status	Any logical error status	Any child transaction rejected status

Adding the Call to generateFA to Your Service

As previously stated, you invoke the `wm.b2b.edi.util:generateFA` service from the service you created to process an inbound EDI document. Where you place the call to the `generateFA` service depends on whether the service you created processes the EDI document all at one time, or iteratively segment by segment. For more information about these two options, see “[Determining the Method to Use to Process](#)” on page 42.

When You Process the Document at One Time

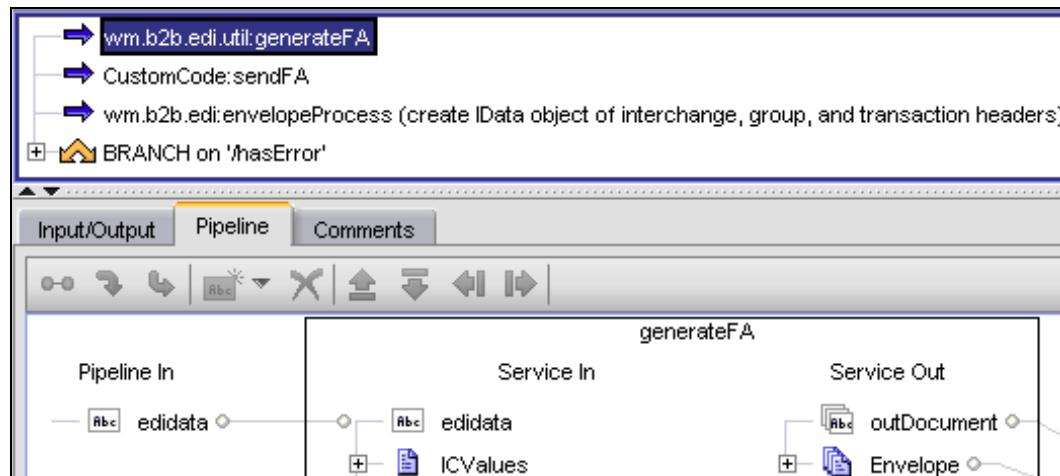
When you process the EDI document at one time, you can invoke the `generateFA` service either before invoking the `wm.b2b.edi:envelopeProcess` service or immediately after invoking the `envelopeProcess` service. For a review of the logic in a service that processes the document at one time, see “[Processing the Entire Document at One Time](#)” on page 43.

Invoking generateFA Before Invoking envelopeProcess

When you invoke the `generateFA` service *before* the `envelopeProcess` service, the service takes as input the unparsed EDI document. The unparsed document is in the pipeline in the variable `edidata`. The `edidata` variable is placed in the pipeline by an EDI content handler. For more information about the `edidata` variable, see “[Inputs to Your Service](#)” on page 41.

The following shows the service from the section “[Processing the Entire Document at One Time](#)” on page 43 that has been updated to include the call to the `generateFA` service. Note that in the Pipeline section of the screen, the `edidata` variable in Pipeline In is mapped to the `edidata` in Service In. For more information about how to set the rest of the `generateFA` service input variables, see “[Specifying Inputs for the generateFA Service](#)” on page 69.

Sample code for invoking `generateFA` before invoking `envelopeProcess`



After invoking the generateFA service to generate the FA, add logic or invoke a service that you create to deliver the FA to the sender of the original EDI document.

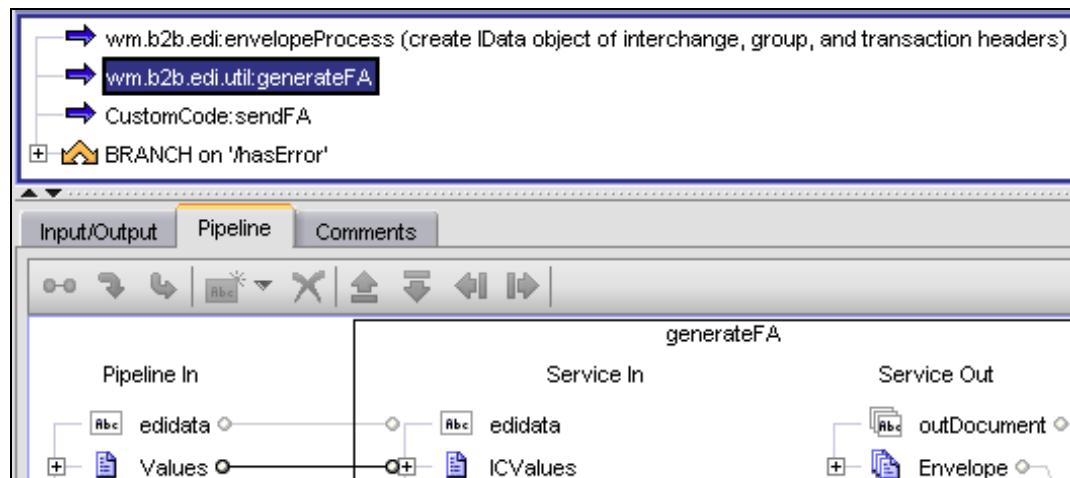
Invoking generateFA Immediately after Invoking envelopeProcess

When you invoke the generateFA service *after* the envelopeProcess service, you can pass as input to the service the generateFA service the parsed EDI document that is the output from the envelopeProcess service.

 Note: If you are using previously parsed and validated data as input to the generateFA service, the input data must be the correctly formatted results of proper services, meaning that the error array must be included in the results.

The following shows the service from the section “[Processing the Entire Document at One Time](#)” on page 43 that has been updated to includes the call to the generateFA service. In the Pipeline section of the screen, the *Values* variable in Pipeline In is the output from the envelopeProcess service. The *ICValues* variable in Service In is the input variable of the generateFA service that accepts a parsed EDI document. In the Pipeline section of the screen, map the *Values* variable to the *ICValues* variable, as shown below. For more information about how to set the rest of the generateFA service input variables, see “[Specifying Inputs for the generateFA Service](#)” on page 69.

Sample code for invoking generateFA immediately after invoking envelopeProcess



After invoking the generateFA service to generate the FA, add logic or invoke a service that you create to deliver the FA to the sender of the original EDI document.

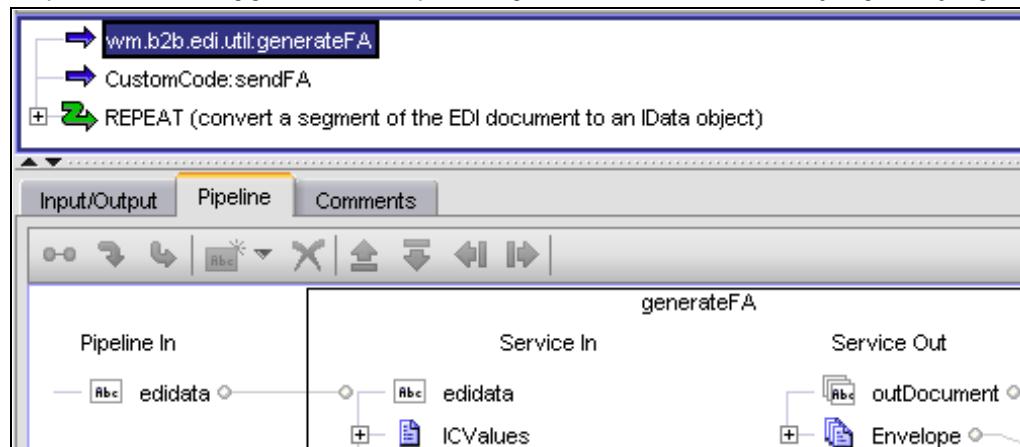
When You Process the Document Iteratively Segment by Segment

When you process the EDI document at iteratively, segment by segment, invoke the generateFA service before performing other processing. For a review of the logic in a service that processes the document iteratively, see [“Processing the Document Iteratively Segment by Segment” on page 47](#).

When you invoke the generateFA service, the service takes as input the unparsed EDI document. The unparsed document is in the pipeline in the variable *edidata*. The *edidata* variable is placed in the pipeline by an EDI content handler. For more information about the *edidata* variable, see [“Inputs to Your Service” on page 41](#).

The following shows the service from the section [“Processing the Document Iteratively Segment by Segment” on page 47](#) that has been updated to include the call to the generateFA service. Note that in the Pipeline section of the screen, the *edidata* variable in Pipeline In is mapped to the *edidata* in Service In. For more information about how to set the rest of the generateFA service input variables, see [“Specifying Inputs for the generateFA Service” on page 69](#).

Sample code for invoking generateFA when processing an inbound document iteratively, segment by segment



After invoking the generateFA service to generate the FA, add logic or invoke a service that you create to deliver the FA to the sender of the original EDI document.

Specifying Inputs for the generateFA Service

For details about the input variables of the `wm.b2b.edi.util:generateFA` service, see the *webMethods EDI Module Built-In Services Reference*.

Output from the generateFA Service

The output from the `wm.b2b.edi.util:generateFA` service is the FA itself. As mentioned, this service does not specify what to do with the FA that it has created. Therefore, any service that you use or develop that calls the `generateFA` service must also invoke a service to deliver the FA to the original sender.

Each FA contains a code that indicates whether the validation was successful. This code appears in a different position for each level. For example, the code in a Transaction Set level FA will appear in AK501. For more information about the location of this code in your documents, see the documentation for your EDI standard and version.

The appropriate position of the transaction, group, or UN/EDIFACT interchange will contain one of the following codes for the FA status:

ANSI X12 997	UN/EDIFACT CONTRL	Description
N	N	Not Allowed
R	R	Rejected
P	R	Partially Accepted (for groups only)
E	R	Accepted, But Errors Were Noted
A	A	Accepted
FA	FA	EDI document is an FA (997 or CONTRL)
NR	NR	Not Required

Mapping Data to Form New Documents

■ Overview	72
■ Before Mapping Data	72
■ Mapping EDI Documents to Other Formats	73
■ Mapping XML Documents to EDI Format	84

Overview

You might need to map documents when you:

- **Process inbound EDI documents.** You might need to map data from the inbound document to another format, for example the format that your back-end system requires.
- **Form outbound EDI documents.** You might need to map data from an internal-format document (e.g., a document from a back-end system), and then send the EDI document outbound.

This chapter provides basic information and suggestions on mapping techniques. Before you begin using this chapter, webMethods recommends that you install the webMethods WmEDIsamples package, which is located in the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>. The samples in this folder have been certified, meaning that they have been tested by webMethods. Many examples in this chapter refer to the Tutorial folder of this package.

For background concepts on mapping variables, using transformers, and using the pipeline, see the *webMethods Developer User's Guide*.

Before Mapping Data

Typically, you need IS document types that define the structure of:

- The EDI document that you are mapping data from or to. You can create an IS document type from a flat file schema from the Flat File Schema Editor of the webMethods Developer.
- The internal-format document that you are mapping data from or to. If the internal-format document is an XML document that has an associated DTD, you can create the IS document type from the DTD.

For instructions about how to create an IS document type, see the *webMethods Developer User's Guide*.



Important! Be sure your IS document types are absolutely correct and complete.

Mapping EDI Documents to Other Formats

To map data from an EDI document to another format, you might need to create a mapping service. Depending on the transaction set and the target document, it might be difficult to perform all your mapping logic in a single mapping service. Consider creating services that map individual segments/sections of the document, especially when those segments have multiple occurrences (e.g., N1, REF, etc.). This technique is called *segment mapping*. Then create a final mapping service that invokes each segment-mapping service.

The advantages of segment mapping include:

- Modular services allow easy testing and debugging
- Small maps reduce the chance of mapping errors because they are self-contained
- Interchangeable maps (For example, if a segment of a transaction set is different for different trading partners, only that segment needs to be replaced instead of the entire map.)
- Mapped segments can be used much like a utility that can be incorporated into different maps

This section describes how to create a segment-mapping service, how to transform data from the source document before mapping it to the target document, best practices to use when creating mapping services, and how to build the final mapping service by invoking the individual segment-mapping services.

Examples

The EDI Module provides the examples used in this section in the Tutorial.EDItoXML folder of the WmEDIsamples package, which is located in the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>. To see the complete services (rather than just screenshots of them), see the services in that folder.

Creating Services that Map Data from a Segment of the EDI Document

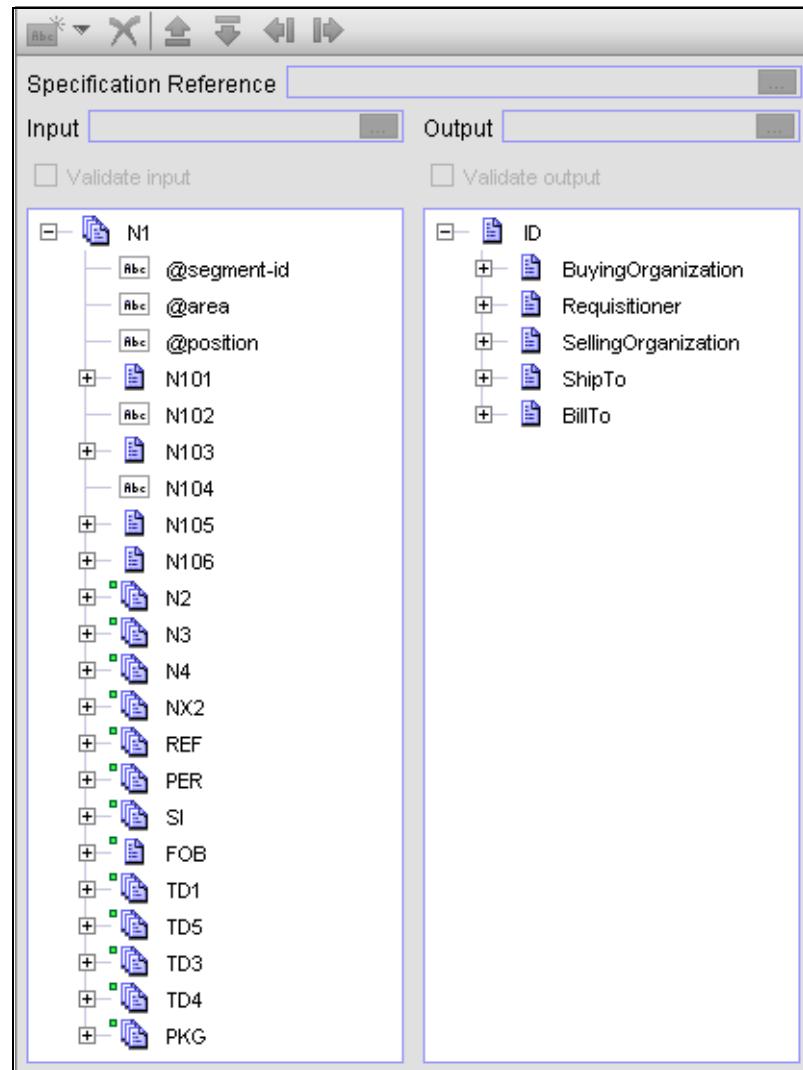
This section describes an example of mapping the N1 segment of an ANSI X12 850 document. An N1 segment is a good candidate for a segment-mapping service because multiple N1 segments can occur in an ANSI X12 850 document and where you want to map the N1 segment in the target document depends on the value of the N101 field.

 Note: For examples of mapping DTM, PO1, and REF segments, see the other services in the Tutorial.EDItoXML.segmentMapping folder, which is in the WmEDIsamples package, which is located in the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>.

The following lists the steps you might perform to create a segment-mapping service for an N1 segment.

- 1 From the webMethods Developer, create a folder named segmentMapping. Use this folder to store all of your segment-mapping services.
- 2 Create an empty flow service named N1Segment that will perform the logic to map the N1 segment.
- 3 Define the input and output variables for the N1Segment service.
 - The input is the N1 segment. When defining the input variables, you can copy and paste the variables for the N1 segment from the IS document type for the EDI 850 transaction.
 - The output is the target IS document (IData object) to which you are mapping the data. In the case of the N1 segment, the output is the section of the target document that contains address information. You can copy and paste the address section of the IS document type for the target document to the output.

Inputs and Outputs of the Tutorial.EDIttoXML.segmentMapping:N1Segment service



- 4 Add a LOOP flow operation to the N1Segment service.

The loop is needed because multiple N1 segments can occur in the EDI document. The N1 segments are in the N1 IData object, which is input to the N1Segment service. To loop over the N1 segments, when defining the LOOP flow operation, in the Properties panel, set the **Input array** property to /N1.

LOOP flow operation in the Tutorial.EDItoXML.segmentMapping:N1Segment service

Properties	
	LOOP over 'N1'
Property	Value
General	
Comments	
Scope	
Timeout	
Label	<input type="button" value="▼"/>
Input array	N1
Output array	

- 5 Add BRANCH flow operations under the LOOP flow operation, so you can perform mapping based on the value of the N101 field.

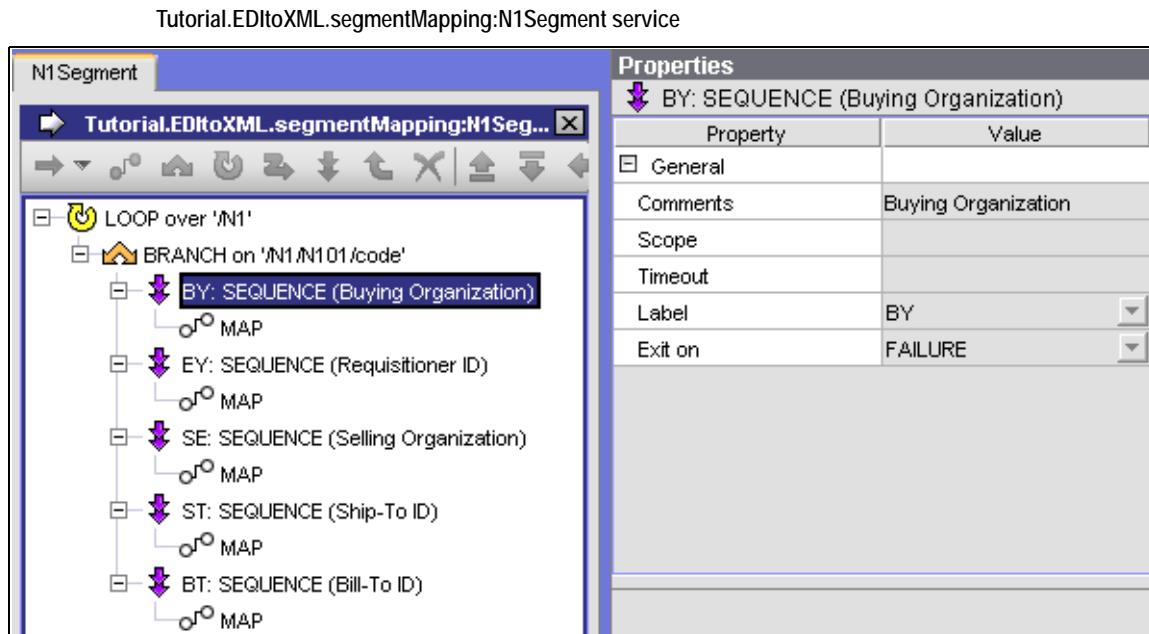
Depending on the value of N101 field, the segment either contains Buying Organization (BY), Requisitioner ID (EY), Selling Organization (SO), Ship-To ID (ST), or Bill-To ID (BT). To branch based on the type of N101 field, when you define the BRANCH, in the Properties panel, set the Switch property to N1/N101/code, which is the variable within the input N1 IData object that contains the value of the N101 field.

BRANCH flow operation in the Tutorial.EDItoXML.segmentMapping:N1Segment service

Properties	
	BRANCH on 'N1/N101/code'
Property	Value
General	
Comments	
Scope	
Timeout	
Label	<input type="button" value="▼"/>
Switch	N1/N101/code
Evaluate labels	<input type="button" value="▼"/>

- 6 Under the BRANCH, add SEQUENCE flow operations. You add one SEQUENCE flow operation for each possible value of the N101 field. When you define each SEQUENCE, in the Properties panel, set the Label property to the value of the N101 field that the SEQUENCE is to handle. For example, set the Label property to BY for the SEQUENCE that is going to contain the mapping logic to map Buying Organization information.

Under each SEQUENCE add a MAP flow operation to perform the appropriate mapping logic. Each sequence maps to a particular section of the target IS document (IData object).



Transforming Source Data Before Mapping it to the Target

Sometimes it is necessary to transform the values of fields in the source document before mapping those values to the corresponding fields in the target document. Some examples are:

- You might need to convert from a six-digit date to an eight-digit date that includes the century, or your target might require hour, minutes, and seconds that is not part of the EDI document.
- You might need to concatenate multiple Strings.

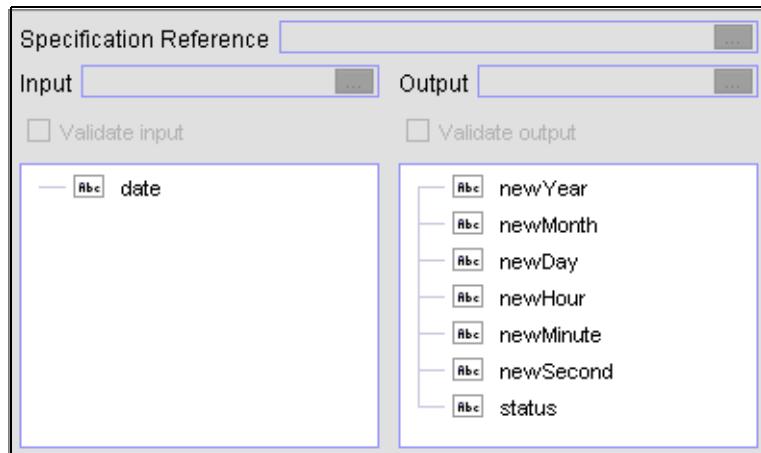
Converting Field Values

To convert field values from one form to another, you can create a Java service that generates the correct data and leaves it in the pipeline for later use.

Suppose that you have a source IS document (IData object) with a date field that can be either six or eight digits. For your target, you require a four-digit year, two-digit month, two-digit day, two-digit hour, two-digit minute, and two-digit second. Because this is a special situation for which no existing flow service is available, you can write a Java service perform the date/time format conversion. Be sure to define the outputs of your Java service, so that they can be referenced in the pipeline for later use.

The following shows the inputs and outputs for the Tutorial.EDItosXML:dateParse service that is provided in the WmEDIsamples package, which is located in the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>.

Inputs and Outputs of the Tutorial.EDItoXML:dateParse service

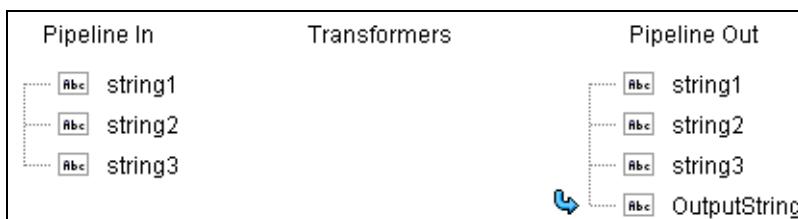


Insert the data conversion service in an appropriate location in your flow service. After the service executes the re-formatted field value that is the output from your service is available in the pipeline for you to use later.

Concatenating Strings

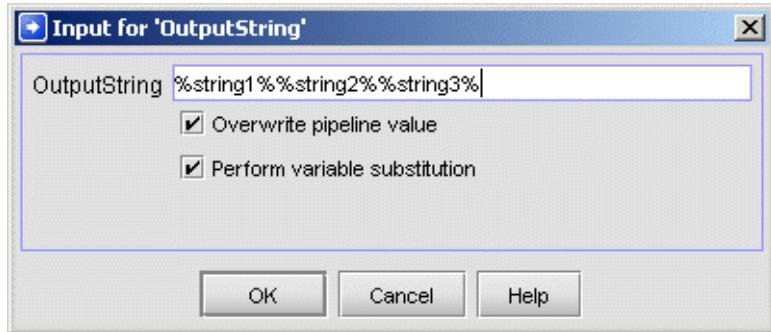
An easy way to append multiple strings is to use variable substitution. Suppose your pipeline has three Strings, *string1*, *string2* and *string3*. You want to map the concatenation of the three strings to the *OutputString* variable.

Sample pipeline for concatenating Strings



To use variable substitution, right click *OutputString* and select Set Value. In the Input for dialog box, check the Perform Variable Substitution check box. Then type the variables that you want to concatenate enclosed in % symbols. For this example, you would type %string1%%string2%%string3% to concatenate the strings.

Using variable substitution to concatenate strings



If *string1* is *Con*, *string2* is *cat*, and *string3* is *enate*, the result is *Concatenate*. You might have to check for null for the Strings. If *string1* is *Con*, *string2* is *cat*, and *string3* is null, your output would be *Concat%string3%*.

Best Practices for Creating Mapping Services

The following lists best practices that you should consider when creating your mapping services:

- Define the inputs and outputs of your services first. It is a good practice to always specify the service inputs/outputs immediately after you create a new service. This way, you will always know what inputs you need from other services and what outputs your service is to produce. If you are working with a group of developers who are each creating services, you can review the services' inputs and outputs to determine whether there is a mismatch or miscommunication between the developers.
- If possible, use copy and paste to define your input/output variables. If you already have an IS document type that defines the structure of the source document that you are mapping from and/or the target document that you are mapping to, copy and paste the sections of the IS document type you need for your service input and/or output.
- Clean up the pipeline. Data that your service can use is in the pipeline. When you combine the entire segment mapping services in a complete mapping service, you have in the pipeline sections of the complete target IS document (IData object). Because it is easy to leave things in the pipeline, you always should keep track of what you have in the pipeline and clean up whenever necessary.

Typically, you might want to leave certain IS documents in the pipeline when you are developing or debugging a service. After the service is working, you should leave only the necessary pipeline variables (usually only the outputs of the service) in the pipeline and drop the pipeline variables that your service does not need.

- Understand how to process and create String Lists and document lists (array of IData objects). To process a String List or document list that is in the input IS document type, use a LOOP flow operation. For each iteration of the loop, your service can act on an

individual String within a String List or an individual IS document (IData object) with in a document list.

If your output IS document type requires a String List or document list, you can use the **out-array** property of the LOOP flow operation. If you output IS document type does not require String Lists or document lists, you can use a combination of a LOOP and BRANCH flow operations to go from an array structure to a flat structure. The flow operation you use depend on the specific mapping situation.

- Reference elements within IS documents (IData objects) from the top of hierarchy. If you need to identify an element within an IS document for a property on the Properties tab of a flow operation, be sure to identify the element from the top of the hierarchy. For example, suppose you have the following IS document:

```
N1  
N101  
code
```

If you want to reference the code element, do so by using N1/N101/code.

Building the Final Mapping Service From Segment-Mapping Services

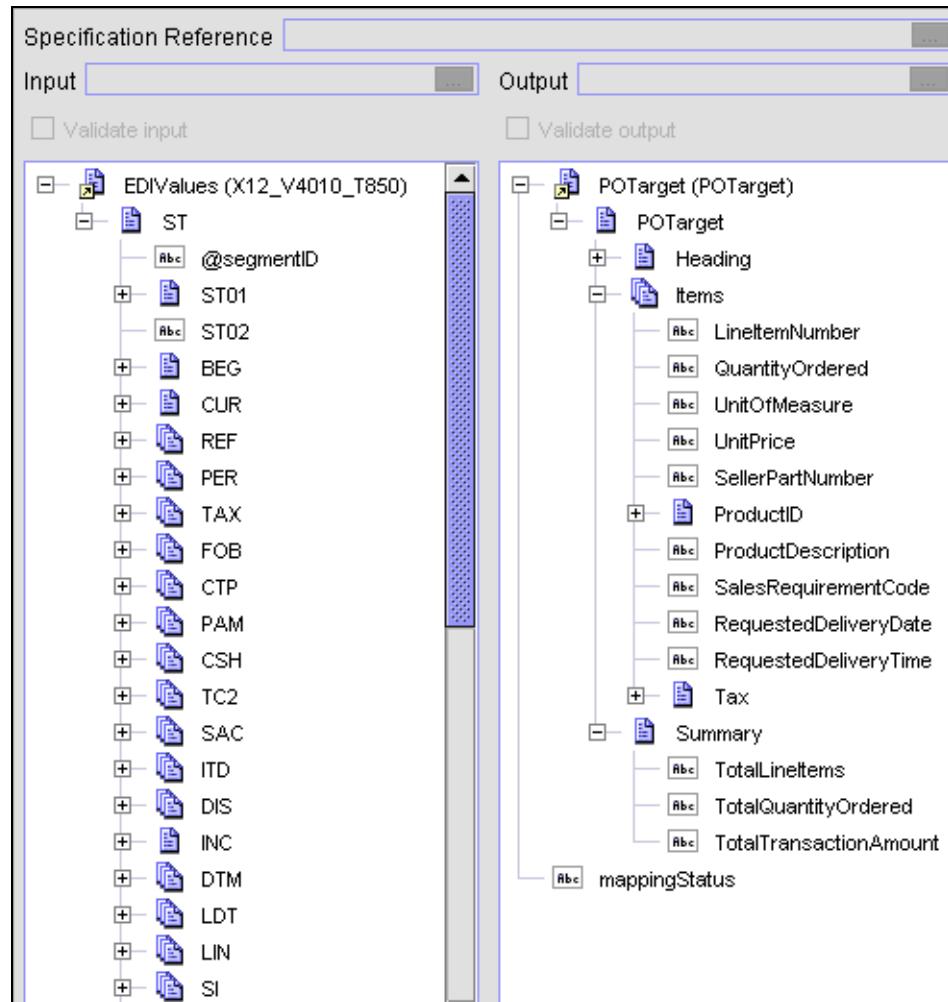
After you create your segment-mapping services, you can create your final mapping service. This service invokes the segment-mapping services to do the complete mapping of the source document to the target document.

This section describes an example of mapping an ANSI X12 850 document (in IData object format) to an IData object for a purchase order. Among other segment-mapping services, the example mapping service described in this section, Tutorial.EDItosXML:EDItosXMLPOMap, invokes the Tutorial.EDItosXML.segmentMapping:N1Segment mapping service described in [“Creating Services that Map Data from a Segment of the EDI Document”](#).

The following lists the steps you might perform to the mapping service.

- 1 From the webMethods Developer, create an empty flow service named EDItosXMLPOMap.
- 2 Define the input and output variables for the EDItosXMLPOMap service.
 - The input is the EDI850_4010 IS document, which is populated with values when the service is invoked.
 - The output of is the purchase order IS document (Tutorial.Records:POTarget).

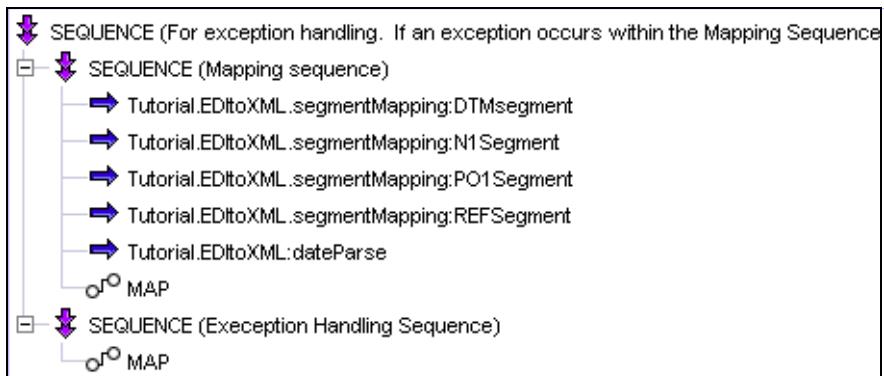
Inputs and Outputs for the Tutorial.EDItoXML:EDItoXMLPOMap service



- 3 Add flow operations to invoke the segment-mapping services and Java services you might have written to get the necessary data in the pipeline for the output target document; that is the data for the purchase order, Tutorial.Records:POTarget.

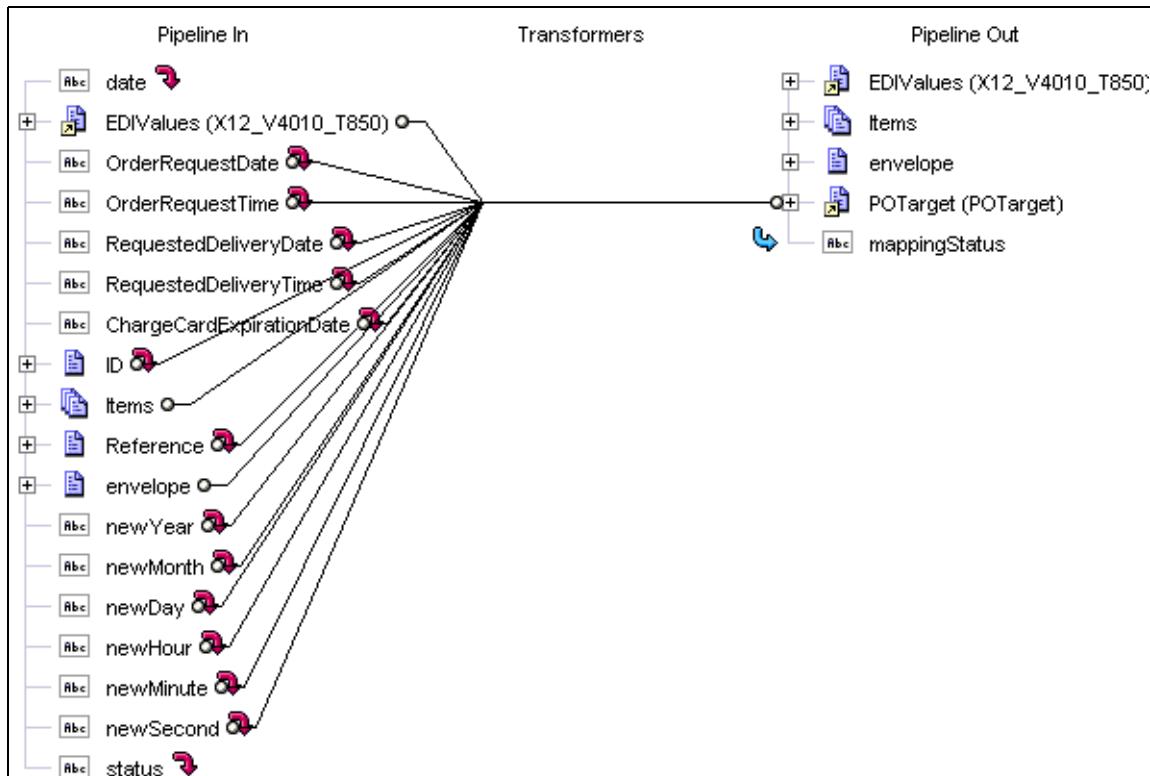
You invoke your segment-mapping services one at a time to map data to one section of the target at a time, and leave the completed sections in the pipeline.

Tutorial.EDItoXML:EDItoXMLPOMap service



- 4 Add a final MAP flow operation at the end of the service to do a final map of all sections to the final target. The final map consists of one-to-one mapping where elements are mapped from source to target and from pipeline to target. See the following illustration.

Final map of the Tutorial.EDItoXML:EDItoXMLPOMap service

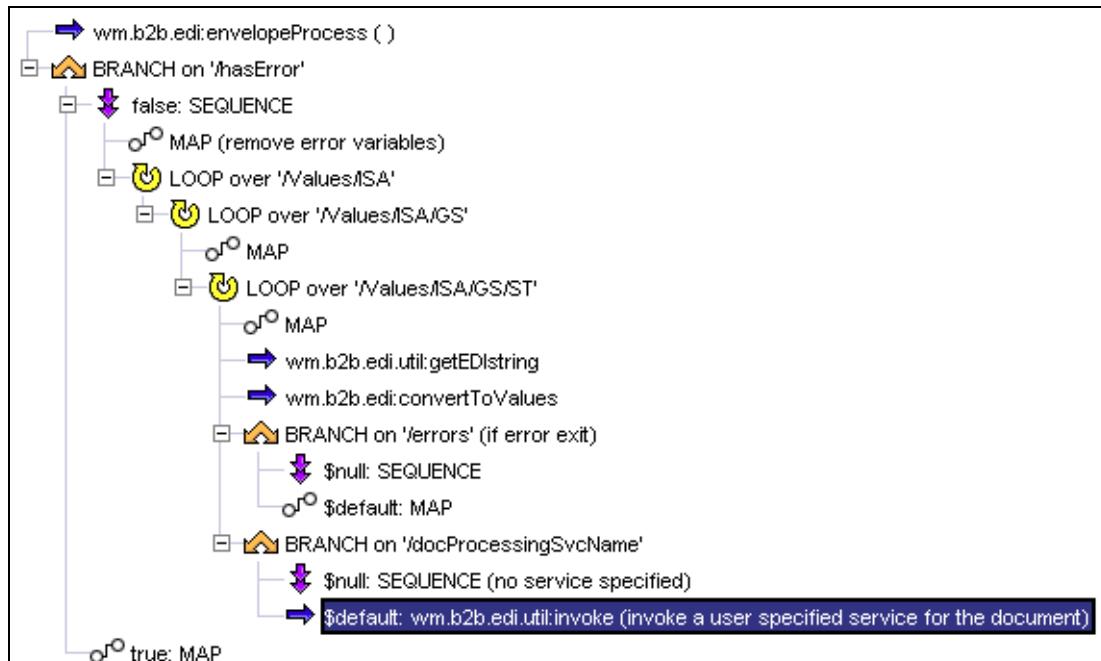


To get a better understanding of the mappings, view the Tutorial.EDItoXML:EDItoXMLPOMap service from the webMethods Developer

Invoking the Final Service from the Service that Processes Your EDI Document

After you complete the final mapping service, you can invoke it from the service you create to process an EDI document. The Tutorial.EDItoXML folder in the WmEDIsamples includes the Tutorial.EDItoXML:processEDI850_4010 service to process the EDI document. This service invokes the sampleServices:X12toValues service, which is also in the WmEDIsamples package, to convert an incoming ANSI X12 EDI document to an IData object.

sampleServices:X12toValues



Note the last BRANCH flow operation in the X12ToValues service. If the variable *docProcessingSvcName* (which is an input variable to the X12ToValues service) is not null, the highlighted flow operation executes:

```
$default: invoke (invoke a user specified service for the document)
```

This INVOKE flow operation, invokes the *wm.b2b.edi.util:invoke* service. The *wm.b2b.edi.util:invoke* service, in turn, invokes the service identified in the *docProcessingSvcName* variable. The output for the *wm.b2b.edi.util:invoke* service is in the variable *output*.

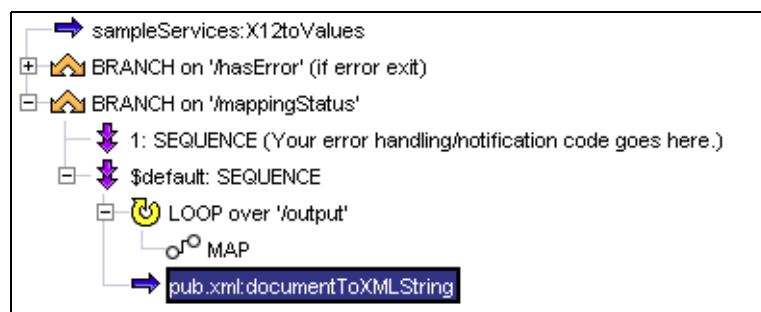
The processEDI850_4010 service sets the value for the *docProcessingSvcName* variable to Tutorial.EDItoXML:EDItoPOXMLMap, and as a result the Tutorial.EDItoXML:EDItoPOXMLMap service is executed to perform the mapping.



Note: Before the `wm.b2b.edi.util:invoke` service will invoke a service, that service must be listed in the `webMethods6\IntegrationServer\packages\WmEDI\config\services.cnf` configuration file. For more information about the `wm.b2b.edi.util:invoke` service, see the *webMethods EDI Module Built-In Services Reference*.

At the end of the `processEDI850_4010` service, the `pub.xml:documentToXMLString` service is invoked to generate the final target XML document. When you use the `pub.xml:documentToXMLString` service, be sure to specify the `documentTypeName` input variable to preserve the order of your XML data.

Tutorial.EDItoXML:processEDI850_4010



Mapping XML Documents to EDI Format

As you did to map data from an EDI document, when you map data from an XML document to an EDI document, you might need to create a segment-mapping services and a final mapping service.

Creating Services that Map Data into a Segment of the EDI Document

Similar to mapping from EDI to XML, when you map XML to EDI, it is useful to map a segment at a time. Segment mapping is important because you will want to break up the complete mapping into smaller chunks (segments) to make mapping and debugging easier. You also may want to reuse some of the segment maps.

For examples of mapping XML data to segments of EDI documents, see the following services in the Tutorial.XMLtoEDI folder that is in the WmEDIsamples package:

- XMLToBEG
- XMLToISA
- XMLToREF
- XMLToDTM
- XMLToN1
- XMLToTD5
- XMLToFOB
- XMLToPO1_CTT
- XMLToTXI
- XMLToGS

The structure of the XML data that these services use is defined in the IS document type Tutorial.Records:POSource.

Typically, XML data has a relatively flat structure (e.g., contains IS documents (IData objects)), while an EDI document contains array structures (e.g., IS document lists (array of IData objects)). As a result, you will have to create document lists from a series of IS documents. To do so, you can use the Tutorial.XMLtoEDI:appendSegment service. For examples of how to use the Tutorial.XMLtoEDI:appendSegment service, review the segment-mapping services listed above.

Building the Final Mapping Service from Segment-Mapping Services

After you create your segment-mapping services, you can create your final mapping service. The final mappings service invokes the segment-mapping services to perform the complete mapping of the source document to the target document. Review the Tutorial.XMLtoEDI:XMLPOtoEDIMap service in the WmEDIsamples package for an example of how to combine several segment-mapping services into a service to map XML data to an EDI document.

To see an example of a document that processes an internal-format document (in XML format) by converting it from String format to an IData object and then maps data to form an EDI document, see the Tutorial.XMLtoEDI:processXMLSource service in the WmEDIsamples package.

Best Practices When Mapping From XML to EDI

The following lists best practices to consider when mapping XML data to an EDI document:

- To convert XML-formatted data to an IData object, use the pub.xml:xmlStringToXmlNode (or pub.xml:loadXmlNode) service and the pub.xml:xmlNodeToDocument service.
- To truncate characters in the source, use the Tutorial.XMLtoEDI:conditionalTruncate. This service performs the truncation if the source String is longer than the specified length. Otherwise, it leaves the source String as is. You can also use the wm.b2b.edi:pad service

to perform truncation. You might need to truncate characters because many EDI elements limit the number of characters that are allowed. `wm.b2b.edi:pad` services.

- To strip off control characters within the XML source String, use the `Tutorial.XMLtoEDI:replaceControlChar` service. This service enables you to replace the control characters with a character you specify. You might need to strip off character if you find new line (`\n`) characters within your XML document. These characters are not allowed within an EDI document because a new line can be used as a segment separator.
- To prevent redundant mapping, use a properties file. Rather than create new mapping services to make a simple change like changing a code because two different partners use different codes, you can place the proper codes to use in a properties file. For example, one partner might want you to use the code AP while the other wants ZZ. Your service can read the properties file using the `Tutorial.XMLtoEDI.properties:getProperties` service. If the code changes for a different trading partner, all you have to do is change the properties file without having to re-map the segment.

Forming EDI Documents and Sending Them Outbound

■ Overview	88
■ Before Creating the Service to Form an EDI Documents	88
■ Inputs to Your Service	88
■ Logic to Include in the Service to Form an EDI Document	89
■ Example	92

Overview

You can send documents to the Integration Server from internal applications (e.g., back-end systems) and map information from them into a standard EDI format document. The basic EDI engine of the EDI Module (i.e., the WmEDI package) provides tools that you need to create the EDI document.

The client that sends the internal-format document to the Integration Server should invoke a service that you create to form the EDI document and deliver it. This chapter provides information about how to create the service to form the EDI document.

To learn more about the basics about forming EDI documents to send outbound, see Chapter 2, "Using the EDI Module without Other webMethods Components" in the *webMethods EDI Module Concepts Guide*.

Before Creating the Service to Form an EDI Documents

Before you create the service to form an EDI document from an internal-format document create:

- The flat file schema that defines the structure of the EDI document that you are forming. The EDI Module uses the schema to create the EDI document from an IData object. For instructions, see "[Creating Flat File Schemas for EDI Documents](#)" on page 20.
- Optionally, the flat file schema that defines the structure of the internal-format document. This is needed if your client passes the internal-format document to your service in String format or as an InputStream. Your service uses the flat file schema as input to the `wm.b2b.edi:convertToValues` service to convert the internal-format document to an IData object and optionally validate the document's structure. If your service receives the document as an IData object, a flat file schema is not needed. Use the webMethods Developer to create the flat file schema. For more information, see the *Flat File Schema Developer's Guide*.
- Optionally, an IS document type for the structure of the internal-format document. This is needed if 1) your client passes the internal-format document to your service as an IData object, *and* 2) you want to validate the internal-format document before forming the EDI document. Your service uses the IS document type as input to the `pub.schema:validate` service, which performs the structure validation. For instructions about how to create an IS document type, see the *webMethods Developer User's Guide*.

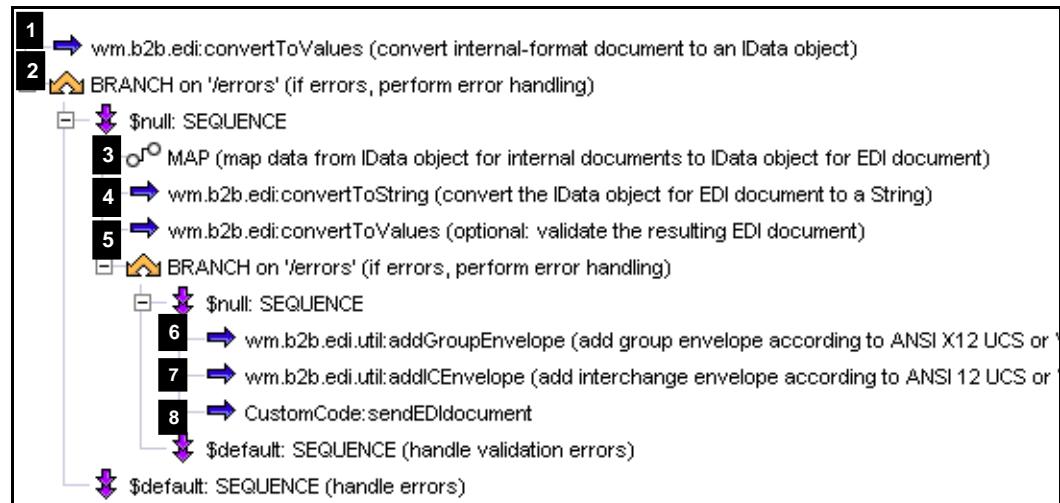
Inputs to Your Service

The inputs depend on the information that the client sends. At a minimum, it should provide the internal-format document.

Logic to Include in the Service to Form an EDI Document

The following shows sample code that includes the basic logic you would include to form an EDI document from an internal-format document and sent it outbound. For details about all of the built-in services that the sample uses, see the *webMethods EDI Module Built-In Services Reference*.

Sample code for forming an EDI document from an internal-format document



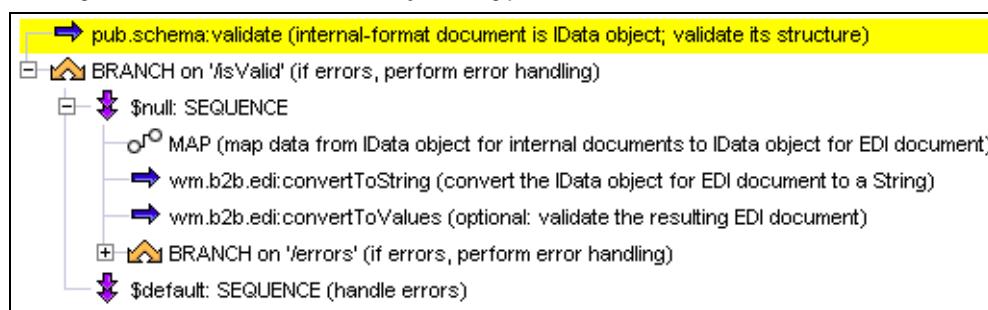
Flow operation	Description
1	<p>Invoke the <code>wm.b2b.edi:convertToValues</code> service to convert the incoming internal-format document that is either a String or InputStream into an IData object. If you want, you can set the input variables of the <code>convertToValues</code> service to have it validate the structure of the internal-format document.</p> <p>The inputs to the <code>convertToValues</code> service include the internal-format document and the flat file schema that defines the structure for the internal-format document. For backward compatibility, you can use an IS document type to define the structure of internal-format document rather than a flat file schema. However, it is recommended that you use flat file schemas.</p> <p>Note: If the internal-format document is passed to your service as an IData object, you can still validate its structure before forming the EDI document. See “Validating the Input Internal-Format Document When it is an IData Object” on page 91.</p>

Flow operation	Description
2	Add your own logic to handle errors that might result from executing the <code>convertToValues</code> service, for example, validation errors.
3	Map data from the internal-format document <code>IData</code> object into the EDI document <code>IData</code> object. Depending on the complexity of your mapping requirements, you might need to add more logic than a MAP flow operation, or create a separate service to perform the mapping. For more information about how to map, see Chapter 5, “Mapping Data to Form New Documents” .
4	Invoke the <code>wm.b2b.edi:convertToString</code> service to convert the EDI document from an <code>IData</code> object to String format. The inputs to the <code>convertToString</code> service include: <ul style="list-style-type: none">■ The <code>IData</code> object that contains the data for your EDI document. Map this <code>IData</code> object to the input variable <i>Values</i> of the <code>convertToString</code> service.■ The flat file schema for the EDI document. The <code>convertToString</code> service uses the flat file schema to determine how to form the EDI document. For backward compatibility, you can use an IS document type as input to the <code>convertToString</code> service rather than a flat file schema for files with delimited fields and records.
5	Optionally, invoke the <code>convertToValues</code> service against the EDI document to validate the structure of your final EDI document. The inputs to the <code>convertToValues</code> service include: <ul style="list-style-type: none">■ The EDI document. The output variable <i>string</i> from the <code>convertToString</code> service contains the EDI document. Map this to the input variable <i>edidata</i> of the <code>convertToValues</code> service.■ The flat file schema that defines the structure for the EDI document.
6	If needed, invoke the <code>wm.b2b.edi.util:addGroupEnvelope</code> service to add the group envelope to the EDI document. If you are creating a UN/EDIFACT EDI document, use the <code>wm.b2b.edi.util:addICEnvelopeEDIFACT</code> service. For more information, see “Adding UN/EDIFACT Envelopes” on page 91 .
7	Invoke the <code>wm.b2b.edi.util:addICEnvelope</code> service to add the interchange envelope to the EDI document. If you are creating a UN/EDIFACT EDI document, use the <code>wm.b2b.edi.util:addICEnvelopeEDIFACT</code> service. For more information, see “Adding UN/EDIFACT Envelopes” on page 91 .
8	Add your own logic or invoke a service that you create to send the EDI document outbound.

Validating the Input Internal-Format Document When it is an IData Object

The logic in the sample above assumes the internal-format document is passed to your service as a String or InputStream. In this situation, you can have the `wm.b2b.edi:convertToValues` service validate the document. If your client passes the internal-format document to your service as an IData object and you want to validate the internal-format document *before* converting it to an EDI document, you can replace the logic that invokes the `convertToValues` service with logic that uses the `pub.schema:validate` service, as shown below:

Validating internal-format that is an IData object using `pub.schema:validate`

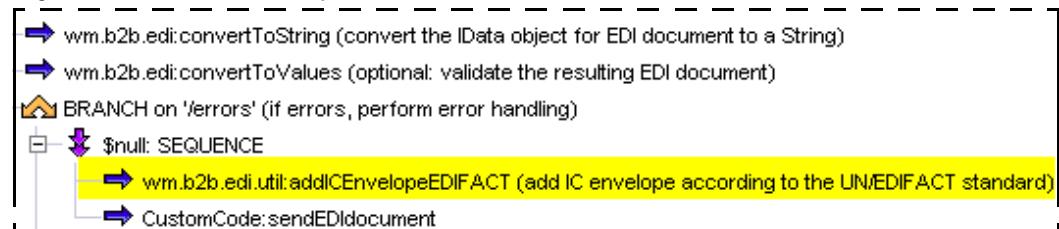


When you use the `pub.schema:validate` service, you must have an IS document type that defines the structure of the internal-format document. For more information about the `pub.schema:validate` service, see the *webMethods Integration Server Built-In Services Reference Guide*.

Adding UN/EDIFACT Envelopes

To add the appropriate group and interchange envelopes for a UN/EDIFACT EDI document, you invoke the `wm.b2b.edi.util:addICEnvelopeEDIFACT` service rather than invoke the `wm.b2b.edi.util:addGroupEnvelope` and `wm.b2b.edi.util:addICEnvelope` services, as shown below:

Logic to add UN/EDIFACT envelopes to the EDI document



For more information about the `wm.b2b.edi.util:addICEnvelopeEDIFACT` service, see the *webMethods EDI Module Built-In Services Reference*.

Checking Your Work

If you want to validate the resulting EDI document envelope and check compliance as a fail-safe measure, you can provide it as input to the `wm.b2b.edi:envelopeProcess` service or the `wm.b2b.edi.util:generateFA` service. For more information, see these services, see the *webMethods EDI Module Built-In Services Reference*.

Example

The `Tutorial.XMLtoEDI:processXMLSource` service in the `WmEDIsamples` package converts an XML document into an outbound EDI (ANSI X12) document. The `WmEDIsamples` package is located in the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>. The samples in this folder have been certified, meaning that they have been tested by webMethods.



Important! If you downloaded the `WmEDIsamples` package from the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>, you should delete the `WmEDIsamples` package before going into production.

Handling Large Documents

■ Overview	94
■ Configuring Large Document Handling	94
■ Creating Clients that Send Documents as InputStreams	97
■ Converting Documents to IData Objects Iteratively	98

Overview

If some or all of the documents that you need the EDI Module to process encounter problems because of memory constraints, you can do the following:

- Configure the EDI Module to handle “large” documents differently. The configuration properties that you set include defining a threshold size at which the EDI Module considers a document large. The EDI Module processes documents that are considered large in a different manner. Specifically, rather than keeping the large documents in memory, the EDI Module temporarily saves the large documents to local hard disk drive space (known as tspace). The built-in services provided with the EDI Module automatically recognize when a document is in tspace and perform their processing accordingly.
- Create clients that send documents to the Integration Server as InputStreams.
- Convert documents to IData objects iteratively by using the *iterator* variable in the `wm.b2b.edi:convertToValues` service to process a document a segment/section at a time.

To learn more about large document handling when using Trading Networks, see [Chapter 19, “Handling Large Documents When Using Trading Networks”](#) in this guide.

Configuring Large Document Handling

To configure the EDI Module to use large document handling, perform the following procedure to update the configuration in memory (so the changes take effect immediately) and in the `WmEDI/config/properties.cnf` file.

To configure the EDI Module for large document handling

- 1 Update the EDI Module large document settings:
 - a Open the Server Administrator if it is not already open.
 - b In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
 - c From the EDI Module home page, in the **Configuration** menu of the navigation panel, click **Configure Properties**.
 - d Click **Edit Properties Settings**.

- e Add or update the following property:

Property	Description
<i>EDIBigDocThreshold</i>	<p>Determines at what threshold size the EDI Module should consider a document to be large.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> -n: Specify a negative whole number to have the EDI Module consider no documents as large. The EDI Module processes all documents in the traditional manner, reading the document content into memory during processing. 0: Specify zero to have the EDI Module consider all documents large. Because all documents that the EDI Module receives contain more than 0 bytes, the EDI Module uses large document handling for all documents. n: Specify a positive whole number to indicate the number of bytes over which the EDI Module considers a document to be large. For example, if you specify the following, the EDI Module considers all documents greater than 1,000,000 bytes to be large: <pre>EDIBigDocThreshold=1000000</pre> <p>Default Value:</p> <ul style="list-style-type: none"> -1: Indicates that the EDI Module considers no documents to be large.

- f Click Save Changes.
- 2 Update the following Trading Networks large document settings:
- From the Server Administrator, select **Settings > Extended** from the menu.
 - Add or update the following Trading Networks properties:

<u>Property</u>	<u>Description</u>
<i>watt.server.tspace.location</i>	<p>Specifies the absolute directory path of the hard disk drive space in which the Integration Server is to temporarily store large documents rather than keep them in memory. Each file that the Integration Server stores in this directory is given the name DocResxxxx.dat, where xxxx is a value that can vary in length and character. Specify the absolute directory path to a directory on the same machine as the Integration Server. The default value is JVM's temporary directory (i.e., the value of java.io.tmpdir).</p> <p>Example: If you want Integration Server to use the LargeDocTemp directory on your D drive, specify the following:</p> <pre>watt.server.tspace.location=D:\\LargeDocTemp</pre>
<i>watt.server.tspace.max</i>	<p>Specifies the maximum number of bytes that can be stored at any one time in the hard disk drive space that you defined using the <i>watt.server.tspace.location</i> property. If the Integration Server attempts to write a large document to the hard disk drive space that will cause the number of bytes you specify to be exceeded, an error message is displayed on the server console, and the document is not stored. Specify a positive whole number of bytes. The default value is 52,428,800 bytes (50 MB).</p> <p>Example: To set the maximum number of bytes that can be stored to 30,000,000 bytes, specify the following:</p> <pre>watt.server.tspace.max=30000000</pre> <p>Note: The size of the hard disk drive space for temporarily saving documents will vary based on the number of documents that you process concurrently and the size of the documents that you process. For example, if your typical concurrent document load is 10, you would need a hard disk drive space that is 10 to 15 times the combined size of the documents being processed concurrently.</p>

-
- 3 Click Save Changes.

When is the EDI Module Configuration File Checked?

The EDI Module checks the EDI Module configuration file during the following actions:

- When the `wm.b2b.edi:envelopeProcess` service is invoked to process an inbound EDI document. The envelopeProcess service leaves the transaction set data within an EDI document as unparsed. These unparsed sections of the document are known as undefined segments. The envelopeProcess service uses `EDIBigDocThreshold` property to determine whether the undefined segment is large.
- When the `wm.b2b.edi:convertToValues` service is invoked to convert a String or `InputStream` to an `IData` object. The convertToValues service uses a flat file schema to determine how to parse the document. When the convertToValues service encounters sections of the EDI document for which the flat file schema does not define the structure, that section of the document is considered an undefined segment. The convertToValues service uses the `EDIBigDocThreshold` property to determine whether the undefined segment is large.

If an undefined segment is considered large, the envelopeProcess and convertToValues services write the undefined segment to tspace and store the pointer to the undefined segment (called a reservation ID) in the element `_RID_`. The envelopeProcess service places the `_RID_` element in its output `IData` object (`Values`) after the element for the corresponding transaction header. The convertToValues service places the `_RID_` element in its output `IData` object (`EDIVValues`) after the last identified segment.

If the undefined segment is *not* considered large, the services write the undefined segment to the element `unDefData` and place the `unDefData` element the output `IData` object (instead of an `_RID_` element).

For more information about the envelopeProcess and convertToValues services, see the *webMethods EDI Module Built-In Services Reference*.

Creating Clients that Send Documents as InputStreams

When you create clients that send documents to the EDI Module for processing, be sure to send those documents as an `InputStream` rather than a String. To ensure your client that sends an EDI document to the Integration Server as an input stream, use the content type `application\EDIstream`. For more information about creating clients, see [Chapter 2, "Creating Clients that Send EDI Documents to the Integration Server"](#).

Converting Documents to IData Objects Iteratively

When you need to convert an EDI document to an IData object, you use the `wm.b2b.edi:convertToValues` service. If you are converting a large EDI document to an IData object, you should process the document iteratively, segment by segment. A large EDI document is defined as an EDI document that contains many interchanges, many groups, and many transactions, or as a transaction that contains many line items.

To use the `convertToValues` service to process an EDI document iteratively, you specify the *iterator* input variable of the `convertToValues` service as `true`. Setting *iterator* to `true` causes the `convertToValues` service to process just a segment or a group of segments of the document at a time. The `convertToValues` service determines how many segments to process based on the flat file structure information in the flat file schema for the EDI document.

For more information about processing EDI documents iteratively, see “[Processing the Document Iteratively Segment by Segment](#)” on page 47.

Using EDI Module with webMethods Trading Networks

■ Before You Can Use Trading Networks to Process EDI Documents	101
■ Defining Partner Information (ANSI X12 and UN/EDIFACT)	109
■ Defining Partner Information (TRADACOMS)	145
■ Defining Control Number Information for Partners	163
■ Creating Clients that Send EDI Documents to Trading Networks	185
■ Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks ...	193
■ Processing Inbound TRADACOMS Documents using Trading Networks	219
■ Optional Inbound Processing When Using Trading Networks	237
■ Forming EDI Documents to Send Outbound When Using Trading Networks	283
■ Batching EDI Documents	309
■ Retrieving and Delivering EDI Documents from and to VANs	335
■ Handling Large Documents When Using Trading Networks	343
■ Including Documents in a Business Process	345

Before You Can Use Trading Networks to Process EDI Documents

- Overview 102
- Setting Up Items Required for Structural Validation and Conversion 103
- Installing TN Document Types and Creating Flat File Schemas 104
- Configuring the EDI Module To Support New Versions of EDI Standards 107

Overview

The following lists the tasks to perform before you can process EDI documents when you are using webMethods EDI Module (EDI Module) with webMethods Trading Networks (Trading Networks).

- Set up the items required for parsing and validating the structure of EDI documents, and converting documents from EDI format to the format required by your internal applications and vice versa. To set up these items, you need to:
 - Configure how you want format services to convert field values in documents from EDI format to internal format, and vice versa.
 - Specify how to associate format service to fields defined in a flat file schema for an EDI document.
 - Create flat file schemas for EDI documents and your internal-format documents.
- For details, see “[Setting Up Items Required for Structural Validation and Conversion](#)” on page 103 .
- Install TN document types for the types of EDI documents you want to process. For details, see “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104.
- Configure the EDI Module to support a new version of any EDI standard that the EDI Module supports. For details, see “[Configuring the EDI Module To Support New Versions of EDI Standards](#)” on page 107.
- Define information about partners. For details, see [Chapter 9, “Defining Partner Information \(ANSI X12 and UN/EDIFACT\)”](#).
- Define settings for inbound control number validation. For details see [Chapter 11, “Defining Control Number Information for Partners”](#).
- If you want to use EDIINT to transport your EDI documents, see the *webMethods EDIINT Module User’s Guide* for tasks to prepare to use EDIINT.

To learn more about:

- Flat file schemas in general and how format services relate to them, including how to specify format services for fields in flat file schemas and dictionaries, see “Format Services” in the “Creating and Editing Flat File Schemas and Dictionaries” chapter of the *Flat File Schema Developer’s Guide*.

To learn more about (continued):

- How EDI Module works with Trading Networks and uses TN document types, profiles, and EDITPAs, see Chapter 3, "Using the EDI Module with Trading Networks" of the *webMethods EDI Module Concepts Guide*.
- TN document types, profiles, and TPAs and how Trading Networks uses them, see the *webMethods Trading Networks Concepts Guide*.

Setting Up Items Required for Structural Validation and Conversion

The EDI Module uses flat file schemas to parse and validate the structure of inbound EDI documents and to convert documents from EDI format to internal application format and vice versa. The EDI Module automatically installs the flat file schemas for you when you install TN document types. For more information, see ["Installing TN Document Types and Creating Flat File Schemas" on page 104](#).

However, there are still some flat file-related tasks that you might need to set up. The following lists the sections in this guide that describe the setup that you might need to perform:

- ["Configuring How the Format Services Convert Field Values" on page 25](#)
- ["Associating the EDI Format Services to EDI Data Types" on page 29](#)
- ["Creating Flat File Schemas for Internal-Format Documents" on page 24](#)

Installing TN Document Types and Creating Flat File Schemas

When Trading Networks receives a document, it matches the inbound document against its TN document types to determine the type of document it received. This is referred to as document recognition.

For Trading Networks to be able to recognize EDI documents, you must install a TN document type corresponding to each EDI transaction set or each TRADACOMS file (of a specific standard and version) that you will be exchanging. For example, if you plan to exchange 850's (purchase orders), 855's (purchase order acknowledgments), and 997's (functional acknowledgments), you would install the three TN document types corresponding to a specific version and standard of each transaction set. Or, if you plan to exchange three TRADACOMS File document types, you would install those three TN document types corresponding to a specific version and standard of each file.

When you install a TN document type, the EDI Module automatically creates a flat file schema for the same EDI transaction set or TRADACOMS file. Trading Networks stores flat file schemas in the EDIFFSchema folder in the WmEDIforTN package, using the naming conventions described in [“Flat File Schema Namespace Conventions” on page 105](#).

For each transaction set or file you want to exchange with trading partners, perform the following procedure to install TN document types and flat file schemas for that transaction set or file.

To install TN document types and flat file schemas for an EDI transaction set or file

- 1 Open the Server Administrator if it is not already open.
- 2 In the Solutions menu of the navigation panel, click EDI. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the Doc Exchange menu of the navigation panel, click **Install TN Document Types**.
- 4 From the Standard, Version, and Transaction Set drop-down lists, select the appropriate EDI standard, version, and transaction set (or TRADACOMS file), respectively.



Note: In TRADACOMS, the transmission, batch, and file document types each have a version. The version you are specifying here is the version of the TRADACOMS file document type, e.g., v4.

-
- 5 Click **Add Document Type Definition to Trading Networks**. The EDI Module displays a message stating that the TN document type was installed.

Flat File Schema Namespace Conventions

When you install a TN document type, the EDI Module uses a SEF file to automatically create a flat file schema for the same EDI transaction set or TRADACOMS file. Trading Networks stores flat file schemas in the EDIFFSchema folder in the WmEDIforTN package, using the namespace conventions described below.

Flat File Schema Namespace Conventions (ANSI X12 and UN/EDIFACT)

For ANSI X12 and UN/EDIFACT (and all supported sub-standards), Trading Networks uses the following namespace convention for flat file schemas:

`EDIFFSchema.Standard.VVersion.TTransactionSet`

For example, when you install the TN document type for the 4010 version of an ANSI X12 850 (*Standard* is X12, *Version* is 4010, *TransactionSet* is 850), Trading Networks creates the following flat file schema in the WmEDIforTN package:

`EDIFFSchema.X12.V4010:T850`

When you install the TN document type for ORDERS (of the UN/EDIFACT standard, 99A version), Trading Networks creates the following flat file schema in the WmEDIforTN packages:

`EDIFFSchema.UNEDIFACT.V99A:TORDERS`

 **Note:** If a transaction in a SEF file contains multiple consecutive HL segments with the same name, the package combines the HL segments into one HL segment in the flat file schema, thus enabling it to be parsed correctly.

Flat File Schema Namespace Conventions (TRADACOMS)

For each TRADACOMS File document type, Trading Networks installs up to four flat file schemas, using the following namespace convention:

`EDIFFSchema.Tradacoms.Vversion.Tname:MheaderMessage`
`EDIFFSchema.Tradacoms.Vversion.Tname:MdetailMessage`
`EDIFFSchema.Tradacoms.Vversion.Tname:MVATMessage`
`EDIFFSchema.Tradacoms.Vversion.Tname:MtrailerMessage`

where:

- *version* represents the version of the TRADACOMS File document type (for example, v2).
- *name* represents the name of the TRADACOMS File document type (for example, TLPRHDR).
- *headerMessage*, *detailMessage*, *VATMessage*, and *trailerMessage* are derived from the MHD0201.

For example: `EDIFFSchema.Tradacoms.v2.TLPRHDR:MLPRHDR`

For information about the attributes stored for TRADACOMS document types, see ["Attributes Stored for TRADACOMS Document Types" on page 106](#).

Attributes Stored for TRADACOMS Document Types

Trading Networks supports the TRADACOMS document types Transmission, Batch, and File. This table shows which attributes are stored for each document type.

Attribute	Transmission Type	Batch Type	File Type
Application Reference	yes	derived from transmission	derived from transmission
Priority Code	yes	derived from transmission	derived from transmission
Detail Message Count	no	no	yes
Has VAT Message	no	no	yes
Has Reconciliation Message	yes	no	no
Is Multiple Envelope	yes	no	no
Version	yes (only valid value is 1)	derived from transmission	no (determined by document type)

After Creating the Flat File Schemas

You might need to customize the flat file schemas according to your specific validation needs. For example, you might want to customize the flat file schema to validate for fewer constraints, or you might want to modify the valid values.

For information about how to work with flat files schemas, see the *Flat File Schema Developer's Guide*.

Configuring the EDI Module To Support New Versions of EDI Standards

You can configure the EDI Module to support a new version of any EDI standard that the EDI Module supports, except TRADACOMS.

To add a new version of an EDI standard

- 1 Obtain the appropriate SEF file for the new version of the EDI standard.

The SEF file's .INI section must specify the EDI standard and version, and the file must contain all the transaction sets that the version supports.

- 2 In the webMethods Developer's navigation panel, run the following built-in service located in the WmEDIforTN package: `wm.b2b.edtn.util.VersionSupport:addNewEDIVersion`.

This service creates a new TN document type file in the config directory of the WmEDIforTN package. It also adds the new version to the WmEDI home page. The new version will be available when installing TN document types.

Specify the following input parameters for the service:

- **SEF fileName:** The fully qualified path of the SEF file.
- **Replace?:** Specifies whether to overwrite an existing SEF file of the same name and its associated TN document type file. Specify `no` (the default) or `yes`.

Note: The service does not update the envelope segments structure, nor does it update the validation for individual fields. However, the service does ensure that the following fields are updated in the IS document type `wm.b2b.edtn.rec:EDIHEADERS` so that TN will recognize the new version at validation time:

Standard	Code list for field	Description
ANSI X12, VICS, UCS	ISA/ISA12	Version in the Interchange header.
	ISA/GS/GS08	Version in the Group header.

 **Note:** If the new version of the EDI standard introduces changes to the structure of the 997 transaction set, these changes will not be reflected in the functional acknowledgments created by the `wm.b2b.edi:generateFA` service.

For complete information about the `wm.b2b.edtn.util.VersionSupport:addNewEDIVersion` service, see the *webMethods EDI Module Built-In Services Reference*.

- 3 Reload the WmEDIforTN package.

- 4 Install the new TN document type and its flat file schema for the transaction set as follows:
 - a Open the Server Administrator if it is not already open.
 - b In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
 - c From the EDI Module home page, in the **Doc Exchange** menu of the navigation panel, click **Install TN Document Types**.
 - d From the **Standard**, **Version**, and **Transaction Set** drop-down lists, select the EDI standard, the new version, and the version's transaction set.
 - e Click **Add Document Type Definition to Trading Networks**.
 - f In the pop-up that appears, provide values for the **SEF File Name**, **Standard**, and **Version** fields and click **OK**.

The system creates the following:

The system creates ...	In the following directory ...
TN document type	<code>WmEDIforTN\config\EDI_Standard\docType.xml</code> where <i>EDI_Standard</i> is X12, UNEDIFACT, ODETTE, EANCOM, VICS, or UCS.
SEF file	<code>webMethods6\IntegrationServer\packages\WmEDI\pub\SEFS\EDI_Standard\version.sef</code> Note: For UN/EDIFACT, ODETTE, and EANCOM, the system adds a CTRL message in the .SETS section of the SEF file.



Important! After you install a new version of an EDI standard and have installed a TN document type from that version of the standard, *do not re-install that version of the EDI standard*. If you re-install them, the internal ID of the document type will change, and any references to that ID will be wrong. In addition, any Trading Networks processing rules that refer to the document type will be wrong as well.

Defining Partner Information (ANSI X12 and UN/EDIFACT)

- Overview 110
- Defining Trading Networks Profiles 113
- Defining EDI Trading Partner Agreements 116

Overview

To process documents when you are using webMethods Trading Networks (Trading Networks) with the webMethods EDI Module (EDI Module), you must define Trading Networks profiles and trading partner agreements (EDITPAs) for the partners that will be identified as the senders and receivers in the EDI documents you expect to process. How you define information for your partners depends on whether you want the EDI Module to use standard or non-standard processing.

Using Standard or Non-Standard Processing

When the EDI Module processes an EDI document, it processes one interchange of the EDI document at a time. An interchange can contain groups and transaction sets. The EDI Module can process the interchange, groups, and transaction sets using either standard or non-standard processing.

- For **standard processing**, the EDI Module processes the interchange, all its groups, and all its transaction sets using settings that you define for the interchange sender/receiver pair. To set up standard processing, you use standard Trading Networks objects.
- For **non-standard processing**, you can specify different processing settings for each group within the interchange. The EDI Module processes the groups and the transaction sets within each group using the settings you define for the group sender/receiver pairs. To set up non-standard processing, in addition to using Trading Networks objects, you must define information from the EDI Module home page, which includes interchange sender/receiver pair information.



Important! The chapters in this guide describe how to use standard processing. If you want to use non-standard processing, see [Appendix A, "Non-Standard Processing"](#) for information about what you must do differently to use non-standard processing. If a task described in one of the chapters of this guide should be performed differently for non-standard processing, this guide includes a reference to the appropriate section in [Appendix A, "Non-Standard Processing"](#) and the reference uses the eye-catcher **Non-standard**. If you are using standard processing, you can ignore these references.

To learn more about how the EDI Module processes inbound documents, see Chapter 3, "Using the EDI Module with Trading Networks" of the *webMethods EDI Module Concepts Guide*.

Advantages and Disadvantages of Standard vs Non-Standard Processing

The table below lists the advantages and disadvantages of using standard and non-standard processing:

Standard Processing	Non-Standard Processing
Advantages	Advantages
<ul style="list-style-type: none"> ■ Information you define is in profiles and EDITPAs, so management of all information can be done through the Trading Networks Console. ■ webMethods recommends this approach. 	<ul style="list-style-type: none"> ■ You only need to define profiles for partners (senders/receivers) at the group level through the Trading Networks Console. You do not need to define profiles for partners at the interchange level. Therefore fewer profiles to set up and maintain.
Disadvantages	Disadvantages
<ul style="list-style-type: none"> ■ You need to define profiles for all partners (senders/receivers) at both the interchange and the group level. 	<ul style="list-style-type: none"> ■ You must maintain information for interchange level sender/receiver pairs through the EDI Module home page, as well as maintain EDITPAs through the Trading Networks Console. ■ You must maintain information about the sender/receiver pairs at the group level that are associated with each interchange level. This information is maintained through the EDI Module home page. ■ If you want to save a copy of the interchange-level document in the Trading Networks database, the document will be saved with the sender and receiver set to unknown. Because you do not create profiles for senders and receivers at the interchange level, Trading Networks is unable to determine the sender and receiver. ■ You <i>cannot</i> use the batching feature of the EDI Module.

Partner Information You Need to Define

The following table lists the information you must define for interchange and group sender receiver pairs that will be in the EDI documents you expect to process:

Non-standard	When you are using non-standard processing, the information you need to define for partners is different. See “ Defining Partner Information ” on page 386 in Appendix A, “Non-Standard Processing” . For more information about the difference between standard and non-standard processing, see “ Using Standard or Non-Standard Processing ” on page 110 .
For...	Define...
Interchange sender/receiver pairs	<ul style="list-style-type: none"> ■ Trading Networks profiles for each interchange sender and receiver. For more information, see “Defining Trading Networks Profiles” on page 113.
Group sender/receiver pairs	<ul style="list-style-type: none"> ■ Trading Networks profiles for each group sender and receiver. For more information, see “Defining Trading Networks Profiles” on page 113.
Tailoring how the EDI Module processes documents	<ul style="list-style-type: none"> ■ Default EDITPA that defines the settings that you want to use for most partner pairs. ■ Partner-specific EDITPA for each interchange sender/receiver pair for which you want to override the settings in the default EDITPA. <p>For more information, see “Defining EDI Trading Partner Agreements” on page 116.</p>
How the EDI Module validates inbound control numbers	<ul style="list-style-type: none"> ■ Whether to validate control numbers. ■ Actions the EDI Module is to take when it encounters an invalid control number. ■ Control number validation settings (i.e., control number cap, minimum, increment, and window). <p>For more information, see Chapter 11, “Defining Control Number Information for Partners”,</p>

To learn more about:

- How EDI Module works with Trading Networks and uses profiles and EDITPAs, see Chapter 3, "Using the EDI Module with Trading Networks" of the *webMethods EDI Module Concepts Guide*.
- Profiles and TPAs and how Trading Networks uses them, see the *webMethods Trading Networks Concepts Guide*.

Defining Trading Networks Profiles

To identify the trading partners with whom you want to exchange documents, set up profiles. You must define profiles for:

- Your own corporation if you have not already done so. This is referred to as the Enterprise profile in Trading Networks.
- Your trading partners at the interchange *and* group level.

Non-standard

When you are using non-standard processing, you need to create profiles only for partners at the group level. See "["Defining Trading Networks Profiles" on page 387](#)" in [Appendix A, "Non-Standard Processing"](#). For more information about the difference between standard and non-standard processing, see "["Using Standard or Non-Standard Processing" on page 110](#)".

When you define the profiles, you need to specify the external IDs your partners use in their documents. The external IDs correspond to the standard EDI ID qualifiers. Trading Networks does not provide external IDs for all EDI ID qualifiers; however, the EDI Module provides a way for you to add them to Trading Networks. Before creating the profiles you might want to add additional external ID types to Trading Networks that correspond to the EDI ID qualifiers. See "["Adding External ID Types for EDI ID Qualifiers" on page 114](#)".

To create profiles

You create profiles using the Trading Networks Console. For steps to create profiles, see the chapter about creating profiles in the *webMethods Trading Networks User's Guide*.

Adding External ID Types for EDI ID Qualifiers

An external ID type indicates the method a corporation uses to identify itself within documents and enables you to identify the partner with whom you are exchanging a document. For example, a corporation might use a D-U-N-S number as an external ID type, which is identified by the EDI ID qualifier "1" or "01." When you define the profile, you select an external ID type and supply the value the partner uses. For example, if the corporation uses a D-U-N-S number, the value is the corporation's D-U-N-S number, such as "123456789."

Trading Networks uses the external ID information in a profile to determine the sender and receiver of a document. EDI ID qualifiers and their corresponding values identify the sender and receiver of the EDI document. The EDI ID qualifiers and their corresponding values are contained in the interchange and group envelope headers. The following shows an example of an ANSI X12 interchange envelope header.

ISA*00* *00* *01*123456789 *ZZ*987654 *990423*1810*U*00200*000005334*

Interchange Sender ID Qualifier = 01
Interchange Sender ID= 123456789

Interchange Receiver ID Qualifier = ZZ
Interchange Receiver ID= 987654

The EDI ID qualifier “01” corresponds to a D-U-N-S number (which is the Trading Networks external ID type, DUNS). The EDI ID qualifier “ZZ” corresponds to the Trading Networks external ID type Mutually-Defined. Trading Networks automatically provides the DUNS and Mutually-Defined external ID types. However, Trading Networks does not provide external ID types for all EDI ID qualifiers. You can add to Trading Networks external ID types that correspond to standard EDI ID qualifiers. For all EDI ID qualifiers and their external ID type equivalents, see the EDI Standards documentation for your EDI standard and version.

For each additional external ID type you want to add to Trading Networks for other EDI ID qualifiers, perform the following procedure.

To add external ID types to Trading Networks for EDI ID qualifiers

- 1 Open the Server Administrator if it is not already open.
 - 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
 - 3 From the EDI Module home page, in the **Partner Set Up** menu of the navigation panel, click **Add External ID Types**.
 - 4 From the **External ID Type** drop-down list, select the EDI ID qualifier for which you want to add an external ID type.



Note: Some ID qualifiers have different meanings in different EDI standards and versions. If you want to override an ID qualifier, see “[Adding and Overriding EDI ID Qualifiers](#)” on page 115.

- 5 Click Add External ID Type to Trading Networks. The EDI Module displays a message stating that the external ID type was installed.



Note: To see the new external ID types in Trading Networks, restart the Trading Networks Console.

Adding and Overriding EDI ID Qualifiers

Some EDI ID qualifiers have different meanings in different EDI standards and versions. For example:

- For UN/EDIFACT, the value of the ID qualifier code 30 is "ISO 6523"
- For ANSI X12, the value of the ID qualifier code 30 is "Federal Tax ID"

The EDI Module considers the value of the ID qualifier code 30 to be "ISO 6523". However, you can create a service to override (or add) any ID qualifier code value (for example, you can override "ISO 6523" with "Federal Tax ID"), as described below.



To create a custom flow service to add or override an ID qualifier

- 1 In any user-defined package, create a service that overrides the value of the ID qualifier code. This service *must* be named `custom.EDIIDMapping:addEDIIDMapping`. The service should include the String input parameter *qualifier* and the String output parameter *idType*. For a sample service, see the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>.
- 2 In your flow, create the `custom.EDIIDMapping:addEDIIDMapping` service.
- 3 Add to the service the String input parameter *qualifier* and the String output parameter *idType*.
- 4 In the service, branch on *qualifier* and add a map step. Label the map step with the ID qualifier that you expect to receive (for example, 30).
- 5 In the map step, hard code the value of the output *idType* with any value, such as "User Defined 1".

Defining EDI Trading Partner Agreements

An EDI trading partner agreement (EDITPA) is a set of variables that you provide to tailor how the EDI Module exchanges documents between two trading partners. The EDI Module supports partner-specific EDITPAs and a single default EDITPA.

- A **partner-specific EDITPA** has a specific sender and receiver associated with it and contains variables that the EDI Module uses only when processing documents sent by the specified sender and to be received by the specified receiver.
- A **default EDITPA** has a sender and receiver set to “unknown.” It contains variables used by all trading partners when partner-specific information is not available.

During processing, the EDI Module first attempts to find values in the partner-specific EDITPA for the sender/receiver pair of a document. If a partner-specific EDITPA does not exist for the sender/receiver pair or the value in the partner-specific EDITPA is null or empty, the EDI Module uses the value from the default EDITPA.

Defining the Default EDITPA

The settings in the default EDITPA should meet the requirements of the majority of your trading partner relationships.

The first time the WmEDIforTN package is loaded into the Integration Server, the EDI Module automatically creates the default EDITPA in Trading Networks. You can modify the default EDITPA when the **Agreement Status** is “Agreed” and the **Data Status** is “Modifiable” or when the **Agreement Status** is “Proposed”.



Note: You should not disable the default EDITPA. If you do disable the default EDITPA, the EDI Module will continue to use it.

To modify the default EDITPA

You modify the default EDITPAs using the Trading Networks Console. For steps to modify EDITPAs, see the chapter about trading partner agreements in the *webMethods Trading Networks User’s Guide*.

When you modify the default EDITPA, be sure to keep the settings listed in the table below, which you can view from the Trading Networks Console **Agreement Details** screen:

For this Agreement Details screen field	Keep this setting
Sender	Unknown
Receiver	Unknown

For this Agreement Details screen field	Keep this setting
Agreement ID	EDITPA
IS Document Type	<p>wm.b2b.editn.TPA:EDITPA</p> <p>You can modify the values for the variables in the wm.b2b.editn.TPA:EDITPA IS document type. To set the values for the variables in the wm.b2b.editn.TPA:EDITPA IS document type, see the description of the variables in “wm.b2b.editn.TPA:EDITPA IS Document Type” on page 118.</p>

Defining a Partner-Specific EDITPA

You only need to create partner-specific EDITPAs if you have one or more sender/receiver pairs that require settings that are different from those you specify in the default EDITPA. When creating a partner-specific EDITPA, you have to specify only the information that is different from the defaults. Define partner-specific EDITPAs for interchange-level sender/receiver pairs.



Note: You can disable partner-specific EDITPAs. When you disable a partner-specific EDITPA, the EDI Module functions as if the partner-specific EDITPA does not exist. That is, the EDI Module uses the values in the default EDITPA.

Non-standard

When you are using non-standard processing, define partner-specific EDITPAs for group-level sender/receiver pairs. See “[Defining a Partner-Specific EDITPA](#)” on page 388 in [Appendix A, “Non-Standard Processing”](#). For more information about the difference between standard and non-standard processing, see “[Using Standard or Non-Standard Processing](#)” on page 110.



To create a partner-specific EDITPA

You create a partner-specific EDITPA using the Trading Networks Console. You can create a partner-specific EDITPA in one of the following three ways:

- Duplicate the default EDITPA and change variable values.
- Duplicate another similar partner-specific EDITPA and change variable values.
- Create an EDITPA from scratch.

For instructions on how to create TPAs (either by duplication or from scratch), see the chapter about trading partner agreements in the *webMethods Trading Networks User’s Guide*.

When creating a partner-specific EDITPA, specify the following on the Trading Networks Console Agreement Details screen:

For this Agreement Details screen field	Specify
Sender	The name of the sender from the partner-specific sender/receiver pair.
Receiver	The name of the receiver from the partner-specific sender/receiver pair.
Agreement ID	EDITPA
IS Document Type	wm.b2b.editn.TPA:EDITPA To set the values for the variables in wm.b2b.editn.TPA:EDITPA, see the description of the variables in “wm.b2b.editn.TPA:EDITPA IS Document Type” on page 118 .
Initialization Service	Leave blank or specify a service that you created to define the default values for the wm.b2b.editn.TPA:EDITPA IS document type. webMethods recommends that you do <i>not</i> use the initialization service (wm.b2b.editn.TPA:initService) that is provided with the EDI Module. The wm.b2b.editn.TPA:initService service is used to populate the values of the default EDITPA.

wm.b2b.editn.TPA:EDITPA IS Document Type

The descriptions and default values for the variables of the wm.b2b.editn.TPA:EDITPA IS document type are listed below. At run time, the EDI Module uses a default value when the value for a variable is neither specified in a partner-specific EDITPA nor the default EDITPA. The default values are also the values that the wm.b2b.editn.TPA:initService initialization service sets in the default EDITPA when the EDI Module initially creates it.

Review variable descriptions to determine the value you want to specify in the default or a partner-specific EDITPA. When creating partner-specific EDITPAs, keep in mind that you should only specify values for the variables for which you want to override the default EDITPA value and leave values blank for the variables for which you want to use the defaults.



Tip! It is helpful to have an understanding about how the EDI Module processes EDI documents to understand how the EDI Module uses the variables in the EDITPA. If you have not already done so, read Chapter 3, "Using the EDI Module with Trading Networks" in the *webMethods EDI Module Concepts Guide*.

<i>splitOption</i> EDITTPA Variable	Default: Transaction								
<p>The <i>splitOption</i> variable indicates how you want the EDI Module to split an interchange segment within an EDI document. The EDI Module creates the following types of documents from an interchange segment based on the value of the <i>splitOption</i> variable:</p> <ul style="list-style-type: none"> ■ Interchange documents that contain the single interchange envelope along with its group segments and transaction sets. ■ Group documents that contain a single group segment along with its transaction sets. ■ Transaction documents that contain a single transaction set. <p>If you want to perform processing on the transactions in an inbound document, set <i>splitOption</i> to <code>Transaction</code> or <code>Group</code>. If you are sending the inbound EDI document through Trading Networks to simply deliver it to a destination without processing individual transactions, set the <i>splitOption</i> to <code>Interchange</code>.</p>									
Interchange	<p>The following table lists the possible values for the <i>splitOption</i> variable and the types of documents that the EDI Module creates for each value.</p> <table border="1"> <thead> <tr> <th><i>splitOption</i> Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Interchange</td><td> <p>The EDI Module creates only the Interchange document.</p> <p>Note: If <i>splitOption</i> is <code>Interchange</code> and the <code>FAReconciliation</code> EDITPA variable is set to <code>true</code>, the EDI Module will split the document at the Group level. For more information, see “FAReconciliation EDITPA Variable” on page 136.</p> <p>Non-standard</p> <p>When you are using non-standard processing and you specify <code>Interchange</code>, the EDI Module will split the document at the Group level. See “wm.b2b.edtn.TPA:EDITPA IS Document Type” on page 388 in Appendix A, “Non-Standard Processing”.</p> </td></tr> <tr> <td>Group</td><td> <p>The EDI Module creates the Interchange document and a Group document for each group segment in the interchange segment.</p> </td></tr> <tr> <td>Transaction</td><td> <p>The EDI Module creates the Interchange document, a Group document for each group segment in the interchange segment, and a Transaction document for each transaction set in the interchange segment.</p> <p>This is the initial value for the <i>splitOption</i> variable in the default EDITPA.</p> </td></tr> </tbody> </table>	<i>splitOption</i> Value	Description	Interchange	<p>The EDI Module creates only the Interchange document.</p> <p>Note: If <i>splitOption</i> is <code>Interchange</code> and the <code>FAReconciliation</code> EDITPA variable is set to <code>true</code>, the EDI Module will split the document at the Group level. For more information, see “FAReconciliation EDITPA Variable” on page 136.</p> <p>Non-standard</p> <p>When you are using non-standard processing and you specify <code>Interchange</code>, the EDI Module will split the document at the Group level. See “wm.b2b.edtn.TPA:EDITPA IS Document Type” on page 388 in Appendix A, “Non-Standard Processing”.</p>	Group	<p>The EDI Module creates the Interchange document and a Group document for each group segment in the interchange segment.</p>	Transaction	<p>The EDI Module creates the Interchange document, a Group document for each group segment in the interchange segment, and a Transaction document for each transaction set in the interchange segment.</p> <p>This is the initial value for the <i>splitOption</i> variable in the default EDITPA.</p>
<i>splitOption</i> Value	Description								
Interchange	<p>The EDI Module creates only the Interchange document.</p> <p>Note: If <i>splitOption</i> is <code>Interchange</code> and the <code>FAReconciliation</code> EDITPA variable is set to <code>true</code>, the EDI Module will split the document at the Group level. For more information, see “FAReconciliation EDITPA Variable” on page 136.</p> <p>Non-standard</p> <p>When you are using non-standard processing and you specify <code>Interchange</code>, the EDI Module will split the document at the Group level. See “wm.b2b.edtn.TPA:EDITPA IS Document Type” on page 388 in Appendix A, “Non-Standard Processing”.</p>								
Group	<p>The EDI Module creates the Interchange document and a Group document for each group segment in the interchange segment.</p>								
Transaction	<p>The EDI Module creates the Interchange document, a Group document for each group segment in the interchange segment, and a Transaction document for each transaction set in the interchange segment.</p> <p>This is the initial value for the <i>splitOption</i> variable in the default EDITPA.</p>								

GSRouting/routingMode EDITPA Variable

Default: OFF

The *GSRouting* variables indicate what you want the EDI Module to use for the sender and receiver that it puts in the Interchange, Group, and Transaction documents that it creates based on the *splitOption* variable in the EDITPA.

The value you select for the *GSRouting* variables affects:

- How you define criteria for Trading Networks processing rules that you create for the Interchange, Group, and Transaction documents. You can use the sender and receiver as criteria that Trading Networks uses to select processing rules.
- The external ID types and values you define in the profiles for your partners. For Trading Networks to match the sender and/or receiver criteria in a processing rule, you must have a profile for your partner that contains an external ID type and value that matches the value that EDI Module will put in the Interchange, Group, and Transaction documents.

For more information about creating processing rules and defining external IDs in profiles, see the *webMethods Trading Networks User's Guide*.

The *routingMode* variable indicates where the EDI Module obtains the sender and receiver for each type of document. The following lists the possible values for the *routingMode* variable. The examples use the following headers:

```
ISA*00***00* *01*123456789 *ZZ*987654321 *020201*1535*U*00300*000004323*0*P
GS*PO*901234572000*908887732000*020201*1535*4369*X*003020
```

<i>routingMode</i> Value	Description
GSOnly	<ul style="list-style-type: none"> ■ For Interchange documents, the EDI Module uses the sender and receiver from the interchange header. For example, the sender uses EDI ID qualifier “01” with value “123456789” and the receiver uses the EDI ID qualifier “ZZ” with the value “987654321”. ■ For Group and Transaction documents, the EDI Module uses the sender and receiver from the group header. For example, the sender uses the value “901234572000” and the receiver uses the value “90888773200”. For the EDI ID qualifiers, see the <i>GSRouting/senderQualifier</i> and <i>GSRouting/receiverQualifier</i> variables.

<i>GSRouting/routingMode</i> EDITPA Variable		Default: OFF
GS&ISA	<ul style="list-style-type: none"> ■ For Interchange documents, the EDI Module uses the sender and receiver from the interchange header. For example, the sender uses EDI ID qualifier “01” with value “123456789” and the receiver uses the EDI ID qualifier “ZZ” with the value “987654321”. ■ For Group and Transaction documents, the EDI Module derives the sender by concatenating ISA05, ISA06, and GS02 fields, using a colon (:) as the separator. The EDI Module derives the receiver ID by concatenating ISA07, ISA08, and GS03. For example, the sender would be 01:123456789:901234572000 and the receiver would be zz:987654321:908887732000. The EDI ID qualifiers are <i>always ZZ</i> (Mutually Defined) when you use GS&ISA. 	
OFF	<ul style="list-style-type: none"> ■ For Interchange, Group, and Transaction documents, the EDI Module uses the sender and receiver from the interchange header. For example, the sender uses EDI ID qualifier “01” with value “123456789” and the receiver uses the EDI ID qualifier “ZZ” with the value “987654321”. <p>Note that this setting does not apply when using non-standard processing. If <i>routingMode</i> is OFF and you are using non-standard processing, the EDI Module uses the <i>GSonly</i> setting. For more information about group level processing, see “Using Standard or Non-Standard Processing” on page 110.</p>	

<i>GSRouting/senderQualifier</i> EDITPA Variable		Default: *									
The <i>senderQualifier</i> variable indicates the EDI ID qualifier that you want the EDI Module to use for the sender when the <i>GSRouting/routingMode</i> variable is <i>GSonly</i> .											
<i>senderQualifier</i> Value	Description										
<table border="1"> <tr> <td>*</td><td>The EDI Module uses the EDI ID qualifier for the sender from the interchange header (e.g., ISA05 field).</td><td></td></tr> <tr> <td>other EDI ID qualifiers</td><td>You can specify any EDI ID qualifier that the EDI standard supports. For all EDI ID qualifiers, see the EDI Standards documentation for your EDI standard and version.</td><td></td></tr> <tr> <td colspan="2"> <p>Note: Be sure to add the external ID types to Trading Networks for the EDI ID qualifier you specify, if necessary. For more information, see “Adding External ID Types for EDI ID Qualifiers” on page 114.</p> </td><td></td></tr> </table>			*	The EDI Module uses the EDI ID qualifier for the sender from the interchange header (e.g., ISA05 field).		other EDI ID qualifiers	You can specify any EDI ID qualifier that the EDI standard supports. For all EDI ID qualifiers, see the EDI Standards documentation for your EDI standard and version.		<p>Note: Be sure to add the external ID types to Trading Networks for the EDI ID qualifier you specify, if necessary. For more information, see “Adding External ID Types for EDI ID Qualifiers” on page 114.</p>		
*	The EDI Module uses the EDI ID qualifier for the sender from the interchange header (e.g., ISA05 field).										
other EDI ID qualifiers	You can specify any EDI ID qualifier that the EDI standard supports. For all EDI ID qualifiers, see the EDI Standards documentation for your EDI standard and version.										
<p>Note: Be sure to add the external ID types to Trading Networks for the EDI ID qualifier you specify, if necessary. For more information, see “Adding External ID Types for EDI ID Qualifiers” on page 114.</p>											

<i>GSRouting/receiverQualifier</i> EDITPA Variable		Default: *
The <i>receiverQualifier</i> variable indicates the EDI ID qualifier that you want the EDI Module to use for the receiver when the <i>GSRouting/routingMode</i> variable is <code>GSOnly</code> .		
<i>receiverQualifier</i> Value	Description	
*	The EDI Module uses the EDI ID qualifier for the receiver from the interchange header (e.g., ISA07 field).	
other EDI ID qualifiers	You can specify any EDI ID qualifier that the EDI standard supports. For all EDI ID qualifiers, see the EDI Standards documentation for your EDI standard and version.	
<p>Note: Be sure to add the external ID types to Trading Networks for the EDI ID qualifier you specify, if necessary. For more information, see “Adding External ID Types for EDI ID Qualifiers” on page 114.</p>		

<i>processingMode</i> EDITPA Variable		Default: Testing
The <i>processingMode</i> variable indicates whether the partners using the EDITPA are in testing mode, production mode, or custom mode. When processing Interchange, Group, and Transaction document, the EDI Module includes the custom attribute EDI Processing Mode in the BizDocEnvelope and sets the value of this attribute based on the <i>processingMode</i> EDITPA variable.		
<i>processingMode</i> Value	Description	
Testing	Use this value when testing the exchange of documents between two partners. For example, when the production mode is <code>Testing</code> , you might create processing rules that accept the documents and do all processing except passing the document to production applications.	
Production	Use this value when you are confident that your logic for exchanging documents is successful.	
Custom	This is a user-defined setting.	

<i>processingMode</i> EDITPA Variable		Default: Testing
Interchange Define	The EDI Module sets the value of the custom attribute, EDI Processing Mode, based on the processing mode defined in the interchange header.	
	■ For an ANSI X12 document, the EDI Module uses the value in the ISA015 field as described below.	
For this value of ISA015...	The EDI Module sets the value of EDI Processing Mode attribute to...	
T or others	Testing	
P	Production	
I	Custom	
	■ For a UN/EDIFACT document, the EDI Module uses the value in the UNB11 field as described below.	
For this value of UNB11...	The EDI Module sets the value of EDI Processing Mode attribute to...	
1,2,3,4 or others	Testing	
Empty	Production	

<i>persistMultipleDocEnvelope</i> EDITPA Variable	Default: true
The <i>persistMultipleDocEnvelope</i> variable indicates whether you want the EDI Module to save the original EDI document to the Trading Networks database. The original EDI document that Trading Networks receives typically contains multiple interchange segments. The EDI Module only uses the <i>persistMultipleDocEnvelope</i> variable from the default EDITPA.	

Note: The EDI Module splits each interchange segment within the original EDI document into Interchange, Group, and Transaction documents based on the setting of the *splitOption* EDITPA variable, and you can control whether the Interchange, Group, and Transaction documents are saved to the Trading Networks database via the processing rule.

<i>persistMultipleDocEnvelope</i> EDITPA Variable	Default: true
<i>persistMultipleDocEnvelope</i> Value	Description
true	The EDI Module saves the original EDI document to the Trading Networks database. Note that the document is saved with the sender and receiver both set to Unknown.
false	The EDI Module does <i>not</i> save the original EDI document to the Trading Networks database. If you specify false, you will not have a way to retrieve the original EDI document.
<i>ControlNumberManagement/validateInboundEnvelopeControlNumbers</i> EDITPA Variable	Default: false
<p>The <i>ControlNumberManagement/validateInboundEnvelopeControlNumbers</i> variable indicates whether you want the EDI Module to validate and track control numbers in the interchange headers of inbound EDI documents. For more information, see “Trading Networks Attributes and EDI Documents” on page 196 in Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”.</p>	
Non-standard	When you are using non-standard processing, EDI Module does not use this EDITPA variable. Instead, it uses the Validate inbound envelope control numbers setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “ Defining Interchange-Level Sender/Receiver Pair Information ” on page 390 in Appendix A , “ Non-Standard Processing ”.
<i>validateInboundEnvelopeControlNumbers</i> Value	Description
true	The EDI Module validates and tracks control numbers in the interchange headers of inbound EDI documents.
false	The EDI Module does <i>not</i> validate or track control numbers in the interchange headers of inbound EDI documents.

ControlNumberManagement/validateInboundGroupControlNumbers
EDITPA Variable

Default: false

The *ControlNumberManagement/validateInboundGroupControlNumbers* variable indicates whether you want the EDI Module to validate and track control numbers in the group headers of inbound EDI documents. For more information, see “[Trading Networks Attributes and EDI Documents](#)” on page 196 in Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”.

validateInboundGroupControlNumbers Value

Description

true

The EDI Module validates and tracks control numbers in the group headers of inbound EDI documents.

false

The EDI Module does *not* validate or track control numbers in the group headers of inbound EDI documents.

ControlNumberManagement/duplicateControlNumberAction EDITPA Variable

Default: Error & Continue

The *ControlNumberManagement/duplicateControlNumberAction* variable indicates the action you want the EDI Module to take when it encounters a duplicate control number in an inbound document when it is validating interchange and/or group control numbers.

For more information, see “[Trading Networks Attributes and EDI Documents](#)” on page 196 in Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”. For more information about each of the following actions, see “[Actions the EDI Module Can Take for Invalid Control Numbers](#)” on page 240.

Non-standard

When you are using non-standard processing, this EDITPA is only used for duplicate group control numbers. To set the action for duplicate interchange control numbers, use the Duplicate control number action setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in Appendix A, “Non-Standard Processing”.

<i>ControlNumberManagement/duplicateControlNumberAction</i> EDITPA Variable	Default: Error & Continue
<i>duplicateControlNumberAction</i> Value	Action the EDI Module takes:
Error & Continue	The EDI Module logs the error; then continues to process the EDI document that contains the invalid control number normally.
ProcessNormally	The EDI Module logs a warning and process the EDI document that contains the invalid control number normally.
Reject	<p>The EDI Module logs the error and does <i>not</i> process the document normally. The EDI Module does <i>not</i> split the EDI document. Typically, the EDI Module splits an inbound EDI based on the EDITPA <i>splitOption</i> variable and sends the documents it splits out to Trading Networks for processing. However, if you set the action to Reject, the EDI Module sends the document without splitting it to Trading Networks processing rules.</p> <p>Additionally, the EDI Module sets the Trading Networks custom attribute EDI Status to Duplicate Control Number. You can use the custom attribute EDI Status in processing rule criteria. You should create a processing rule to handle this rejected document. For information, see “Defining Processing Rules to Handle Documents with Invalid Control Numbers” on page 247.</p> <p>You can later force processing of the duplicate document if you want. For more information, see “Reprocessing EDI Documents with Invalid Control Numbers” on page 250.</p>

ControlNumberManagement/useReverseRouting
EDITPA Variable

Default: False

The *ControlNumberManagement/useReverseRouting* variable indicates that if a UNG is not present in a UN/EDIFACT document, the partner IDs for UN/EDIFACT transaction documents will be determined as follows:

<i>useReverseRouting</i> Value	Action the EDI Module takes:
false	The EDI Module uses the <i>ID</i> and <i>Qualifier</i> portions of the UN/EDIFACT Sender/Receiver ID to determine the partner ID of the document. This is the default.
true	The EDI Module uses the <i>ReverseRoute Id</i> and <i>Qualifier</i> portions of the UN/EDIFACT Sender/Receiver ID to determine the partner ID of the document.

Note: Once your production system is running, do *not* change the value of the *useReverseRouting* variable. For example, if you send a document with *useReverseRouting* set to false, the EDI Module considers your partner ID to be the values of *ID* and *Qualifier*, and it stores these values in the EDITRACKING table. If you change *useReverseRouting* to true and then receive an FA from the receiver, the EDI Module would try to use the *ID* and *ReverseRoute Id* values to try to determine the sender of the FA you received. In this case, the EDI Module cannot reconcile the FA with the document you sent.

ControlNumberManagement/duplicateControlNumberAction
EDITPA Variable

Default: Error & Continue

The *ControlNumberManagement/duplicateControlNumberAction* variable indicates the action you want the EDI Module to take when it encounters a duplicate control number in an inbound document when it is validating interchange and/or group control numbers.

For more information, see “[Trading Networks Attributes and EDI Documents](#)” on page 196 in Chapter 13, “[Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks](#)”. For more information about each of the following actions, see “[Actions the EDI Module Can Take for Invalid Control Numbers](#)” on page 240.

ControlNumberManagement/duplicateControlNumberAction EDITPA

Variable

Default: Error & Continue

Non-standard

When you are using non-standard processing, this EDITPA is only used for duplicate group control numbers. To set the action for duplicate interchange control numbers, use the **Duplicate control number action** setting that you set from the **Interchange Information Detail** screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in [Appendix A, “Non-Standard Processing”](#).

duplicateControlNumberAction

Value

Action the EDI Module takes:

Error & Continue

The EDI Module logs the error; then continues to process the EDI document that contains the invalid control number normally.

ProcessNormally

The EDI Module logs a warning and process the EDI document that contains the invalid control number normally.

Reject

The EDI Module logs the error and does *not* process the document normally. The EDI Module does *not* split the EDI document. Typically, the EDI Module splits an inbound EDI based on the EDITPA *splitOption* variable and sends the documents it splits out to Trading Networks for processing. However, if you set the action to Reject, the EDI Module sends the document without splitting it to Trading Networks processing rules.

Additionally, the EDI Module sets the Trading Networks custom attribute EDI Status to Duplicate Control Number. You can use the custom attribute EDI Status in processing rule criteria. You should create a processing rule to handle this rejected document. For information, see “[Defining Processing Rules to Handle Documents with Invalid Control Numbers](#)” on page 247.

You can later force processing of the duplicate document if you want. For more information, see “[Reprocessing EDI Documents with Invalid Control Numbers](#)” on page 250.

<i>FAGeneration/addGroup</i> EDITPA Variable	Default: Off
--	--------------

The *FAGeneration/addGroup* variable indicates whether you want to add a functional group to the UN/EDIFACT FA (e.g., CONTRL). For more information, see “[Turning Automatic FA Generation On or Off](#)” on page 259 in Chapter 15, “Optional Inbound Processing When Using Trading Networks”.

<i>addGroup</i> Value	The EDI Module:
true	Add a functional group to the FA.
false	Do not add a functional group to the FA. This is the default.

<i>FAGeneration/autoGenerateFA</i> EDITPA Variable	Default: Off
--	--------------

The *FAGeneration/autoGenerateFA* variable indicates whether you want the EDI Module to automatically generate FAs for an inbound EDI document. For more information, see “[Turning Automatic FA Generation On or Off](#)” on page 259 in Chapter 15, “Optional Inbound Processing When Using Trading Networks”.

Non-standard

When you are using non-standard processing, the EDI Module does not use this EDITPA variable. Instead, it uses the Auto Generate FA setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in Appendix A, “Non-Standard Processing”.

<i>autoGenerateFA</i> Value	The EDI Module:
On	Always automatically generates FAs for inbound EDI documents
Off	Never automatically generates FAs for inbound EDI documents
Per Document	Automatically generates FAs based on the indicator flag in the interchange header (e.g., ISA14 or UNB09)

<i>FAGeneration/FALevel</i> EDITPA Variable	Default: Default
---	------------------

The *FAGeneration/FALevel* variable defines the level of detail that the EDI Module acknowledges in the FAs that it automatically generates. For more information, see “[Variables that Affect How the EDI Module Generates the FA](#)” on page 260 in Chapter 15, “Optional Inbound Processing When Using Trading Networks”.

<i>FAGeneration/FALevel</i> EDITPA Variable	Default: Default
Non-standard	When you are using non-standard processing, the EDI Module does not use this EDITPA variable. Instead, it uses the FA Level setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “ Defining Interchange-Level Sender/Receiver Pair Information ” on page 390 in Appendix A, “Non-Standard Processing”.
<i>FALevel</i> Value	The EDI Module acknowledges at the:
Default	Envelope level (group for ANSI X12 documents and interchange for UN/EDIFACT documents)
TransactionSet	Transaction set level
Segment	Segment level
Element	Element level
<p>Note: If you are generating FAs at the element level, be sure to configure the maximum number of errors to report per FA transaction. For more information, see “Configuring the Maximum Number of Transaction Errors” on page 58.</p>	

<i>FAGeneration/generateControlNumber</i> EDITPA Variable	Default: FromInboundDocument
The <i>FAGeneration/generateControlNumber</i> variable defines how the EDI Module is to generate the control numbers that it uses in the interchange and group headers of the FAs that it automatically generates. For more information, see “ Variables that Affect How the EDI Module Generates the FA ” on page 260 in Chapter 15, “Optional Inbound Processing When Using Trading Networks”.	
Non-standard	When you are using non-standard processing, the EDI Module does not use this EDITPA variable. Instead, it uses the Generate Control Number setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “ Defining Interchange-Level Sender/Receiver Pair Information ” on page 390 in Appendix A, “Non-Standard Processing”.

<i>FAGeneration/generateControlNumber</i> EDITPA Variable		Default: FromInboundDocument
<i>generateControlNumber</i>		
Value	The EDI Module:	
FromInboundDocument	Use the control numbers from the corresponding headers of the inbound EDI document that the FA acknowledges.	
Random	Randomly generate control numbers for the interchange and group headers of the FA.	
FromControlNumberTable	Obtain the control numbers from the EDIControlNumber table.	

<i>FAGeneration/rejectionRules/syntaxErrorStatus</i> EDITPA Variable		Default: Rejected
The <i>FAGeneration/rejectionRules/syntaxErrorStatus</i> variable indicates how you want the EDI Module to report the syntax error status for a transaction, group, or UN/EDIFACT interchange.		
Non-standard	When you are using non-standard processing, the EDI Module does not use this EDITPA variable. Instead, it uses the Syntax Error Status setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “ Defining Interchange-Level Sender/Receiver Pair Information ” on page 390 in Appendix A, “Non-Standard Processing”.	

<i>FAGeneration/rejectionRules/syntaxErrorStatus</i> EDITPA Variable		Default: Rejected
<i>syntaxErrorStatus</i> Value	The EDI Module:	
Rejected	<p>Reports the syntax error status:</p> <ul style="list-style-type: none"> ■ “Accepted” if there are no syntax errors ■ “Rejected” if there are syntax errors <p>Use this setting if you want to reject the element (e.g., transaction) because of the syntax errors.</p>	
Accepted, But Errors Were Noted	<p>Reports the syntax error status:</p> <ul style="list-style-type: none"> ■ “Accepted” if there are no syntax errors ■ “Accepted, But Errors Were Noted” if there are syntax errors <p>Use this setting if you want to know whether there were syntax errors, but do not want to reject the element (e.g., transaction) because of the syntax errors.</p>	
Accepted	<p>Always reports the syntax error status as “Accepted” regardless of whether there are any syntax errors.</p> <p>Use this setting if you do not want to check for syntax errors.</p>	

<i>FAGeneration/rejectionRules/logicalErrorStatus</i> EDITPA Variable		Default: Rejected
<p>The <i>FAGeneration/rejectionRules/logicalErrorStatus</i> variable indicates how you want the EDI Module to report the logical error status. For more information, see “Logical Error Status” on page 265 in Chapter 15, “Optional Inbound Processing When Using Trading Networks”.</p>		
Non-standard	When you are using non-standard processing, the EDI Module does not use this EDITPA variable. Instead, it uses the Logical Error Status setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “ Defining Interchange-Level Sender/Receiver Pair Information ” on page 390 in Appendix A, “Non-Standard Processing”.	

<i>FAGeneration/rejectionRules/logicalErrorStatus</i> EDITPA Variable		Default: Rejected
<i>logicalErrorStatus</i> Value	The EDI Module:	
Rejected	<p>Reports the logical error status:</p> <ul style="list-style-type: none"> ■ “Accepted” if there are no logical errors ■ “Rejected” if there are logical errors <p>Use this setting if you want to reject the element (e.g., transaction) because of the logical errors.</p>	
Accepted, But Errors Were Noted	<p>Reports the logical error status:</p> <ul style="list-style-type: none"> ■ “Accepted” if there are no logical errors ■ “Accepted, But Errors Were Noted” if there are logical errors <p>Use this setting if you want to know whether there were logical errors, but do not want to reject the element (e.g., transaction) because of the logical errors.</p>	
Accepted	<p>Always reports the logical error status as “Accepted” regardless of whether there are any logical errors.</p> <p>Use this setting if you do not want to check for logical errors.</p>	

FAGeneration/rejectionRules/childTransactionRejectedStatus EDITPA Variable

Default: Rejected

The *FAGeneration/rejectionRules/childTransactionRejectedStatus* variable indicates how you want the EDI Module to report the child transaction rejected status. The child transaction rejected status indicates whether child elements of a group or UN/EDIFACT interchange have an FA status of “Rejected”. For more information, see “[Child Transaction Rejected Status](#)” on page 267 in [Chapter 15, “Optional Inbound Processing When Using Trading Networks”](#).

Non-standard

When you are using non-standard processing, the EDI Module does not use this EDITPA variable. Instead, it uses the [Child Transaction Rejected Status](#) setting that you set from the [Interchange Information Detail](#) screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in [Appendix A, “Non-Standard Processing”](#).

<i>FAGeneration/rejectionRules/childTransactionRejectedStatus</i> EDITPA Variable		Default: Rejected
<i>childTransactionRejectedStatus</i>	Value	The EDI Module:
	Rejected	<p>Reports the child transaction rejected status as:</p> <ul style="list-style-type: none"> ■ “Rejected” if the FA status of any of the child transactions is either “Rejected” or “Accepted, But Errors Were Noted”. ■ “Accepted” if the FA statuses of all the child transactions are “Accepted”
	Partially Accepted	<p>Reports the child transaction rejected status as:</p> <ul style="list-style-type: none"> ■ “Rejected” if the FA statuses of all of the child transactions are “Rejected”. ■ “Partially Accepted” if the FA status of at least one child transaction is “Accepted”, but the FA status of other child transactions are “Rejected” and/or “Accepted, But Errors Were Noted”. ■ “Accepted, But Errors Were Noted” if the FA statuses of the child transactions are either “Rejected” and/or “Accepted, But Errors Were Noted” -and- no child transactions are “Accepted”. ■ “Accepted” if the FA statuses of all the child transactions are “Accepted”.
	Accepted, But Errors Were Noted	<p>Reports the child transaction rejected status as:</p> <ul style="list-style-type: none"> ■ “Rejected” if all the child transactions are “Rejected”. ■ “Accepted, But Errors Were Noted” if the FA statuses of the child transactions are “Rejected”, “Accepted, But Errors Were Noted”, and “Accepted”. ■ “Accepted” if the FA statuses of all the child transactions are “Accepted”.

<i>FAGeneration/processDocument</i> EDITPA Variable	Default: All
---	--------------

The *FAGeneration/processDocument* variable indicates how you want the EDI Module to process a transaction, group, or UN/EDIFACT interchange based on its FA status. Use this EDITPA variable to define the FA statuses that are acceptable and unacceptable.

- For acceptable FA statuses, the EDI Module processes a transaction, group, or UN/EDIFACT interchange using its normal processing.
- For unacceptable FA statuses, the EDI Module performs different processing.

For more information, see “[Actions the EDI Module Takes Based on FA Status](#)” on page 272 in Chapter 15, “Optional Inbound Processing When Using Trading Networks”.

Non-standard

When you are using non-standard processing, the EDI Module does not use this EDITPA variable. Instead, it uses the Child Transaction Rejected Status setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in Appendix A, “Non-Standard Processing”.

<i>processDocument</i>	Value	Acceptable FA Statuses	Unacceptable FA Statuses
All		all FA statuses	No unacceptable FA statuses
Only Accepted		Accepted	Not Allowed Rejected Partially Accepted Accepted, But Errors Were Noted
Not Rejected		Not Allowed Partially Accepted Accepted, But Errors Were Noted Accepted	Rejected

FAReconciliation EDITPA Variable	Default: false
----------------------------------	----------------

The *FAReconciliation* variable indicates whether you want to enable FA reconciliation, so the EDI Module:

- Reconciles inbound FA acknowledgments that it receives with the outbound EDI documents that it has sent.

-AND-

- Reconciles outbound FA acknowledgments that it sends with the inbound EDI documents that it has received.

To do FA reconciliation, the EDI Module makes a record of each Group/Interchange EDI document that it sends and receives through Trading Networks. For ANSI EDI documents, the EDI Module records each Group document it sends or receives. For UN/EDIFACT EDI documents, the EDI Module records each Group document if the EDI document contains Group-level documents; if it does not, the EDI Module records Interchange-level document. The EDI Module records the information about these documents to the EDITRACKING table, which is an EDI Module-specific table in the Trading Networks database.

Note: If you form an outbound EDI document and send it directly to the Trading Networks processing rules rather than allow Trading Networks to recognize the document with its TN document types, you will need to invoke the `wm.b2b.editn:trackEDIdocs` service to record the outbound document in the EDITRACKING table. For more information “[Routing the Outbound EDI Document to Trading Networks](#)” on page 301 in Chapter 16, “Forming EDI Documents to Send Outbound When Using Trading Networks”.

When FA reconciliation is enabled, the EDI Module updates the status for each Group/Interchange document in the EDITRACKING table when it sends or receives the Group/Interchange document’s corresponding FA. For more information about FA reconciliation, see [Chapter 21, “Reconciling Functional Acknowledgments”](#).

<i>FAReconciliation</i> EDITPA Variable		Default: false
<i>FAReconciliation</i>	Description	
true	<p>FA reconciliation is enabled.</p> <ul style="list-style-type: none"> ■ When the EDI Module sends or receives a Group/Interchange document through Trading Networks, it adds an entry to the EDITRACKING table the document and sets the FA status to None. ■ When the EDI Module receives or sends the corresponding FA, it attempts to locate the corresponding Group/Interchange document in the EDITRACKING table. <ul style="list-style-type: none"> ■ If it locates a matching entry, it updates the FA status. For example, if the EDI Module locates one matching document and the FA has the "A" (Accept) status, the EDI Module updates the FA status to Accept. For a list of all possible statuses, see ""FA Statuses" on page 365. ■ If the EDI Module cannot find a matching entry, it adds an entry for the FA to the EDITRACKING table. 	
false	<p>FA reconciliation is disabled.</p> <ul style="list-style-type: none"> ■ When the EDI Module sends or receives a Group/Interchange document through Trading Networks, it adds an entry to the EDITRACKING table the document and sets the FA status to Disable. ■ When the EDI Module receives or sends the corresponding FA, it attempts to locate the corresponding Group/Interchange document in the EDITRACKING table. <ul style="list-style-type: none"> ■ If it locates a matching entry and the status is Disable, it does nothing. ■ If it locates a matching entry and the status is None, it updates the status to Disable. <p>If it cannot find a matching entry, it adds an entry for the FA to the EDITRACKING table.</p>	

<i>UNAmode</i> EDITPA Variable	Default: auto
The <i>UNAmode</i> variable indicates whether you want to create a UNA segment prior to the interchange in an outbound UN/EDIFACT EDI document.	
When you use the EDI Module batching feature to create an outbound EDI document, the batchProcess service honors this setting. For more information, see Chapter 17, “Batching EDI Documents” . Additionally, services you create to form outbound UN/EDIFACT EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
<i>UNAmode</i> Value	Description
yes	Create the UNA segment prior to the interchange in the outbound EDI UN/EDIFACT document.
no	Do <i>not</i> create UNA segment prior to the interchange in the outbound EDI UN/EDIFACT document.
auto	Create the UNA segment prior to the interchange in the outbound EDI UN/EDIFACT document.
<i>delimiters/record</i> EDITPA Variable	Default: (no default)
The <i>delimiters/record</i> variable indicates the segment terminator to use in an outbound EDI document, for example, +. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When you use the EDI Module batching feature to create an outbound EDI document, if you do not specify the <i>delimiters</i> input variable to the batchProcess service, the batchProcess service retrieves this setting from the EDITPA. For more information, see “Delimiters Used for the Batch EDI Document” on page 318 in Chapter 17, “Batching EDI Documents” .	
<i>delimiters/field</i> EDITPA Variable	Default: (no default)
The <i>delimiters/field</i> variable indicates the field separator to use for each EDI segment in an outbound EDI document, for example, !. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When you use the EDI Module batching feature to create an outbound EDI document, if you do not specify the <i>delimiters</i> input variable to the batchProcess service, the batchProcess service retrieves this setting from the EDITPA. For more information, see “Delimiters Used for the Batch EDI Document” on page 318 in Chapter 17, “Batching EDI Documents” .	

<i>delimiters/subfield</i> EDITPA Variable	Default: (no default)
The <i>delimiters/subfield</i> variable indicates the separator to use for composite elements in an outbound EDI document, for example, *. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When you use the EDI Module batching feature to create an outbound EDI document, if you do not specify the <i>delimiters</i> input variable to the <code>batchProcess</code> service, the <code>batchProcess</code> service retrieves this setting from the EDITPA. For more information, see “ Delimiters Used for the Batch EDI Document ” on page 318 in Chapter 17, “Batching EDI Documents” .	
<i>delimiters/release</i> EDITPA Variable	Default: (no default)
The <i>delimiters/release</i> variable indicates the release character to use for an outbound EDI document, for example, the release character ?. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When you use the EDI Module batching feature to create an outbound EDI document, if you do not specify the <i>delimiters</i> input variable to the <code>batchProcess</code> service, the <code>batchProcess</code> service retrieves this setting from the EDITPA. For more information, see “ Delimiters Used for the Batch EDI Document ” on page 318 in Chapter 17, “Batching EDI Documents” .	

***publishBatchFailEvent* EDITPA Variable**

Default: false

The *publishBatchFailEvent* variable indicates whether you want the EDI Module to publish an IS document when it is unable to include an EDI document that is queued for batching into the final batch EDI document. The format of the IS document is defined by the `wm.b2b.editn.publishedDocs:batchFailDocument`. For more information about this IS document type, see the *webMethods EDI Module Built-In Services Reference*. For more information about how this EDITPA variable is used during batching and how to handle the failure, see “[Updating the Task Status and Publishing Documents in the Case of Failure](#)” on page 319.

The following table lists the possible values for the *publishBatchFailEvent* variable:

<i>publishBatchFailEvent</i> value	Description
true	The EDI Module publishes an IS document when it encounters errors including an EDI document into the final batch EDI document.
false	The EDI Module does <i>not</i> publish an IS document when it encounters errors including an EDI document into the final batch EDI document.

***envelopeIdentifier/sender/ID* EDITPA Variable**

Default: (no default)

The *envelopeIdentifier/sender/ID* variable indicates the sender ID to use for an outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-in Services Reference*.

When using the EDI Module batching feature:

- If the `batchProcess` service *oneBatchQueue* input variable is `NONE`, the `batchProcess` service uses its input variables to locate the EDITPA, and then uses the *envelopeIdentifier* variables to set the sender/receiver in the interchange and group headers of the outbound batch EDI document. For more information, see [Appendix B, “Using the 6.0.1 Version of the Batching Feature”](#).
- If you are using batching with *oneBatchQueue* set to `SINGLEOUTPUT` or `MULTIPLEOUTPUTS`, you should not specify the *envelopeIdentifier/sender/ID* variable.

<i>envelopeIdentifier/sender/qualifier</i> EDITPA Variable	Default: (no default)
--	-----------------------

The *envelopeIdentifier/sender/qualifier* variable indicates the EDI ID qualifier associated with the value you specify for *envelopeIdentifier/sender/ID*. For example, if you specified a D-U-N-S number for the *envelopeIdentifier/sender/ID*, for *envelopeIdentifier/sender/qualifier* specify 01, which is the EDI ID qualifier for a D-U-N-S number. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-in Services Reference*.

When using the EDI Module batching feature:

- If the batchProcess service *oneBatchQueue* input variable is `NONE`, the batchProcess service uses its input variables to locate the EDITPA, and then uses the *envelopeIdentifier* variables to set the sender/receiver in the interchange and group headers of the outbound batch EDI document. For more information, see [Appendix B, "Using the 6.0.1 Version of the Batching Feature"](#).
- If you are using batching with *oneBatchQueue* set to `SINGLEOUTPUT` or `MULTIPLEOUTPUTS`, you should not specify the *envelopeIdentifier/sender/ID* variable.

<i>envelopeIdentifier/receiver/ID</i> EDITPA Variable	Default: (no default)
---	-----------------------

The *envelopeIdentifier/receiver/ID* variable indicates the receiver ID to use for an outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-in Services Reference*.

When using the EDI Module batching feature:

- If the batchProcess service *oneBatchQueue* input variable is `NONE`, the batchProcess service uses its input variables to locate the EDITPA, and then uses the *envelopeIdentifier* variables to set the sender/receiver in the interchange and group headers of the outbound batch EDI document. For more information, see [Appendix B, "Using the 6.0.1 Version of the Batching Feature"](#).
- If you are using batching with *oneBatchQueue* set to `SINGLEOUTPUT` or `MULTIPLEOUTPUTS`, you should not specify the *envelopeIdentifier/receiver/ID* variable.

<i>envelopeIdentifier/receiver/qualifier</i> EDITPA Variable	Default: (no default)
The <i>envelopeIdentifier/receiver/qualifier</i> variable indicates the EDI ID qualifier associated with the value you specify for <i>envelopeIdentifier/receiver/ID</i> . For example, if you specified a D-U-N-S number for the <i>envelopeIdentifier/receiver/ID</i> , for <i>envelopeIdentifier/receiver/qualifier</i> specify 01, which is the EDI ID qualifier for a D-U-N-S number. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When using the EDI Module batching feature:	
<ul style="list-style-type: none"> ■ If the <code>batchProcess</code> service <i>oneBatchQueue</i> input variable is <code>NONE</code>, the <code>batchProcess</code> service uses its input variables to locate the EDITPA, and then uses the <i>envelopeIdentifier</i> variables to set the sender/receiver in the interchange and group headers of the outbound batch EDI document. For more information, see Appendix B, “Using the 6.0.1 Version of the Batching Feature”. ■ If you are using batching with <i>oneBatchQueue</i> set to <code>SINGLEOUTPUT</code> or <code>MULTIPLEOUTPUTS</code>, you should not specify the <i>envelopeIdentifier/sender/ID</i> variable. 	
<i>ICheaderInfo/ISA/ISA01</i> EDITPA Variable	Default: 00
The <i>ICheaderInfo/ISA/ISA01</i> variable indicates the value to use for the ISA01 element of an ANSI X12 interchange header in an outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When using the EDI Module batching feature, the EDI Module uses this setting when creating the interchange headers of the outbound batch EDI document.	
<i>ICheaderInfo/ISA/ISA02</i> EDITPA Variable	Default: (no default)
The <i>ICheaderInfo/ISA/ISA02</i> variable indicates the value to use for the ISA02 element of an ANSI X12 interchange header in an outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When using the EDI Module batching feature, the EDI Module uses this setting when creating the interchange headers of the outbound batch EDI document.	

ICheaderInfo/ISA/ISA03 EDITPA Variable**Default: 00**

The *ICheaderInfo/ISA/ISA03* variable indicates the value to use for the ISA03 element of an ANSI X12 interchange header in an outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-in Services Reference*.

When using the EDI Module batching feature, the EDI Module uses this setting when creating the interchange headers of the outbound batch EDI document.

ICheaderInfo/ISA/ISA04 EDITPA Variable**Default: (no value)**

The *ICheaderInfo/ISA/ISA04* variable indicates the value to use for the ISA04 element of an ANSI X12 interchange header in an outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-in Services Reference*.

When using the EDI Module batching feature, the EDI Module uses this setting when creating the interchange headers of the outbound batch EDI document.

ICheaderInfo/ISA/ISA11 EDITPA Variable**Default: U**

The *ICheaderInfo/ISA/ISA11* variable indicates the value to use for the ISA11 element of an ANSI X12 interchange header in an outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-in Services Reference*.

When using the EDI Module batching feature, the EDI Module uses this setting when creating the interchange headers of the outbound batch EDI document.

ICheaderInfo/GS/GS07 EDITPA Variable**Default: (no value)**

The *ICheaderInfo/GS/GS07* variable indicates the value to use for the GS07 element of an ANSI X12 group header in an outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-in Services Reference*.

When using the EDI Module batching feature, the EDI Module uses this setting when creating the interchange headers of the outbound batch EDI document.

<i>ICheaderInfo/UNB/UNB01</i> EDITPA Variable	Default: (no value)
The <i>ICheaderInfo/UNB/UNB01</i> variable indicates the value to use for the UNB01 element of a UN/EDIFACT UNB header in the outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When using the EDI Module batching feature, the EDI Module uses this setting when creating the interchange headers of the outbound batch EDI document.	
<i>ICheaderInfo/UNB/UNB06</i> EDITPA Variable	Default: (no value)
The <i>ICheaderInfo/UNB/UNB06</i> variable indicates the value to use for the UNB06 element of a UN/EDIFACT UNB header in the outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When using the EDI Module batching feature, the EDI Module uses this setting when creating the interchange headers of the outbound batch EDI document.	
<i>ICheaderInfo/UNB/UNB08</i> EDITPA Variable	Default: (no value)
The <i>ICheaderInfo/UNB/UNB08</i> variable indicates the value to use for the UNB08 element of a UN/EDIFACT UNB header in the outbound EDI document. Services you create to form outbound EDI documents can retrieve this setting from the EDITPA using the <code>wm.tn.tpa:getTPA</code> service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i> .	
When using the EDI Module batching feature, the EDI Module uses this setting when creating the interchange headers of the outbound batch EDI document.	

Defining Partner Information (TRADACOMS)

- Overview 146
- Defining Trading Networks Profiles 147
- Defining EDI Trading Partner Agreements 150

Overview

To process documents when you are using webMethods Trading Networks (Trading Networks) with the webMethods EDI Module (EDI Module), you must define Trading Networks profiles and trading partner agreements (EDITPAs) for the partners that will be identified as the senders and receivers in the EDI documents you expect to process. the EDI Module processes the transmission, its batch, and its file using settings that you define for the transmission sender/receiver pair.

To learn more about how the EDI Module processes inbound documents, see Chapter 3, "Using the EDI Module with Trading Networks", of the *webMethods EDI Module Concepts Guide*.

Partner Information You Need to Define

The following table lists the information you must define for transmission and batch sender receiver pairs that will be in the EDI documents you expect to process:

For...	Define...
Transmission sender/receiver pairs	<ul style="list-style-type: none"> ■ Trading Networks profiles for each transmission sender and receiver. For more information, see "Defining Trading Networks Profiles" on page 147.
Tailoring how the EDI Module processes documents	<ul style="list-style-type: none"> ■ Default EDITPA that defines the settings that you want to use for most partner pairs. ■ Partner-specific EDITPA for each transmission sender/receiver pair for which you want to override the settings in the default EDITPA. <p>For more information, see "Defining EDI Trading Partner Agreements" on page 150.</p>
How the EDI Module validates inbound control numbers	<ul style="list-style-type: none"> ■ Whether to validate control numbers. ■ Actions the EDI Module is to take when it encounters an invalid control number. ■ Control number validation settings (i.e., control number cap, minimum, increment, and window).

For...

Define...

Note: The term *control number* is a term used in the EDI standards ANSI X12 and UN/EDIFACT (and all supported sub-standards). It refers to a number in the header of an EDI document that is used for validation and for the ordering of documents exchanged between trading partners. A control number is equivalent to the transmission reference numbers specified in the STX and BAT segments of your TRADACOMS documents. Whether your EDI standard includes control numbers or transmission reference numbers, you define them to Trading Networks in the same manner; the only difference is in the terminology. For simplicity, Trading Networks and the EDI Module use the term *control number* to mean either control number or transmission reference number.

For more information, see [Chapter 11, "Defining Control Number Information for Partners"](#).

To learn more about:

- How EDI Module works with Trading Networks and uses profiles and EDITPAs, see Chapter 3, "Using the EDI Module with Trading Networks" of the *webMethods EDI Module Concepts Guide*.
- Profiles and TPAs and how Trading Networks uses them, see the *webMethods Trading Networks Concepts Guide*.

Defining Trading Networks Profiles

To identify the trading partners with whom you want to exchange documents, set up profiles. You must define profiles for:

- Your own corporation if you have not already done so. This is referred to as the Enterprise profile in Trading Networks.
- Your trading partners.

When you define the profiles, you need to specify the external IDs that your partners use in their documents. The external IDs correspond to the standard EDI ID qualifiers.

The EDI Module automatically installs two external IDs that Trading Networks can use to process TRADACOMS documents: Tradacoms Code and Tradacoms Name. For more information, see ["External ID Types for TRADACOMS" on page 148](#).

To create profiles

You create profiles using the Trading Networks Console. For steps to create profiles, see the chapter about creating profiles in the *webMethods Trading Networks User's Guide*.

External ID Types for TRADACOMS

The sender and receiver of a TRADACOMS transmission must be identified in Trading Networks. To identify senders and receivers, Trading Networks can use information in the transmission's STX segment to look up a partner profile for the sender and receiver of a document. That information will be associated with the BizDocEnvelope created for the transmission.

The external ID types that the EDI Module automatically installs (Tradacoms Code and Tradacoms Name) correspond to the STX segment's FROM and UNTO data elements. Each data element is a composite with two optional fields: Code and Name. Either or both of these fields could be used to identify the sender or receiver of a TRADACOMS transmission.

When Trading Networks examines partner profiles to look up a sender or receiver, there can be four possible results, depending on which fields have values:

Code present?	Name present?	Result
No	No	Unknown trading partner
Yes	No	Use Code to look up trading partner
No	Yes	Use Name to look up trading partner
Yes	Yes	Use both Name and Code to look up trading partner

For example:

If the sender's FROM data element contains...	Then the sender will be the trading partner with...
Code field="ABC" Name field is empty	The external ID type of Tradacoms Code defined to have the value "ABC".
Code field is empty Name field="ABC"	The external ID type of Tradacoms Name defined to have the value "ABC".

If the sender's FROM data element contains...	Then the sender will be the trading partner with...
Code field="ABC" Name field="XYZ"	<p>Both of the following:</p> <ul style="list-style-type: none"> ■ The external ID type of Tradacoms Code defined to have the value "ABC" and ■ The external ID type of Tradacoms Name defined to have the value "XYZ". <p>Note: To use <i>only</i> Tradacoms Code to look up partners when both Code and Name have values, see "Optionally Configuring the External ID Types" on page 149.</p>

Optionally Configuring the External ID Types

Perform the following procedure to configure the Tradacoms Code and Tradacoms Name external ID types. When you configure these external ID types, the EDI Module updates the configuration in memory (so the changes take effect immediately) and in the WmEDI/config/properties.cnf file.

To configure the external ID types

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **Configuration** menu of the navigation panel, click **Configure Properties**.
- 4 Click **Edit Properties Settings**.
- 5 Add the following properties:
 - To override the default external ID name **Tradacoms Code**, add this property:
`wm.edi.tradacoms.CodeExternalIDType=value`
 - To override the default external ID name **Tradacoms Name**, add this property:
`wm.edi.tradacoms.NameExternalIDType=value`
 - When both the Code and Name fields have values, Trading Networks uses both Code and Name to look up a trading partner. To override this behavior so that Trading Networks *only* uses Code to look up a trading partner, add this property:
`wm.edi.tradacoms.CodeOnly=true`



Note: If you set `CodeOnly=true` and if `Code` is blank, Trading Networks will look up a trading partner as described in the table in “[External ID Types for TRADACOMS](#)” on page 148.

Defining EDI Trading Partner Agreements

An EDI trading partner agreement (EDITPA) is a set of variables that you provide to tailor how the EDI Module exchanges documents between two trading partners. The EDI Module supports partner-specific EDITPAs and a single default EDITPA.

- A **partner-specific EDITPA** has a specific sender and receiver associated with it and contains variables that the EDI Module uses only when processing documents sent by the specified sender and to be received by the specified receiver.
- A **default EDITPA** has a sender and receiver set to “unknown.” It contains variables used by all trading partners when partner-specific information is not available.

During processing, the EDI Module first attempts to find values in the partner-specific EDITPA for the sender/receiver pair of a document. If a partner-specific EDITPA does not exist for the sender/receiver pair or the value in the partner-specific EDITPA is null or empty, the EDI Module uses the value from the default EDITPA.

Defining the Default EDITPA

The settings in the default EDITPA should meet the requirements of the majority of your trading partner relationships.

The first time the WmEDIforTN package is loaded into the Integration Server, the EDI Module automatically creates the default EDITPA in Trading Networks. You can modify the default EDITPA when the **Agreement Status** is “Agreed” and the **Data Status** is “Modifiable” or when the **Agreement Status** is “Proposed”.



Note: You should not disable the default EDITPA. If you do disable the default EDITPA, the EDI Module will continue to use it.

To modify the default EDITPA

You modify the default EDITPAs using the Trading Networks Console. For steps to modify EDITPAs, see the chapter about trading partner agreements in the *webMethods Trading Networks User’s Guide*.

When you modify the default EDITPA, be sure to keep the settings listed in the table below, which you can view from the Trading Networks Console **Agreement Details** screen:

For this Agreement Details screen field	Keep this setting
Sender	Unknown
Receiver	Unknown
Agreement ID	EDITPA
IS Document Type	wm.b2b.editn.TPA:EDITPA
	You can modify the values for the variables in the <code>wm.b2b.editn.TPA:EDITPA</code> IS document type. To set the values for the variables in the <code>wm.b2b.editn.TPA:EDITPA</code> IS document type, see the description of the variables in “ wm.b2b.editn.TPA:EDITPA IS Document Type ” on page 152.

Defining a Partner-Specific EDITPA

You only need to create partner-specific EDITPAs if you have one or more sender/receiver pairs that require settings that are different from those you specify in the default EDITPA. When creating a partner-specific EDITPA, you have to specify only the information that is different from the defaults. Define partner-specific EDITPAs for transmission-level sender/receiver pairs.

 Note: You can disable partner-specific EDITPAs. When you disable a partner-specific EDITPA, the EDI Module functions as if the partner-specific EDITPA does not exist. That is, the EDI Module uses the values in the default EDITPA.

To create a partner-specific EDITPA

You create a partner-specific EDITPA using the Trading Networks Console. You can create a partner-specific EDITPA in one of the following three ways:

- Duplicate the default EDITPA and change variable values.
- Duplicate another similar partner-specific EDITPA and change variable values.
- Create an EDITPA from scratch.

For instructions on how to create TPAs (either by duplication or from scratch), see the chapter about trading partner agreements in the *webMethods Trading Networks User's Guide*.

When creating a partner-specific EDITPA, specify the following on the Trading Networks Console Agreement Details screen:

For this Agreement Details screen field	Specify
Sender	The name of the sender from the partner-specific sender/receiver pair.
Receiver	The name of the receiver from the partner-specific sender/receiver pair.
Agreement ID	EDITPA
IS Document Type	wm.b2b.editn.TPA:EDITPA To set the values for the variables in <code>wm.b2b.editn.TPA:EDITPA</code> , see the description of the variables in “wm.b2b.editn.TPA:EDITPA IS Document Type” on page 152 .
Initialization Service	Leave blank or specify a service that you created to define the default values for the <code>wm.b2b.editn.TPA:EDITPA</code> IS document type. webMethods recommends that you do <i>not</i> use the initialization service (<code>wm.b2b.editn.TPA:initService</code>) that is provided with the EDI Module. The <code>wm.b2b.editn.TPA:initService</code> service is used to populate the values of the default EDITPA.

wm.b2b.editn.TPA:EDITPA IS Document Type

The descriptions and default values for the variables of the `wm.b2b.editn.TPA:EDITPA` IS document type are listed below. At run time, the EDI Module uses a default value when the value for a variable is neither specified in a partner-specific EDITPA nor the default EDITPA. The default values are also the values that the `wm.b2b.editn.TPA:initService` initialization service sets in the default EDITPA when the EDI Module initially creates it.

Review variable descriptions to determine the value you want to specify in the default or a partner-specific EDITPA. When creating partner-specific EDITPAs, keep in mind that you should only specify values for the variables for which you want to override the default EDITPA value and leave values blank for the variables for which you want to use the defaults.



Tip! It is helpful to have an understanding about how the EDI Module processes EDI documents to understand how the EDI Module uses the variables in the EDITPA. If you have not already done so, read Chapter 3, "Using the EDI Module with Trading Networks", in the *webMethods EDI Module Concepts Guide*.

TRADACOMS/splitOption EDITPA Variable

Default: File

This variable indicates how you want the EDI Module to split a transmission segment within an EDI document. The EDI Module creates the following types of documents from a transmission segment, based on the value of this variable:

- Transmission documents that contain the single transmission envelope along with its batch segment and file.
- Batch documents that contain a single batch segment along with its files.
- File documents that contain a single file.

If you want to perform processing on the file in an inbound document, set *splitOption* to **File** or **Batch**. If you are sending the inbound EDI document through Trading Networks to simply deliver it to a destination without processing the file, set *splitOption* to **Transmission**.

Note: Validation errors can prevent the EDI Module from splitting a transmission document. The kinds of errors that will cause *splitOption* to fail are listed in “[Errors That Prevent the Splitting of Transmissions](#)” on page 161.

The following table lists the possible values for this variable and the types of documents that the EDI Module creates for each value.

<u><i>splitOption</i> Value</u>	<u>Description</u>
Transmission	The EDI Module creates only the Transmission document.
Batch	The EDI Module creates the Transmission document and a Batch document for each batch segment in the transmission segment.
File	The EDI Module creates the Transmission document, a Batch document for each batch segment in the transmission segment, and a File document for each file in the transmission segment.
	This is the initial value for the <i>splitOption</i> variable in the default EDITPA.

TRADACOMS/storageOption EDITPA Variable Default: One Message

If a File document contains many detail messages, the Trading Networks Console may not be able to easily display all of the content parts. In this case, you might want to consider using an alternative storage option.

<i>storageOption</i> Value	Description
One Message	Default. For a TRADACOMS File, Trading Networks creates one File document type that contains all the detail messages contained in the file.
<i>N</i> Messages	Instead of storing all message details in a single File document type, Trading Networks creates a separate File document type for each detail message. For more information, see “ Storage Options For File Document Types ” on page 159.

Note: When you use the Trading Networks Console’s Transaction Analysis screen to view transactions, and the value of *storageOption* is One Message, the control number displayed is the control number in the FIL segment.

If the value of *storageOption* is *N* Messages, the control number displayed is taken from the sequence number in the MHD segment.

TRADACOMS/ControlNumberManagement/validateInboundTransmissionControlNumbers EDITPA Variable Default: false

This variable indicates whether you want the EDI Module to validate and track control numbers in the transmission headers of inbound EDI documents.

<i>validateInboundTransmissionControlNumbers</i> Value	Description
true	The EDI Module validates and tracks control numbers in the transmission headers of inbound EDI documents.
false	The EDI Module does <i>not</i> validate or track control numbers in the transmission headers of inbound EDI documents.

TRADACOMS/ControlNumberManagement/validateInboundBatchControlNumbers EDITPA Variable Default: false

This variable indicates whether you want the EDI Module to validate and track control numbers in the batch headers of inbound EDI documents.

*TRADACOMS/ControlNumberManagement/validateInboundBatch
ControlNumbers* EDITPA Variable

Default: false

*validateInboundBatchControl
Numbers* Value

Description

true

The EDI Module validates and tracks control numbers in the batch headers of inbound EDI documents.

false

The EDI Module does *not* validate or track control numbers in the batch headers of inbound EDI documents.

*TRADACOMS/ControlNumberManagement/validateInboundFile
ControlNumbers* EDITPA Variable

Default: false

This variable indicates whether you want the EDI Module to validate and track control numbers in the file headers of inbound EDI documents.

*validateInboundFileControl
Numbers* Value

Description

true

The EDI Module validates and tracks control numbers in the file headers of inbound EDI documents.

false

The EDI Module does *not* validate or track control numbers in the file headers of inbound EDI documents.

*TRADACOMS/ControlNumberManagement/duplicateControlNumber
Action* EDITPA Variable

Default: Error & Continue

This variable indicates the action you want the EDI Module to take when it encounters a duplicate control number in an inbound document when it is validating transmission and/or batch control numbers.

For more information about each of the following actions, see “[Actions the EDI Module Can Take for Invalid Control Numbers](#)” on page 240.

*duplicateControlNumberAction
Value*

Action the EDI Module takes:

Error & Continue

The EDI Module logs the error; then continues to process the EDI document that contains the invalid control number normally.

ProcessNormally

The EDI Module logs a warning and process the EDI document that contains the invalid control number normally.

TRADACOMS/ControlNumberManagement/duplicateControlNumber
Action EDITPA Variable

Default: Error & Continue

Reject

The EDI Module logs the error and does *not* process the document normally. The EDI Module does *not* split the EDI document. Typically, the EDI Module splits an inbound EDI based on the EDITPA *splitOption* variable and sends the documents it splits out to Trading Networks for processing. However, if you set the action to Reject, the EDI Module sends the document without splitting it to Trading Networks processing rules.

Additionally, the EDI Module sets the Trading Networks custom attribute EDI Status to Duplicate Control Number. You can use the custom attribute EDI Status in processing rule criteria. You should create a processing rule to handle this rejected document. For information, see [“Defining Processing Rules to Handle Documents with Invalid Control Numbers” on page 247](#).

You can later force processing of the duplicate document if you want. For more information, see [“Reprocessing EDI Documents with Invalid Control Numbers” on page 250](#).

TRADACOMS/ControlNumberManagement/outOfSequenceControlNumber
Action EDITPA Variable

Default: Error & Continue

This variable indicates the action you want the EDI Module to take when it encounters an out-of-sequence control number in an inbound document when it is validating transmission and/or batch control numbers.

For more information about each of the following actions, see [“Actions the EDI Module Can Take for Invalid Control Numbers” on page 240](#).

outOfSequenceControlNumber
Action Value

Action the EDI Module takes:

Error & Continue

The EDI Module logs the error; then continues to process the EDI document that contains the invalid control number normally.

ProcessNormally

The EDI Module logs a warning and process the EDI document that contains the invalid control number normally.

TRADACOMS/ControlNumberManagement/outOfSequenceControlNumberAction EDITPA Variable

Default: Error & Continue

Reject

The EDI Module logs the error and does *not* process the document normally. The EDI Module does *not* split the EDI document. Typically, the EDI Module splits an inbound EDI based on the EDITPA *splitOption* variable and sends the documents it splits out to Trading Networks for processing. However, if you set the action to Reject, the EDI Module sends the document without splitting it to Trading Networks processing rules.

Additionally, the EDI Module sets the Trading Networks custom attribute EDI Status to Out of Sequence Control Number. You can use the custom attribute EDI Status in processing rule criteria. You should create a processing rule to handle this rejected document. For information, see [“Defining Processing Rules to Handle Documents with Invalid Control Numbers” on page 247](#).

You can later force processing of the out-of-sequence document if you want. For more information, see [“Reprocessing EDI Documents with Invalid Control Numbers” on page 250](#).

TRADACOMS/enforceApplicationReference EDITTPA Variable

Default: False

Optional. Use this variable to validate that the contents of the APRF field in the STX segment match the file type of the messages in the transmission.

enforceApplicationReference Value

Description

true

The EDI Module validates that the APRF field contents match the file type of the messages.

false

The EDI Module does not perform this validation.

<i>persistMultipleDocEnvelope</i> EDITPA Variable	Default: true
--	---------------

The *persistMultipleDocEnvelope* variable indicates whether you want the EDI Module to save the original EDI document to the Trading Networks database. The original EDI document that Trading Networks receives typically contains multiple transmission segments. The EDI Module only uses the *persistMultipleDocEnvelope* variable from the default EDITPA.

Note: The EDI Module splits each transmission segment within the original EDI document into Transmission, Batch, and File documents based on the setting of the TRADACOMS/*splitOption* EDITPA variable, and you can control whether the Transmission, Batch, and File documents are saved to the Trading Networks database via the processing rule.

<u><i>persistMultipleDocEnvelope</i> Value</u>	Description
true	<p>The EDI Module saves the original EDI document to the Trading Networks database. Note that the document is saved with the sender and receiver both set to Unknown. This is the default.</p> <p>Note: This behavior occurs only if the original EDI document contains <i>multiple</i> transmission segments.</p>
false	<p>The EDI Module does <i>not</i> save the original EDI document to the Trading Networks database. If you specify false, you will not have a way to retrieve the original EDI document.</p>

publishBatchFailEvent EDITPA Variable

Default: false

The *publishBatchFailEvent* variable indicates whether you want the EDI Module to publish an IS document when it is unable to include an EDI document that is queued for batching into the final batch EDI document. The format of the IS document is defined by the `wm.b2b.editn.publishedDocs:batchFailDocument`. For more information about this IS document type, see the *webMethods EDI Module Built-In Services Reference*. For more information about how this EDITPA variable is used during batching and how to handle the failure, see “[Updating the Task Status and Publishing Documents in the Case of Failure](#)” on page 319.

<i>publishBatchFailEvent</i> value	Description
<code>true</code>	The EDI Module publishes an IS document when it encounters errors including an EDI document into the final batch EDI document.
<code>false</code>	The EDI Module does <i>not</i> publish an IS document when it encounters errors including an EDI document into the final batch EDI document.



Note: If you plan to form EDI documents that can be sent *outbound*, you can use other variables to control outbound processing. For more information about outbound processing, see “[Specifying Variables that Affect Outbound Processing \(TRADACOMS\)](#)” on page 295 in Chapter 16, “Forming EDI Documents to Send Outbound When Using Trading Networks”.

Storage Options For File Document Types

As mentioned in “[TRADACOMS/storageOption](#) EDITPA Variable” on page 154, you can use the *TRADACOMS/storageOption* variable in your EDITPA to control how you want Trading Networks to store the content parts of your File document types.

When Trading Networks receives TRADACOMS data, it creates up to three document types: Transmission, Batch (if present), and File. The Transmission and Batch document types have a single content part called *EDIdata*, which contains all the text in that transmission or batch. In contrast, by default Trading Networks stores a File document type so that it contains *multiple* content parts—one content part for each of the following:

- Transmission/Batch header information (STX/BAT segments)
- Message header
- Message detail (one content part per message detail)
- VAT message (if applicable)

- Message trailer
- Transmission/Batch trailer information (END/EOB segments)

Example:

Message contents	The default storage method creates...
STX Header1 Detail1 Detail2 Detail3 VAT1 Trailer1 END	One File document type with eight content parts as follows: STX Header1 Detail1 Detail2 Detail3 VAT1 Trailer1 END



Note: These are not the actual names of the content parts. You should not write code to access the content parts directly; instead, use the built-in services provided in the `wm.b2b.edi.tradacoms.doc` folder to access the content parts. For details, see the *webMethods EDI Module Built-In Services Reference*.

Storing File document types in this way enables you to build logic to process individual detail messages yet still be able to easily access related summary information from the header, VAT, and trailer messages.

However, if a single File document contains many detail messages (for example, fifty or more), the Trading Networks Console may not be able to easily display all of the content parts. In this case, you might want to consider using an alternative storage option.

The alternative storage option stores the Transmission and Batch information in the same manner as the default storage option. However, instead of storing all message details in a single File document type, Trading Networks creates a separate File document type for each detail message.

Example:

Message contents	The default storage method creates...	The alternative storage method creates...
STX Header1 Detail1 Detail2 Detail3 VAT1 Trailer1 END	<p>One File document type with eight content parts as follows:</p> <p>STX Header1 Detail1 Detail2 Detail3 VAT1 Trailer1 END</p>	<p>Three file document types (one per detail message), each with six content parts as follows:</p> <p>Document type 1: STX Header1 Detail1 VAT1 Trailer1 END</p> <p>Document type 2: STX Header1 Detail2 VAT1 Trailer1 END</p> <p>Document type 3: STX Header1 Detail3 VAT1 Trailer1 END</p>

To select which storage option to use, set the *TRADACOMS/storageOption* variable in your EDITPA, as described in “[TRADACOMS/storageOption EDITPA Variable](#)” on page 154.

Errors That Prevent the Splitting of Transmissions

The following categories of errors will prevent the *splitOption* EDITPA variable from splitting a TRADACOMS transmission. For information about setting *splitOption*, see “[TRADACOMS/splitOption EDITPA Variable](#)” on page 153.

Validation errors—If the processing rule triggered by a transmission document is set to validate the structure of the document, and if any of the following errors occur, *splitOption* will not split the transmission:

- Data error:

- The message structure is invalid, or a field contains an invalid value, or a mandatory field is missing. These errors are detected by the `wm.b2b.edi:convertToValues` service when its `validate` flag is set to true. This only applies to errors within the STX, BAT, MHD, MTR, EOB, and END segments. Errors within the MHD and MTR segments will *not* prevent `splitOption` from splitting the entire transmission.
- Missing or invalid RSGRSG message:
 - ANAA specified in the STX segment:
 - No RSGRSG message
 - Control number in RSGRSG does not match RSGRSG in STX
 - Sender Reference in RSGRSG does not match RSGRSG in STX
 - ANAA is *not* specified in STX segment:
 - RSGRSG is present
- Document count mismatch:
 - Number of messages or batches between STX/END does not match reported count in END segment
 - Number of messages between BAT/EOB does not match reported count in EOB segment
 - Number of segments between MHD/MTR segments (including the MHD/MTR segments) does not match the reported count in MTR segment
- Invalid MHD sequence number—The MHD segment contains a sequence number. The transmission will only be split if the first MHD segment is 1 and all subsequent MHD segment sequence numbers are incremented by one.

Invalid file structure—When any of the following messages in a file are missing, `splitOption` will not split the transmission:

- Message detail missing
- VAT message missing (if required)
- Message trailer missing

Defining Control Number Information for Partners

■ Overview	164
■ Turning Inbound Control Number Validation On and Off	168
■ Defining Actions for Invalid Control Numbers	170
■ Control Number Cap, Minimum, Increment, and Window	172
■ Defining Control Number Settings	177
■ Searching for Existing Control Number Settings	181

Overview

A *control number* is a number in the header of an ANSI X12 or UN/EDIFACT EDI document that is used for validation and for the ordering of documents exchanged between trading partners. Control numbers can be used to detect duplicate, missing, or out-of-sequence documents.



Note: If you use the TRADACOMS EDI standard, the term *control number* is equivalent to the transmission reference numbers specified in the STX and BAT segments of your TRADACOMS documents. Whether your EDI standard includes control numbers or transmission reference numbers, you define them to Trading Networks in the same manner; the only difference is in the terminology. For simplicity, Trading Networks and the EDI Module use the term *control number* to mean either control number or transmission reference number.

The webMethods EDI Module (EDI Module) maintains information about control numbers in the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database.

The EDI Module maintains control number information for unique combinations of:

- Sender/receiver
- EDI standard and version
- Production mode (e.g., Production or Test)
- Type, which is either:
 - For ANSI X12 or UN/EDIFACT (and all supported sub-standards):
 - Envelope, if the control number is to be used for an interchange header
 - OR-
 - Group, if the control number is to be used for a group header
 - For TRADACOMS:
 - Envelope, if the control number is to be used for a TRADACOMS transmission header
 - OR-
 - Batch, if the control number is to be used for a TRADACOMS batch header, a TRADACOMS file header (e.g., ORDHDR, ACKHDR, INVHDR, etc.), or a message type for a TRADACOMS file type, (e.g., ORDHDR, ACKHDR, INVHDR, etc.).

When the EDI Module is processing an EDI document and requires control number information, the EDI Module uses the above information from the EDI document being processed to look up the control number information to use.

For example, if the EDI Module is validating a group control number, it uses the following information from the group header to locate the EDIControlNumber table entry:

Information from the Group Header	
Sender	A
Receiver	B
EDI standard	ANSI X12
Version	4010
Production Mode	Testing
Group Type	PO

The control number information that the EDI Module maintains in an EDIControlNumber table entry includes the:

- Next expected control number
- Maximum allowed control number (referred to as the control number cap)
- Minimum allowed control number
- Control number increment
- Control number window

For a description of these settings, see “[Control Number Cap, Minimum, Increment, and Window](#)” on page 172.

The EDI Module provides a user interface from the EDI Module home page to allow you to view and manage control number information that it stores in the EDIControlNumber table. From the EDI Module home page, you can:

- Add entries to the EDIControlNumber table for unique combinations of sender/receiver/EDI standard/version/production mode/type.
The EDI Module automatically adds entries to the table when it is validating inbound control numbers. However, if you want to define the first control number to expect, you can add an entry. Additionally, you can add entries that you want to use for outbound processing. For more information, see “[Defining Control Number Settings](#)” on page 177.
- Configure the control number settings (control number cap, minimum, increment, and window). For more information, see “[Defining Control Number Settings](#)” on page 177.

- Update the next expected control number or the control number settings. To do so, first search for the EDIControlNumber table entry that you want to update; then update the settings. For more information, see “[Searching for Existing Control Number Settings](#)” on page 181.

The control number information in the EDIControlNumber table can be used when processing both inbound and outbound EDI documents. For more information, see “[Processing Inbound EDI Documents](#)” on page 166 and “[Processing Outbound EDI Documents](#)” on page 168.

Processing Inbound EDI Documents

For inbound EDI documents, the EDI Module can perform control number validation against the control numbers in the interchange or group headers (or TRADACOMS transmission, batch, or file headers) of the inbound EDI document. By default, inbound control number validation is turned off. You can turn on validation for interchange (or transmission) control numbers and/or group (or batch) control numbers. For more information, see “[Turning Inbound Control Number Validation On and Off](#)” on page 168.

The inbound control number validation determines whether the control numbers are in order, and therefore whether the EDI documents arrived in order. The validation does *not* guarantee that EDI documents are processed in the same order in which documents are received.

To determine whether a control number is valid or invalid, the EDI Module looks up the EDIControlNumber table entry that corresponds to the interchange or group header (or transmission or batch header) of the EDI document it is processing. It then compares the control number from the interchange or group header (or transmission or batch header) to the next expected control number in the EDIControlNumber table entry. The EDI Module determines that the control number from the header is:

- Valid if the control number matches the next expected control number in the EDIControlNumber table entry. When the control number is valid, the EDI Module updates the next expected control in the EDIControlNumber table and continues processing the EDI document.
- Invalid if the control number does not match the next expected control number in the EDIControlNumber table entry. When the control number is invalid, the EDI Module determines whether the invalid control number is:
 - Duplicate control number, which might indicate a duplicate document. A duplicate control number is one that the EDI Module believes has already been used, indicating a duplicate document.

-OR-

- Out-of-sequence control number, which might indicate that there was one or more preceding documents that should have already arrived, but have not. A control number is also considered out-of-sequence if the control number is not numerical.

Because the EDI Module only maintains the next expected control number (not a list of all previously used control numbers) and because the next expected control number can be manually set, the EDI Module might determine that a control number is a:

- Duplicate even though the control number was never before received.
- Out-of-sequence even though the control number has already been received.

For more information about how EDI Module determines whether an invalid control number is duplicate or out-of-sequence, see [“Using the Settings to Determine the Type of Invalid Control Number” on page 175](#).

For more information about processing inbound EDI documents, see [Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”](#) or [Chapter 14, “Processing Inbound TRADACOMS Documents using Trading Networks”](#).

Information You Can Set for Inbound Control Number Validation

The following table describes the control number information that you can define for inbound EDI documents.

Setting	Define in...	Setting is associated with...
Whether you want the EDI Module to validate inbound control numbers. For more information, see “Turning Inbound Control Number Validation On and Off” on page 168 .	EDITPA	sender/receiver pair identified in the EDITPA
Actions you want the EDI Module to take when it encounters an invalid control number. For more information, see “Defining Control Number Settings” on page 177 .		
Settings used to track control numbers and determine whether an invalid control number is a duplicate or out-of-sequence. For more information, see “Control Number Cap, Minimum, Increment, and Window” on page 172 .	EDIControlNumber table	Unique combination of: <ul style="list-style-type: none"> ■ sender/receiver ■ production mode (e.g., Testing or Production) ■ type (Envelope or Group)
Initial value of the control number you expect to receive in an inbound document.		

Processing Outbound EDI Documents

For outbound EDI documents, the service you create to form an outbound EDI document can access the EDIControlNumber table to obtain the control numbers to use for group and interchange headers (or batch and transmission headers) of the document. You can set the initial control number value you want to use for an outbound EDI document. For more information, see [Chapter 16, “Forming EDI Documents to Send Outbound When Using Trading Networks”](#).

Turning Inbound Control Number Validation On and Off

You can have the EDI Module validate interchange control numbers, group control numbers, or both (or any combination of TRADACOMS transmission, batch, and file control numbers). Because you turn validation on or off using EDITPA variables, you can control whether the EDI Module validates control numbers for all sender/receiver pairs or for specific sender/receiver pairs.



Important! The EDI Module does *not* validate control numbers that contain non-numeric characters.

For ANSI X12 or UN/EDIFACT documents (and all supported sub-standards), use these EDITPA variables:

Type of Control Number	How to turn validation on or off
Interchange	<p>Use the EDITPA variable <code>ControlNumberManagement/validateInboundEnvelopeControlNumbers</code>. To turn interchange control number validation:</p> <ul style="list-style-type: none">■ On, set this EDITPA variable to <code>true</code>.■ Off, set this EDITPA to <code>false</code>. <p>For more information, see “ControlNumberManagement/validateInboundEnvelopeControlNumbers EDITPA Variable” on page 124.</p>
Non-standard	<p>For ANSI X12 and UN/EDIFACT users: When you are using non-standard processing, to turn validation of inbound interchange control numbers on or off, use the Validate inbound envelope control numbers setting on the EDI Module home page. For more information, see “Turning Inbound Control Number Validation On or Off” on page 408 in Appendix A, “Non-Standard Processing”.</p>

Type of Control Number	How to turn validation on or off
Group	<p>Use the EDITPA variable <i>ControlNumberManagement/validateInboundGroupControlNumbers</i>. To turn group control number validation:</p> <ul style="list-style-type: none"> ■ On, set this EDITPA variable to <code>true</code>. ■ Off, set this EDITPA variable to <code>false</code>. <p>For more information, see “ControlNumberManagement/validateInboundGroupControlNumbers EDITPA Variable” on page 125.</p>

For TRADACOMS documents, use these EDITPA variables:

Type of Control Number	How to turn validation on or off
Transmission	<p>Use the EDITPA variable <i>TRADACOMS/ControlNumberManagement/validateInboundTransmissionControlNumbers</i>. To turn transmission control number validation:</p> <ul style="list-style-type: none"> ■ On, set this EDITPA variable to <code>true</code>. ■ Off, set this EDITPA to <code>false</code>. <p>For more information, see “TRADACOMS/ControlNumberManagement/validateInboundTransmissionControlNumbers EDITPA Variable” on page 154.</p>
Batch	<p>Use the EDITPA variable <i>TRADACOMS/ControlNumberManagement/validateInboundBatchControlNumbers</i>. To turn batch control number validation:</p> <ul style="list-style-type: none"> ■ On, set this EDITPA variable to <code>true</code>. ■ Off, set this EDITPA variable to <code>false</code>. <p>For more information, see “TRADACOMS/ControlNumberManagement/validateInboundBatchControlNumbers EDITPA Variable” on page 154.</p>

Type of Control Number	How to turn validation on or off
File	<p>Use the EDITPA variable <code>TRADACOMS/ControlNumberManagement/validateInboundFileControlNumbers</code>. To turn file control number validation:</p> <ul style="list-style-type: none">■ On, set this EDITPA variable to <code>true</code>.■ Off, set this EDITPA variable to <code>false</code>. <p>For more information, see "TRADACOMS/ControlNumberManagement/validateInboundFileControlNumbers EDITPA Variable" on page 155.</p>

Defining Actions for Invalid Control Numbers

You can define the actions that you want the EDI Module to take when it encounters a duplicate or out-of-sequence control number. The actions you can specify are:

- **Error & Continue.** The EDI Module logs the error; then continues to process the EDI document that contains the invalid control number normally.
- **Process Normally.** The EDI Module logs a warning; then continues to process the EDI document that contains the invalid control number normally.
- **Reject.** The EDI Module logs the error and does *not* process the document normally. The EDI Module does *not* split the EDI document. Typically, the EDI Module splits an inbound EDI document based on the EDITPA *splitOption* variable and sends the documents it splits out to Trading Networks for processing. However, if you set the action to **Reject**, the EDI Module sends the document without splitting it to Trading Networks.

Additionally, the EDI Module sets the Trading Networks custom attribute EDI Status, which is associated with the rejected document, to either Duplicate Control Number or Out of Sequence Control Number. You can use the custom attribute EDI Status in processing rule criteria. You should create a processing rule to handle the rejected document. For more information, see ["Defining Processing Rules to Handle Documents with Invalid Control Numbers" on page 247](#). You can later force processing of the document if you want. For more information, see ["Reprocessing EDI Documents with Invalid Control Numbers" on page 250](#).

The behavior of these actions is detailed in ["Actions the EDI Module Can Take for Invalid Control Numbers" on page 240](#).

For ANSI X12 or UN/EDIFACT Documents

The following table describes how to define the action you want the EDI Module to take for invalid control numbers in ANSI X12 and UN/EDIFACT documents (and all supported sub-standards):

Type of invalid control number	How to define the action
Duplicate control number	Use the EDITPA variable <i>ControlNumberManagement/duplicateControlNumberAction</i> . For a list of the settings for this EDITPA variable, see “ControlNumberManagement/duplicateControlNumberAction EDITPA Variable” on page 125 .
Out-of-sequence control number	Use the EDITPA variable <i>ControlNumberManagement/outOfSequenceControlNumberAction</i> . For a list of the settings for this EDITPA variable and a description of the processing the EDI Module performs for each, see “ControlNumberManagement/useReverseRouting EDITPA Variable” on page 127 .

Non-standard

When using non-standard processing, the EDITPA variables described in the table above apply *only* to invalid group control numbers. You specify actions for invalid interchange control numbers using the **Interchange Information Detail** screen of the EDI Module home page. For more information, see [“Defining Actions for Invalid Control Numbers” on page 408](#) in [Appendix A, “Non-Standard Processing”](#). For more information about the difference between standard and non-standard processing, see [“Using Standard or Non-Standard Processing” on page 110](#).

For TRADACOMS Documents

The following table describes how to define the action you want the EDI Module to take for invalid control numbers in TRADACOMS documents:

Type of invalid control number	How to define the action
Duplicate control number	Use the EDITPA variable <i>TRADACOMS/ControlNumberManagement/duplicateControlNumberAction</i> . For a list of the settings for this EDITPA variable, see “ TRADACOMS/ControlNumberManagement/duplicateControlNumber Action EDITPA Variable ” on page 155.
Out-of-sequence control number	Use the EDITPA variable <i>TRADACOMS/ControlNumberManagement/outOfSequenceControlNumberAction</i> . For a list of the settings for this EDITPA variable and a description of the processing the EDI Module performs for each, see “ TRADACOMS/ControlNumberManagement/outOfSequenceControl NumberAction EDITPA Variable ” on page 156.

Control Number Cap, Minimum, Increment, and Window

The EDI Module uses these settings (control number cap, minimum, increment, and window) to determine:

- The next control number to expect
- Whether an invalid control number is a duplicate control number or an out-of-sequence control number
- Whether a control number is valid

The table below describes the control number settings and their defaults:

Setting	Description	Default
<i>control number</i>	The next control number to expect. You can set an initial value. EDI Module updates this setting so it is always set to the next expected control number.	1
<i>control number increment</i>	The value you want the EDI Module to use to increment a control number when determining the next expected control number.	1
<i>control number cap</i>	The highest number the control number can be.	999999
<i>control number minimum</i>	The lowest number the control number can be.	1

Setting	Description	Default
<i>control number window</i>	A range of numbers that the EDI Module uses when determining whether an invalid control number is a duplicate control number or an out-of-sequence control number. For more information, see "Using the Settings to Determine the Type of Invalid Control Number" on page 175 .	100

The EDI Module maintains these settings in the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database. You can define these settings using the EDI Module home page. For instructions, see ["Defining Control Number Settings" on page 177](#).

The settings are for unique combinations of the following. In other words, the EDIControlNumber table contains an entry for unique combinations of the following.

- Sender/receiver
- Production mode (e.g., Testing or Production)
- EDI standard and version
- Type, which is either:
 - For ANSI X12 or UN/EDIFACT (and all supported sub-standards):
 - Envelope, if the control number is to be used for an interchange header
 - OR-
 - Group, if the control number is to be used for a group header
 - For TRADACOMS:
 - Envelope, if the control number is to be used for a TRADACOMS transmission header
 - OR-
 - Batch, if the control number is to be used for a TRADACOMS batch header, a TRADACOMS file header (e.g., ORDHDR, ACKHDR, INVHDR, etc.), or a message type for a TRADACOMS file type, (e.g., ORDHDR, ACKHDR, INVHDR, etc.).

For example, when the EDI Module receives an Interchange document that has an interchange header identifying sender A, receiver B, and production mode Production, to locate the control number settings to use, the EDI Module locates the EDIControlNumber table entry for sender A, receiver B, where production mode is Production, and type is "Envelope".

If the EDI Module is validating a control number in an inbound document and an EDIControlNumber table entry does not exist for the combination of sender/receiver, EDI

standard/version, production mode, and type; the EDI Module adds an entry to the table and sets:

- The control number cap, minimum, increment, and window to their defaults
- The next expected control to the sum of the control number from the inbound EDI document plus the control number increment.



Note: It is possible for the EDI Module to encounter more than one sender/receiver pair that maps to the same Trading Networks internal ID. For example, in one document the sender and receiver might be identified by the D-U-N-S number and in another document the same sender and receiver are identified by a mutually defined ID. Both the D-U-N-S number and mutually identified ID for the same Trading Networks profile and therefore the same Trading Networks ID.

Because the EDI Module uses Trading Networks ID to identify senders and receivers in the EDIControlNumber table, both sender/receiver pair combinations correlate to the same row in the EDIControlNumber table. As a result, when the EDI Module receives a document from either sender/receiver pair, it will update the next expected control number in the same row of the EDIControlNumber table. This might not be the expected behavior. The expected behavior might be that the EDI Module maintains separate control number sequences for each sender/receiver pair. If this is the expected behavior, the EDI Module will indicate that control numbers are invalid when they are not because it will be checking control number sequences for two sender/receiver pairs using the same EDIControlNumber entry.

Using the Settings to Determine Whether a Control Number is Valid

To determine whether a control number is valid, the EDI Module compares the control number from the header in an inbound EDI document to the next expected control number in the EDIControlNumber table. The EDI Module determines the control number is:

- **Valid** if the control number in the inbound EDI document matches the next expected control number in the EDIControlNumber table. When the EDI Module determines a control number is valid, it calculates the next expected control number and saves it in the EDIControlNumber table and processes the document normally. For more information, see [“Using the Settings to Determine the Next Expected Control Number” on page 175](#).
- **Invalid** if the control number in the inbound EDI document does *not* match the next expected control number in the EDIControlNumber table. In this case, the EDI Module determines whether the invalid control number is a duplicate or out-of-sequence, and takes the action you define for invalid control numbers. For more information, see [“Using the Settings to Determine the Type of Invalid Control Number” on page 175](#).

For more information about how the EDI Module performs inbound control number validation, see:

- For ANSI X12 or UN/EDIFACT users (and all supported sub-standards): [“Trading Networks Attributes and EDI Documents” on page 196](#) in Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”.
- For TRADACOMS users: [“Trading Networks Attributes and EDI Documents” on page 222](#) in Chapter 14, “Processing Inbound TRADACOMS Documents using Trading Networks”.

Using the Settings to Determine the Next Expected Control Number

To determine the next expected control, the EDI Module adds the *control number increment* to the valid control number. If the result is greater than the *control number cap*, the EDI Module sets the next expected control number to the *control number minimum*.

Using the Settings to Determine the Type of Invalid Control Number

When the EDI Module determines that a control number is *not* valid, it needs to determine whether the invalid control number is a duplicate control number or an out-of-sequence control number.

Duplicate Control Numbers

Generally the EDI Module considers a control number to be a *duplicate control number* if the control number from the header is less than the next expected control number in the EDIControlNumber table. For example, if the EDI Module receives an EDI document with control number 6, but the expected control number is 8.

However, a duplicate control number might be larger than the expected control number when the next expected control number is close to the minimum. The EDI Module uses

the control number window to determine a range of numbers greater than the expected control number that should be considered duplicate, as shown in the example below:

minimum	cap	window	expected	duplicate control numbers
1	999	5	1	999, 998, 997, 996, 995
			2	1, 999, 998, 997, 996
			3	1, 2, 999, 998, 997
			4	1, 2, 3, 999, 998
			5	1, 2, 3, 4, 999
			6	1, 2, 3, 4, 5



Note: Because the EDI Module only maintains the next expected control number (not a list of all previously used control numbers) and because the next expected control number can be manually set, the EDI Module might determine that a control number is a duplicate even though the control number was never before received.

Out-of-Sequence Control Numbers

An *out-of-sequence control number* indicates that there might be missing EDI documents; that is EDI documents that you should have already received have not yet arrived.

Generally a control number is considered out-of-sequence if it is greater than the expected control number. For example, if the EDI Module receives an EDI document with control number 8, but the expected control number is 6, indicating that you did not receive the EDI documents with control numbers 6 and 7.

However, an out-of-sequence control number might be lower than the expected control number as the expected control number approaches the control number cap. EDI Module uses the control number window to determine a range of numbers lower than the expected control number that should be considered out-of-sequence, as shown in the example below:

minimum	cap	window	expected	out-of-sequence control numbers
1	999	5	994	995, 996, 997, 998, 999
			995	996, 997, 998, 999, 1
			996	997, 998, 999, 1, 2
			997	998, 999, 1, 2, 3
			998	999, 1, 2, 3, 4
			999	1, 2, 3, 4, 5



Note: Because the EDI Module only maintains the next expected control number (not a list of all previously used control numbers) and because the next expected control number can be manually set, the EDI Module might determine that a control number is out-of-sequence even though the control number has already been received.

Defining Control Number Settings

This section describes how to define the control number settings described in “[Control Number Cap, Minimum, Increment, and Window](#)” on page 172. The control number settings are for a unique combination of:

- Sender/receiver
- Production mode (e.g., Testing or Production)
- EDI standard and version
- Type, which is either:
 - For ANSI X12 or UN/EDIFACT (and all supported sub-standards):
 - Envelope, if the control number is to be used for an interchange header
 - OR-
 - Group, if the control number is to be used for a group header
 - For TRADACOMS:
 - Envelope, if the control number is to be used for a TRADACOMS transmission header
 - OR-
 - Batch, if the control number is to be used for a TRADACOMS batch header, a TRADACOMS file header (e.g., ORDHDR, ACKHDR, INVHDR, etc.), or a message type for a TRADACOMS file type, (e.g., ORDHDR, ACKHDR, INVHDR, etc.).

When you define the control number settings, the EDI Module adds an entry to the EDIControlNumber table for the sender/receiver, production mode, EDI standard/version, and type combination.

To define the control number settings

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.

- 3 From the EDI Module home page, in the Control Numbers menu of the navigation panel, click Manage Control Numbers.
- 4 Click Add Control Number. The EDI Module displays the following:

ID Pair Information			
Sender ID	<input type="text"/>	Sender Qualifier	<input type="text"/>
Receiver ID	<input type="text"/>	Receiver Qualifier	<input type="text"/>

Control Number Information			
Production Mode	<input type="button" value="Production"/>	Standard	<input type="button" value="X12"/>
Version	<input type="text"/>	Group Type	<input type="radio"/> Envelope <input checked="" type="radio"/> Group <input type="text"/>
Control Number	<input type="text" value="1"/>	Control Number Cap	<input type="text" value="999999"/>
Control Number Minimum	<input type="text" value="1"/>	Control Number Increment	<input type="text" value="1"/>
Control Number Window	<input type="text" value="1000"/>		
Save Changes			

- 5 In the ID Pair Information section of the screen, specify the sender/receiver pair associated with the control number you are setting.

<u>For this field...</u>	<u>Specify</u>
Sender ID	The Trading Networks internal ID for the sender. To determine the internal ID, you can use the <code>wm.b2b.editn:ediPartnerIDToTNPartnerID</code> service. You can run the <code>ediPartnerIDToTNPartnerID</code> service from the webMethods Developer. Copy the resulting internal ID from the bottom of the Results tab and paste it into the field on the WmEDIforTN home page.
Sender Qualifier	Leave this blank.

For this field...	Specify
Receiver ID	The Trading Networks internal ID for the receiver. To determine the internal ID, you can use the <code>wm.b2b.edith:ediPartnerIDToTNPartnerID</code> service.
Receiver Qualifier	Leave this blank.

Non-standard For ANSI X12 and UN/EDIFACT users: When you are using non-standard processing, the values you specify for Sender ID, Sender Qualifier, Receiver ID, and Receiver Qualifier are different. You specify the IDs and EDI ID qualifiers as they appear in the interchange and group headers. For more information, see [“Defining the Control Number Settings” on page 409 in Appendix A, “Non-Standard Processing”](#). For more information about the difference between the two types of processing, see [“Using Standard or Non-Standard Processing” on page 110 in Chapter 9, “Defining Partner Information \(ANSI X12 and UN/EDIFACT\)”](#).

- 6 In the **Control Number Information** section of the screen, specify information that describes when to use the control number and the value to use for the control number.

For this field...	Specify
Production Mode	Optional. The production mode associated with the interchange or group (or transmission or batch) header for which to use the control number. Specify Production, Test, or Custom.

Note: For TRADACOMS, the only valid value is Production.

Standard	Optional. The EDI standard of the documents in which to use the control number. Select X12, UNEDIFACT, EANCOM, UCS, VICS, or TRADACOMS.
Version	The version of the EDI standard. For example, for an ANSI X12 or UN/EDIFACT interchange the version might be 00401 and for a group it might be 4010. For a TRADACOMS batch or transmission, the version can only be 1. For a TRADACOMS file, specify the version of the file document type.

For this field...	Specify
Group Type	<p>Whether the control number is to be used for an interchange (or transmission) header or a group (or batch) header.</p> <ul style="list-style-type: none">■ Select Envelope if you are adding a control number for an interchange-level (or transmission-level) document.■ Select Group if you are adding a control number for a group-level (or batch-level) document. Optionally, you can specify the group type (e.g., PO or IN) or, for TRADACOMS documents you can specify the batch or file type (e.g., INVHDR or ORDHDR).
Control Number	The next control number that you expect for the interchange or group (or transmission or batch) to use.
Control Number Cap	The highest number that the control number can be. If the EDI Module calculates the next expected control number and that number is greater than the control number cap, the EDI Module sets the control number to the value you specify for Control Number Minimum. For information about the Control Number Cap, Control Number Minimum, and Control Number Increment, see “ Using the Settings to Determine the Next Expected Control Number ” on page 175.
Control Number Minimum	The lowest number that the control number can be.
Control Number Increment	The value that the EDI Module uses to increment the control number to determine the next expected control number.
Control Number Window	A number that indicates a range of numbers that the EDI Module uses to determine whether a number is duplicate or out-of-sequence. For more information, see “ Using the Settings to Determine the Type of Invalid Control Number ” on page 175.

Note: When setting the values for Control Number Cap, Control Number Minimum, and Control Number Window, the difference between Control Number Cap and Control Number Minimum must be at least two times as large as the value you specify for Control Number Window.

- 7 Click **Save Changes**. The EDI Module adds the control number information you defined to the EDIControlNumber table.

Searching for Existing Control Number Settings

You can search for existing control number settings to view and/or update the settings.

To search for existing control number settings

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click EDI. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **Control Numbers** menu of the navigation panel, click **Manage Control Numbers**.

Set search criteria			
Sender ID		Sender Qualifier	
Receiver ID		Receiver Qualifier	
Standard	X12 UN/EDIFACT EANCOM	Production Mode	Production Testing Custom
Version		Sort By	Sender ID
Group Type	<input type="checkbox"/> Envelope <input checked="" type="radio"/> All Groups <input type="radio"/> Other Group	Maximum Results	20
Search			

- 4 Specify the following fields to search for a specific control number in the EDIControlNumber table.

 **Note:** You can leave any of these fields blank, or use '%' as a wildcard to indicate any number of characters.

For this field...	Specify
Sender ID	<p>The sender ID for the interchange or group (or transmission or batch). Typically, this is the Trading Networks internal ID for the sender. To determine the internal ID, you can use the <code>wm.b2b.edith:ediPartnerIDToTNPartnerID</code> service.</p> <p>However, if you use the batching feature of the EDI Module, the EDI Module might encounter a sender/receiver pair for which the sender, receiver, or both the sender and receiver do not have a Trading Networks profiles and therefore no Trading Networks internal ID. If either the sender or receiver does not have a Trading Networks internal ID, the EDI Module saves the ID and EDI qualifier for the sender and receiver from the EDI document in the <code>EDIControlNumber</code> table. To search for control numbers for a sender/receiver pair that do not have Trading Networks profiles, specify the ID of the sender in the Sender ID field and the EDI ID qualifier for the sender in the Sender Qualifier field.</p>
Non-standard	For ANSI X12 and UN/EDIFACT users: When you are using non-standard processing, for Sender ID specify the sender ID used in EDI documents, for example, a D-U-N-S number. For more information about the difference between the two types of processing, see “ Using Standard or Non-Standard Processing ” on page 110 in Chapter 9, “Defining Partner Information (ANSI X12 and UN/EDIFACT)”.
Sender Qualifier	If you specified the Trading Networks internal ID for Sender ID , leave this blank. If you are searching for information about a control number that was added by the EDI Module batching feature for a sender that does not have Trading Networks profile, specify the EDI ID qualifier associated with ID you specified for Sender ID .
Receiver ID	The receiver ID for the interchange or group. Typically, this is the Trading Networks internal ID for the receiver. To determine the internal ID, you can use the <code>wm.b2b.edith:ediPartnerIDToTNPartnerID</code> service. However, if you are searching for information about a control number that was added by the EDI Module batching feature and for which either the sender or receiver that does not have a Trading Networks profile, specify the ID of the receiver in the Receiver ID field and the EDI ID qualifier for the receiver in the Receiver Qualifier field.

For this field...	Specify
Non-standard	For ANSI X12 and UN/EDIFACT users: When you are using non-standard processing, for Receiver ID specify the receiver ID used in EDI documents, for example, a D-U-N-S number. For more information about the difference between the two types of processing, see "Using Standard or Non-Standard Processing" on page 110 in Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT)".
Receiver Qualifier	If you specified the Trading Networks internal ID for Receiver ID, leave this blank. If you are searching for information about a control number that was added by the EDI Module batching feature for a receiver that does not have a Trading Networks profile, specify the EDI ID qualifier associated with ID you specified for Receiver ID.
Standard	The EDI standard of the documents in which to use the control number. Select X12, UNEDIFACT, EANCOM, UCS, VICS, or TRADACOMS.
Production Mode	The production mode associated with the interchange (or transmission) or group (or batch) header for which to use the control number. Specify Production, Test, or Custom. By default, the EDI Module uses Production.
<hr/>	
Note: For TRADACOMS, the only valid value is Production.	
<hr/>	
Version	The version of the EDI standard. For example, for an ANSI X12 or UN/EDIFACT interchange the version might be 00401 and for a group it might be 4010. For a TRADACOMS batch or transmission, the version can only be 1. For a TRADACOMS file, specify the version of the file document type.
Sort By	The criteria by which you want to sort the results from the control number search. The possible values include the fields on this screen.

<u>For this field...</u>	<u>Specify</u>
Group Type	<p>Whether the control number is for an interchange or group header (or a TRADACOMS transmission, batch, or file) header.</p> <ul style="list-style-type: none">■ If you want to search for control numbers for interchange or TRADACOMS transmission headers, select Envelope.■ If you want to search for control numbers for group or batch headers for all group or TRADACOMS batch types, select All Groups.■ If you want to search for control numbers for group or TRADACOMS batch headers of a specific type, select Other and in the Group field specify the group type (e.g., PO) or the batch TRADACOMS type BATCH.■ If you want to search for control numbers for TRADACOMS transmissions, select Other and in the Group field specify the TRADACOMS file type.
Maximum Results	Maximum number of matching control numbers that you want the EDI Module to return.

- 5 Click **Search**. The EDI Module returns information for all control numbers that match your selection criteria.

Creating Clients that Send EDI Documents to Trading Networks

■ Overview	186
■ Content Type to Use	186
■ Service the Client Invokes	188
■ Submitting EDI Documents to Trading Networks via HTTP/S	188
■ Submitting EDI Documents to Trading Networks via FTP	189
■ Submitting EDI Documents to Trading Networks via File Polling	191
■ Using a Service to Send Multiple EDI Documents to Trading Networks	191

Overview

If you want to process your EDI documents through webMethods Trading Networks (Trading Networks), create a client to the Integration Server that submits the documents to Trading Networks. For more information about creating clients, see the *webMethods Developer User's Guide*. For more information about sending documents to Trading Networks, see the *webMethods Trading Networks Concepts Guide*.

The client you create must:

- Send the document to the Integration Server using a content type that the EDI Module handles. For more information, see “[Content Type to Use](#)” on page 186.
- Invoke the appropriate service to handle the document.
- Use HTTP, HTTPS, FTP, File Polling, or EDIINT to transport the document to the Integration Server. For more information, see one of the following sections in this chapter:
 - “[Submitting EDI Documents to Trading Networks via HTTP/S](#)” on page 188
 - “[Submitting EDI Documents to Trading Networks via FTP](#)” on page 189
 - “[Submitting EDI Documents to Trading Networks via File Polling](#)” on page 191

For more information about using EDIINT, see the *webMethods EDIINT Module User's Guide*.

To learn more about the use of clients for inbound EDI document processing when you are using Trading Networks with the EDI Module, see Chapter 3, “Using the EDI Module with Trading Networks” of the *webMethods EDI Module Concepts Guide*.

Content Type to Use

The content type your client should use to send the EDI documents to the Trading Networks depends on the type of data you are sending and/or the type of transport you want to use.

If you want the client to...	Use one of the content types described in this section
Send EDI documents without mainframe data using HTTP, HTTPS, FTP, or File Polling	“ Content Type for EDI Documents ” on page 187
Send EDI documents that contain mainframe data using HTTP, HTTPS, FTP, or File Polling	“ Content Types for EDI Documents with Mainframe Data ” on page 187

Content Type for EDI Documents

If your client is using the HTTP, HTTPS, FTP, or File Polling, you typically will use the content type, `application/EDIstream`. When you use `application/EDIstream` and the Integration Server receives the document, it passes the document to the EDI Module content handler as an `InputStream`. The EDI Module content handler forms the pipeline with the variable `edidata` and assigns this variable the pointer of the `InputStream`.

 Note: For backward compatibility, the EDI Module also has content handlers to accept documents with the content types `application/EDI`, `application/X12`, and `application/UNEDIFACT`. With these content types, the EDI Module content handler must convert the document to a String that it places in the pipeline. This can potentially consume a lot of pipeline space and use a significant amount of memory. As a result, it is recommended that you use the content type, `application/EDIstream`, because it conserves system memory.

Content Types for EDI Documents with Mainframe Data

If your EDI document contains mainframe data that have characters at certain points in the document that define the boundary of each record, use one of the content types described in this section. These boundary characters identify returns or new lines. The content handlers for the `application/x-wmedi...` content types remove these characters allowing the Integration Server to properly unwrap and process the mainframe documents. Use the information in the following table to determine the content type you should use:

Use this content type	If you need to removes...
<code>application/x-wmediwrap80</code>	The 81st character from every record in the document.
<code>application/x-wmediwrap132</code>	The 133nd character from every record in the document.
<code>application/x-wmedisemiwrap80</code>	The 81st character from a record <i>only</i> when a segment is longer than 80 characters.
<code>application/x-wmedisemiwrap132</code>	The 132nd character from a record <i>only</i> when a segment is longer than 133 characters.

When you use one of the `application/x-wmedi...` content types, the Integration Server passes the document to the EDI Module content handler as an `InputStream`. The content handler removes the return and new line characters as specified by the content type. Then the content handler forms the pipeline with the variable `edidata` and assigns this variable the pointer of the `InputStream`.

Service the Client Invokes

After the content type handler forms the pipeline, it invokes the service that the client specifies. When your client uses HTTP, HTTPS, FTP, or File Polling, your client should invoke the `wm.tn:receive` service to send the document directly to Trading Networks.

If your client uses EDIINT, see the *webMethods EDIINT Module User's Guide* for information about how to create the client and the service the client is to invoke.

Submitting EDI Documents to Trading Networks via HTTP/S

You can create a client that uses HTTP or HTTPS to post an EDI document to the Trading Networks, specifically to the `wm.tn:receive` service.

The requirements and logic for an HTTP client to submit EDI documents to Trading Networks are almost the same as the requirements and logic described in “[Sending EDI Documents to the Integration Server via HTTP](#)” on page 33. The one difference is the service that the client invokes. To create a client that submits a document to Trading Networks, the client should identify the `wm.tn:receive` service rather than identify a service that you created to process the EDI documents.

For example, part of the logic of the HTTP client is to address the request to the URL of the service you want to invoke. Your client should use an URL similar to the following:

`http://rubicon:5555/invoke/wm.tn/receive`

Example of Input Variables for the `pub.client:http` Service

The following example describes the values that you would set when using the `pub.client:http` service to POST an EDI document to the `wm.tn:receive` service. For a complete description of the `pub.client:http` service, see the *webMethods Integration Server Built-In Services Reference Guide*.

<u>Set this Variable...</u>	<u>Data type</u>	<u>To...</u>
<code>url</code>	String	The URL of the <code>wm.tn:receive</code> service; that is, the service you want to invoke. The following value would invoke the <code>wm.tn:receive</code> service on the “rubicon” server with port number “5555.”
<code>method</code>	String	<code>post</code>

Example: `http://rubicon:5555/invoke/wm.tn/receive`

<u>Set this Variable...</u>	<u>Data type</u>	<u>To...</u>
<i>loadAs</i>	String	Indicates the data type of the input data source. Specify either bytes or stream: <ul style="list-style-type: none"> ■ bytes if the data source is a byte[] ■ stream if the data source is an InputStream
<i>data/string</i>	String	The EDI document that you want to post.
<i>headers</i>	document	An IData object that contains the following:
<u>Variable</u>	<u>Value</u>	
<i>Name</i>	Content-type	
<i>Value</i>	The content type for the document, for example, application/EDISTream	

Also, the client can set any optional HTTP variables, such as authorization information, that are required by your application.

Submitting EDI Documents to Trading Networks via FTP

You can create a client that FTPs an EDI document to the Integration Server's FTP listening port.

The logic for an FTP client to submit EDI documents to Trading Networks is almost the same as the logic described in ["Sending EDI Documents to the Integration Server via FTP" on page 35](#). The one difference is the service that the client invokes. To create a client that submits a document to Trading Networks, the client should identify the `wm.tn:receive` service rather than identify a service that you created to process the EDI documents.

For example, part of the logic of the FTP client is to change to the directory that represents the service you want to invoke. Your client should use the following command:

```
cd \ns\wm\tn\receive
```

Example of Input Variables for the pub.client:ftp Service

The following example describes the values that you would set when using the pub.client:ftp service to FTP an EDI document to the `wm.tn:receive` service. For a complete description of this service, see the *webMethods Integration Server Built-In Services Reference Guide*.

Set this Variable...	Data type	To...
<code>serverhost</code>	String	Name of the machine running the Integration Server.
<code>serverport</code>	String	Port on which the Integration Server listens for FTP requests.
<code>username</code>	String	A valid user name of an Integration Server user account.
<code>password</code>	String	The password for the user name.
<code>command</code>	String	<code>put</code>
<code>dirpath</code>	String	The path representing the <code>wm.tn:receive</code> service; that is the service to invoke: <code>\ns\wm\tn\receive</code>
<code>localfile</code>	String	Name of the source file containing the EDI document, e.g., <code>x12_850</code>
<code>remotefile</code>	String	Name to assign the file on the Integration Server and the content type. Use the following format: <i>filename;content type:content sub-type</i> For example: <code>x12_850;application:EDISTREAM</code>
<code>secure</code>	Document	Indicates whether the FTP session is with a secure FTP server. The variable <code>auth</code> specifies the kind of authentication mechanism to use (SSL, TLS, or TLS-P), and the variable <code>securedata</code> specifies whether the client is sending PROT C (Data Channel Protection Level Clear) or PROT P (Data Channel Protection Level Private).

Submitting EDI Documents to Trading Networks via File Polling

You can use File Polling to submit documents to Trading Networks. The configuration and setup of the File Polling listener to submit EDI documents to Trading Networks is similar to the configuration and setup described in [“Sending EDI Documents to the Integration Server via File Polling” on page 37](#).

There are only two differences:

- To add a File Polling listener port that submits documents to Trading Networks, set the Processing Service to `wm.tn:receive` rather than identify a service that you created to process the EDI documents.
- When setting the value of the Content Type field, you can use any of the content types listed in [“Content Type to Use” on page 186](#).

Using a Service to Send Multiple EDI Documents to Trading Networks

If you create a client that obtains multiple EDI documents (e.g., reading them from file) and in one invocation of your client you want to loop and send each EDI document to Trading Networks, you need to be careful to cleanup the pipeline between sending each document.

To send the document, you use either the `wm.tn:receive` or `wm.tn.doc.xml:routeXml` service. Be sure to drop the `editn_env` variable from the pipeline after each invocation of either of these services.

Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks

■ Overview	194
■ Before You Can Process Inbound EDI Documents	195
■ Specifying EDITPA Variables that Affect Inbound Processing	196
■ Trading Networks Attributes and EDI Documents	196
■ Defining Processing Rules to Process Inbound EDI Documents	206
■ Coding Services to Process Transaction and Group Documents	210

Overview

When webMethods Trading Networks (Trading Networks) receives an EDI document, it passes it to an EDI recognizer that performs the initial processing. The EDI recognizer is installed into Trading Networks when you install the webMethods EDI Module (EDI Module). You can tailor how the EDI recognizer processes an inbound EDI document. For more information, see [“Specifying EDITPA Variables that Affect Inbound Processing” on page 196](#).

To prepare the EDI document for the processing that you want to perform against it, the EDI recognizer splits the original inbound EDI document based on the *splitOption* variable you specify in the EDITPA. The EDI recognizer can split the original EDI document into Interchange, Group, and/or Transaction documents. After forming the Interchange, Group, and Transaction documents, the EDI recognizer sends the documents into normal Trading Networks processing.

You define the processing that Trading Networks performs against the Interchange, Group, and Transaction documents by defining processing rules. Processing rules consists of:

- Criteria that Trading Networks uses to select the appropriate processing rule for a document
- Actions that Trading Networks is to take against the document

To set the criteria, you can use Trading Networks attributes along with other information that Trading Networks records about a document, for example, the TN document type used for the document. This allows you to set up criteria for processing rules, so Trading Networks selects the correct processing rule for your document. For example, you can set up criteria, so Trading Networks selects a processing rule only if it is from specific sender/receivers and is in a specific processing mode (e.g., testing or production). For more information about the document attributes that are available for your use, see [“Trading Networks Attributes and EDI Documents” on page 196](#).

You will typically want to use the Execute a Service action in a processing rule to invoke a service that you create to process the Interchange, Group, or Transaction document. For more information about defining processing rules and creating a service to process an Interchange, Group, or Transaction document, see [“Defining Processing Rules to Process Inbound EDI Documents” on page 206](#).



Note: This chapter describes how to set up basic inbound processing when using Trading Networks. You can also set up to use the following optional inbound processing:

- Validate interchange and/or group control numbers
- Automatically generate functional acknowledgments (FA)

For more information about optional inbound processing, see [Chapter 15, “Optional Inbound Processing When Using Trading Networks”](#).

To learn more about:

- Processing inbound EDI documents including an overview of the processing, see Chapter 3, "Using the EDI Module with Trading Networks" in the *webMethods EDI Module Concepts Guide*.
- Trading Networks document attributes and processing rules, see the *webMethods Trading Networks Concepts Guide* and the *webMethods Trading Networks User's Guide*.

Before You Can Process Inbound EDI Documents

Before you can set up inbound processing for EDI documents, do the following:

- Install the TN document types for the EDI documents that you want to process. For instructions on how to install TN document types for EDI documents, see "[Installing TN Document Types and Creating Flat File Schemas](#)" on page 104.
- Install the flat file schemas that define the structure of the EDI documents to process. The EDI Module uses the flat file schema for parsing, converting, and validating the structure of an inbound EDI document. The flat file schemas are installed when you install the TN document types for EDI documents.
- Optionally, create the flat file schema that defines the structure of an internal-format document. The service you create to process an Interchange, Group, or Transaction document might create an internal-format document (e.g., the format required by a back-end system). The EDI Module provides a service that you can invoke to create the internal-format document based on the flat file schema. Use the *webMethods Developer* to create the flat file schema. For more information, see the *Flat File Schema Developer's Guide*.
- Define profiles for the senders and receivers in the inbound EDI document. For instructions on how to create profiles, see "[Defining Trading Networks Profiles](#)" on page 113 and the chapter about profiles in the *webMethods Trading Networks User's Guide*.
- Define the default and optionally partner-specific EDITPAs for the sender/receiver pairs in the EDI document. For instructions on how to create the EDITPAs, see "[Defining EDI Trading Partner Agreements](#)" on page 116. For information about the variables in the EDITPAs that affect inbound processing, see "[Specifying EDITPA Variables that Affect Inbound Processing](#)" on page 196.

Non-standard

When using non-standard processing, you also have to define the interchange sender/receiver pair information. For instructions on defining this information, see "[Defining Interchange-Level Sender/Receiver Pair Information](#)" on page 390 in Appendix A, "[Non-Standard Processing](#)". For more information about the difference between standard and non-standard processing, see "[Using Standard or Non-Standard Processing](#)" on page 110.

Specifying EDITPA Variables that Affect Inbound Processing

To tailor how you want the EDI Module to perform inbound processing, you use EDITPA variables. The EDI Module processes an inbound EDI document one interchange segment at a time.

For each interchange segment in an inbound EDI document, the EDI Module obtains the EDITPA values to use for interchange sender/receiver pair. The EDI Module uses those values when processing all documents (Interchange, Group, and Transaction) for the interchange segment.

For a complete list and description of the EDITPA variables, see ["wm.b2b.editn.TPA:EDITPA IS Document Type" on page 118](#).

Non-standard

When using non-standard processing, the EDI Module uses EDITPAs for group-level sender/receiver pairs and additionally uses settings that you define from the EDI Module home page. See ["Variables that Affect Inbound Processing" on page 411](#) in [Appendix A, "Non-Standard Processing"](#). For more information about the difference between standard and non-standard processing, see ["Using Standard or Non-Standard Processing" on page 110](#).

Trading Networks Attributes and EDI Documents

Trading Networks attributes identify selected content from a document that can be used for later processing. For example, you can set up processing rule criteria to select a processing rule based on the value of an attribute. Trading Networks supports two types of attributes: 1) system attributes that are defined by Trading Networks out-of-the-box and 2) custom attributes that are additional attributes that are added to Trading Networks. The EDI Module defines some custom attributes for EDI Module use. Additionally, you can define your own custom attributes and assign them values.

At run time when processing an EDI document, the EDI Module sets values for some of the system attributes and custom attributes using information from the EDI document being processed. If you want to define your own custom attributes, you must write code to set the values of those custom attributes.

You can use the attributes to:

- Process EDI documents using content-based processing rules; that is, select a processing rule for a document based on the value of the attributes extracted from the document
- Search for saved documents based on attribute values
- Report on documents based on attribute values

System Attributes that the EDI Module Sets

The EDI Module sets the DocumentID, GroupID, and ConversationID system attributes using information from the EDI document. The EDI Module uses different values based on whether it is assigning the attribute to an Interchange, Group, or Transaction document. The sections below describe how the EDI Module sets each attribute for each type of document (Interchange, Group, and Transaction) and gives examples that use the sample below:

Control numbers in ANSI X12 ISA, GS, and ST headers

```
ISA*00*      *00*      *01*123456789  *ZZ*987654  *030423*1810*U*00200 000005334*0*P>1
GS*PO*186704136*138183702*030423*1810*1*X*003040\ 
ST*850 0001
```

DocumentID

The DocumentID system attribute is an identifier of the document. The EDI Module sets the DocumentID as follows:

Type of Document	Value used for DocumentID system attribute
Interchange	The EDI Module uses the interchange control number. For example, for the above sample, DocumentID would be set 000005334, which is the value of ISA13.
Group	The EDI Module uses the group control number. For example, for the above sample, DocumentID would be set to 1, which is the value of GS06.
Transaction	The EDI Module uses the transaction control number. For example, for the above sample, DocumentID would be set to 0001, which is the value of ST02.

GroupID

The EDI Module hierarchically assigns the GroupID system attribute, so that by viewing the GroupID, you can tell to which envelope an element originally belonged. The EDI Module sets the GroupID as follows:

Type of Document	Value used for GroupID system attribute
Interchange	The EDI Module uses the interchange control number. For example, for the above sample, GroupID would be set 000005334, which is the value of ISA13.

Type of Document	Value used for GroupID system attribute
Group	The EDI Module uses the interchange control number, so you can determine to which interchange the Group document belongs. For example, for the above sample, GroupID would be set 000005334, which is the value of ISA13.
Transaction	The EDI Module uses the group control number, so you can determine to which group the Transaction document belongs. For example, for the above sample, GroupID would be set to 1, which is the value of GS06.

ConversationID

The *ConversationID* system attribute is an identifier that links all documents that are part of the same business process (also called a conversation). That is, all documents in the same business process need to have the same *ConversationID*.

As described in the *webMethods Trading Networks User's Guide*, Trading Networks can extract *ConversationIDs* from EDI Envelope, Group, and Transaction documents and use them to pass documents to the *webMethods Process Run Time* (PRT) after Trading Networks performs the actions identified by a processing rule. The Process Run Time is the facility of the Integration Server that executes and manages business processes in the Integration Server.

By default Trading Networks does *not* extract the *ConversationID* from EDI documents.

Enabling the Extraction of *ConversationIDs* From EDI Documents

To enable the extraction of *ConversationIDs* from EDI documents, so that Trading Networks can pass them to the *webMethods Process Run Time* (PRT), do the following.

To enable the extraction of *ConversationIDs* from EDI documents

- 1 Obtain a license for the *webMethods Process Run Time* (PRT) and install it, if it is not already installed.
- 2 For Transaction document types, enable the EDI Module to assign *ConversationIDs* by creating an *instance ID query* that identifies the value to set for the conversation ID. You define instance ID queries for specific transaction sets (e.g., X12 4010 850) using the WmEDIforTN home page. For details, see “[Defining an Instance ID Query for a Transaction](#)” on page 354 in Chapter 20, “[Including Documents in a Business Process](#)”.

 Note: By default the EDI Module assigns the Interchange and Group documents conversation IDs. As a result, because the Interchange and Group documents have a conversation ID, they are passed to the Process Run Time (PRT).

- 3 Using webMethods Developer, add a dependency in the WmEDIforTN package on the WmPRT package.
- 4 Configure a PRT database connection. Follow the PRT installation instructions to configure a data source.
- 5 Reload the WmEDIforTN package.

Disabling the Extraction of *ConversationIDs* From EDI Documents

By default Trading Networks does *not* extract the *ConversationID* from EDI documents. However, if this default has been overridden, you can disable the extraction of *ConversationIDs* from EDI documents as follows.

To disable the extraction of *ConversationIDs* from EDI documents

- 1 For Envelope and Group document types, set the property `disableCIDSsupport=true` in the `webMethods6\IntegrationServer\packages\WmEDI\config\properties.cnf` file.
- 2 For Transaction document types, use the webMethods Developer to remove the dependency in the WmEDIforTN package on the WmPRT package.
- 3 Reload the WmEDIforTN package.

To learn more about:

- An overview of processing EDI documents in business processes, including an illustration of how EDI documents are passed to a business process, see Chapter 4, "EDI Documents in Business Processes", in the *webMethods EDI Module Concepts Guide*.
- The actions you need to take so the EDI Module can set a value for the *ConversationID* and about including EDI documents in business processes, see [Chapter 20, "Including Documents in a Business Process"](#).

Custom Attributes that the EDI Module Sets

The EDI Module sets the value of the following custom attributes that you might want to use in processing rule criteria. These custom attributes are added to Trading Networks when you install the EDI Module.

Custom Attribute	Description
EDI Group Type	For Group documents, the EDI Module sets this attribute to the value of the functional identifier (e.g., PO, or RC) from the group header.
EDI Outbound FA	<p>The EDI Module sets this attribute to indicate whether a document is an outbound FA that it generated. The value of the EDI Outbound FA custom attribute is set to:</p> <ul style="list-style-type: none"> ■ <code>true</code>, for an outbound FA that the EDI Module generated ■ <code>false</code>, for all other documents
EDI Processing Mode	<p>The EDI Module sets this attribute to <code>Testing</code>, <code>Production</code>, or <code>Custom</code> based on the value of the EDITPA <code>processingMode</code> variable. For more information, see “processingMode EDITPA Variable” on page 122.</p>
EDI Status	<p>The EDI Module sets this attribute based on the outcome of optional inbound processing, that is, inbound control number validation and automatic FA generation. The EDI Module only performs this optional inbound processing when the EDITPA indicates to do so. For more information, see Chapter 15, “Optional Inbound Processing When Using Trading Networks”.</p> <p>If EDITPA indicates to do so, the EDI Module first performs inbound control number validation. If the EDI Module determines that a document contains:</p> <ul style="list-style-type: none"> ■ Duplicate control number, it sets this attribute to <code>Duplicate Control Number</code>. ■ Out-of-sequence control number, it sets this attribute to <code>Out of Sequence Control Number</code>.

Custom Attribute	Description
	<p>Whether control number validation was performed or not, if the EDITPA indicates to automatically generate FAs, the EDI Module generates FA. If the EDI Module encounters unacceptable FA statuses, it sets the EDI Status attribute based on the unacceptable FA status; that is, to one of the following:</p> <ul style="list-style-type: none"> ■ Generate FA - Not Allowed ■ Generate FA - Rejected ■ Generate FA - Partially Accepted ■ Generate FA - Accepted, But Errors Were Noted <p>If the control numbers are valid -and- the FA status is acceptable, the EDI Module sets the EDI Status attribute to Processed. Additionally, if the EDI Module neither validates control number validation nor generates FAs, it sets the EDI Status attribute to Processed.</p>
EDI Version	The EDI Module sets this attribute to the version of the EDI standard that the document uses (e.g., 4010).
Envelope CntrlNum Status	These attributes indicate the control number validation status of documents, thus enabling child documents to know the validation status of their parent documents.
Group CntrlNum Status	<p>When validating document control numbers, the EDI Module sets this attribute to Valid, Duplicate, Out of Sequence, or Not Validated.</p> <ul style="list-style-type: none"> ■ For interchange documents, the EDI Module only sets the attribute Envelope CntrlNum Status. ■ For group documents, the EDI Module sets the following attributes: <ul style="list-style-type: none"> ■ Envelope CntrlNum Status (indicates the status of the associated interchange document) ■ Group CntrlNum Status (indicates the status of the associated group document)

Custom Attribute	Description
	<ul style="list-style-type: none"> ■ For transaction documents, the EDI Module sets the following attributes: <ul style="list-style-type: none"> ■ Envelope CntrlNum Status (indicates the status of the associated interchange document) ■ Group CntrlNum Status (indicates the status of the associated group document)
	<p>Note: For UN/EDIFACT transaction documents the EDI Module may also set this attribute to Not Present.</p>
EDI FA Status	<p>Indicates the status of a functional acknowledgment (FA) that the EDI Module has sent to the sender of an inbound document.</p> <p>For inbound ANSI X12 group documents and UN/EDIFACT interchange documents, this attribute may return the following values:</p> <ul style="list-style-type: none"> ■ Not Acknowledged—The EDI Module has not yet received or sent an FA to acknowledge this document. ■ Duplicate—The EDI Module has one or more other documents recorded in the EDITRACKING table that match the FA for this document. ■ Accepted—The EDI Module received a single FA that matches this document and the FA has either: <ul style="list-style-type: none"> ■ An “A”(Accept) status on the confirmed level (ANSI X12) ■ A “7” status on the confirmed level (UN/EDIFACT) ■ Accepted w/Errors—The EDI Module received a single FA that matches this document and the FA has an “E” status (Errors) on the confirmed level. ■ Accepted - Partial—Used only for group documents. The EDI Module received a single FA that matches this document and the FA has an “P” status (Partially Accept) on the confirmed level.

Custom Attribute	Description
	<ul style="list-style-type: none"> ■ Rejected—The EDI Module received a single FA that matches this document and the FA has either: <ul style="list-style-type: none"> ■ An “R” (Reject) status on the confirmed level (ANSI X12) ■ An “4” status on the confirmed level (UN/EDIFACT) ■ FA Error—The EDI Module encountered other errors, such as unknown or invalid FA status. ■ Disabled—The <i>FAReconciliation</i> EDITPA variable is set to <code>false</code>, which disables FA reconciliation and reporting. For more information, see “FAReconciliation EDITPA Variable” on page 136.

For functional acknowledgments (e.g., ANSI X12 997s and UN/EDIFACT CONTRL documents), this attribute may return the following values:

- **Matched**—A matching document exists in the EDITRACKING table.
- **Unmatched**—No matching document exists in the EDITRACKING table.
- **Duplicated**—The EDI Module has one or more other documents recorded in the EDITRACKING table that match the FA for this document.
- **Disabled**—The *FAReconciliation* EDITPA variable is set to `false`, which disables FA reconciliation and reporting. For more information, see [“FAReconciliation EDITPA Variable” on page 136](#).
- **DUP_FA**—The EDI Module received more than one FA that matches this document.

Note: Be sure to turn on automatic FA generation, as described in [“Turning Inbound Control Number Validation On and Off” on page 168](#). Otherwise, the FA Status attribute will always return Not Acknowledged or Disabled.

Defining Your Own Custom Attributes for EDI Documents

You can create custom document attributes and assign them to TN document types for EDI documents. At run time, services that you create must set the values of the attributes.

At Design Time Creating Attributes and Associating Them with TN Document Types

Perform the following procedure to associate custom attributes with EDI documents.

To associate a custom attribute with an EDI document

- 1 From the WmEDIforTN home page, install the TN document type for the type of EDI document with which you want to associate the custom attribute. For more information about how to complete this step, see “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104.
- 2 In the Trading Networks Console, create the custom attribute(s). For more information about how to complete this step, see the document attributes chapter in *webMethods Trading Networks User’s Guide*.
- 3 Invoke the `wm.b2b.editn:addAttributeTypeToBizDoc` service to associate the custom document attribute that you created in [step 2](#) with the TN document type you installed in [step 1](#). For more information about this service, see the *webMethods EDI Module Built-In Services Reference*.

Setting the Values of Attributes at Run Time

You must create a service to set the attribute values at run time. To have your service executed at run time, you can use it in the Execute a Service processing action of a Trading Networks processing rule. Note that your EDI document type or processing rule must use the Save Document to Database pre-processing action to save the document content and attributes to the database.

Create a service that performs the following logic:

- Obtain the values you want to use for each custom attribute.
- For each attribute, invoke the `wm.tn.doc:setAttribute` service to set the value of the attribute in the BizDocEnvelope.
- Invoke the `wm.tn.doc:updateAttributes` service to update the values of the attributes in the saved copy of the document that is in the Trading Networks database.

Example

When working with an EDI ANSI X12 850 document, you might want to associate the *PurchaseOrderNumber* attribute with the 850 document. To do so, you would:

- 1 Install the TN document type for the 850 EDI document.
- 2 Create the *PurchaseOrderNumber* attribute.
- 3 Invoke the `wm.b2b.editn:addAttributeTypeToBizDoc` service to associate the *PurchaseOrderNumber* attribute with the TN document type for the 850 EDI document.
- 4 Create a processing rule that invokes a service that extracts the value for the *PurchaseOrderNumber* attribute from the 850 EDI document. Your service should invoke the `wm.tn.doc:setAttribute` service to set the value for the attribute in the `BizDocEnvelope`, and then invoke the `wm.tn.doc:updateAttributes` service to update the attributes in the saved copy of the document in the Trading Networks database.

Defining a Processing Rule that Uses Your Custom Attribute as Criteria

To use the custom attributes you defined as criteria in a processing rule, you need to:

- Associate custom attributes with EDI document types (as described in [“At Design Time Creating Attributes and Associating Them with TN Document Types” on page 204](#) in this chapter).
- Create a processing rule to process the document. This processing rule does not use content-based processing. Leave the **Extended Criteria** tab of the **Processing Rules** screen blank. For this processing rule, use the **Execute a Service** processing rule to invoke a service that sets the values of the custom attributes (as described in [“Setting the Values of Attributes at Run Time” on page 204](#)). Additionally, this service should invoke the `wm.tn.reroute` service to have Trading Networks select another processing rule for the document. The next processing rule can use your custom attributes as criteria.
- Create a processing rule that uses specifies the custom attribute as criteria on the **Extended Criteria** tab.

 Note: When you order your processing rules, be sure to list the processing rule that uses your custom attributes as criteria *before* the processing rule that sets the attribute values. In this way, Trading Networks will skip the processing rule that uses the custom attribute criteria the first time the document is processed, but select it the second time.

Defining Processing Rules to Process Inbound EDI Documents

To specify the processing that you want to perform against Interchange, Group, and Transaction documents that the EDI Module has split from the original inbound EDI document, you define processing rules.

When you define a processing rule, you define the criteria that you want Trading Networks to use to select the processing rule, and the pre-processing and processing actions that you want Trading Networks to perform against the document.

The processing rules you need to define are based on the *splitOption* that you specify in the EDITPA.

If you set <i>splitOption</i> to...	Then define processing rules for...
Interchange	<ul style="list-style-type: none">■ Interchange document
Group	<ul style="list-style-type: none">■ Interchange document■ Group documents
Transaction	<ul style="list-style-type: none">■ Interchange document■ Group documents■ Each type of Transaction document (e.g., ANSI X12 850, ANSI X12 810, UN/EDIFACT INVOIC, or UN/EDIFACT ORDERS)

To define a processing rule

You define processing rules using the Trading Networks Console. For steps to define processing rules, see the chapter about processing rules in the *webMethods Trading Networks User's Guide*.

See the sections below for more information about setting the criteria, pre-processing actions, and processing actions for EDI documents.

Specifying Processing Rule Criteria

When you create a processing rule, you define criteria that Trading Networks uses to select the appropriate processing rule for an Interchange, Group, or Transaction document. The following lists the criteria you can specify.

Use this criteria	To have Trading Networks select a processing rule based on...														
Sender	<p>The sender of the document.</p> <p>The EDITPA variables, <i>GSRouting/routingMode</i> and <i>GSRouting/senderQualifier</i> can affect the sender that the EDI Module uses for an Interchange, Group, and Transaction. For more information, see “GSRouting/routingMode EDITPA Variable” on page 120 and “GSRouting/senderQualifier EDITPA Variable” on page 121.</p>														
Receiver	<p>The receiver of the document.</p> <p>The EDITPA variables, <i>GSRouting/routingMode</i> and <i>GSRouting/receiverQualifier</i> can affect the receiver that the EDI Module uses for an Interchange, Group, and Transaction. For more information, see “GSRouting/routingMode EDITPA Variable” on page 120 and “GSRouting/receiverQualifier EDITPA Variable” on page 122.</p>														
Document Type	<p>The TN document type for the document.</p> <p>Use this criterion to process a document based on whether it is an Interchange, Group, or Transaction document.</p> <p>Examples</p> <table> <thead> <tr> <th style="text-align: left;">To have Trading Networks select a processing rule for...</th> <th style="text-align: left;">Set the Document Type criterion to this TN document type...</th> </tr> </thead> <tbody> <tr> <td>ANSI X12 Interchange document</td> <td>X12 Envelope</td> </tr> <tr> <td>ANSI X12 Group document</td> <td>X12 Group</td> </tr> <tr> <td>ANSI X12 850 Transaction document of version 4010</td> <td>X12 4010 850</td> </tr> <tr> <td>UN/EDIFACT Interchange document</td> <td>UNEDIFACT Envelope</td> </tr> <tr> <td>UN/EDIFACT Group document</td> <td>UNEDIFACT Group</td> </tr> <tr> <td>UN/EDIFACT ORDERS Transaction document of version 99A</td> <td>UNEDIFACT 99A ORDERS</td> </tr> </tbody> </table>	To have Trading Networks select a processing rule for...	Set the Document Type criterion to this TN document type...	ANSI X12 Interchange document	X12 Envelope	ANSI X12 Group document	X12 Group	ANSI X12 850 Transaction document of version 4010	X12 4010 850	UN/EDIFACT Interchange document	UNEDIFACT Envelope	UN/EDIFACT Group document	UNEDIFACT Group	UN/EDIFACT ORDERS Transaction document of version 99A	UNEDIFACT 99A ORDERS
To have Trading Networks select a processing rule for...	Set the Document Type criterion to this TN document type...														
ANSI X12 Interchange document	X12 Envelope														
ANSI X12 Group document	X12 Group														
ANSI X12 850 Transaction document of version 4010	X12 4010 850														
UN/EDIFACT Interchange document	UNEDIFACT Envelope														
UN/EDIFACT Group document	UNEDIFACT Group														
UN/EDIFACT ORDERS Transaction document of version 99A	UNEDIFACT 99A ORDERS														

Use this criteria	To have Trading Networks select a processing rule based on...
Recognition Errors	Whether Trading Networks or the EDI recognizer encountered any errors during the recognition process.
Extended Criteria	<p>The value of custom attributes.</p> <p>For example, on the Extended Criteria tab of the Processing Rules Detail screen, you might specify the EDI Processing Mode attribute must have the value Production for Trading Networks to use the processing rule. For more information about the custom attributes that the EDI Module sets, see “Custom Attributes that the EDI Module Sets” on page 200. For information about how to set your own custom attributes, see “Defining Your Own Custom Attributes for EDI Documents” on page 204.</p>



Note: The EDI Module does not set the User Status system attribute, so you should not use the User Status criterion.

Specifying Pre-Processing Actions

Default pre-processing actions are defined in TN document types. However, you can override the settings in the processing rule.

To learn more about how the pre-processing actions apply to EDI documents, see the section about pre-processing actions in Chapter 3, "Using the EDI Module with Trading Networks" in the *webMethods EDI Module Concepts Guide*.

The following table lists the pre-processing actions you can set in a processing rule and the default settings for each action as set in the TN document types for EDI documents.

Pre-processing action	Description	Default setting in TN document type for the EDI document
Validate Structure	Validates the structure of the EDI document.	Validate the structure of the envelope
Check for Duplicate Document	Determines if Trading Networks already has this document in its database.	Do not use Trading Networks check for duplication
Save Document to Database	Saves a copy of the document content, attributes, and/or activity log information to the Trading Networks database.	Save document content, attributes, and activity log

If a pre-processing action fails, Trading Networks records the error and continues processing. Trading Networks records the error in the *errors* variable in the BizDocEnvelope, which is in the *bizdoc* pipeline variable. Subsequent processing that you add, for example a service invoked by the Execute a Service processing action, can access the error information.

 Note: You cannot use the Verify Digital Signature pre-processing action because values for the SignedBody and Signature system attributes are not set for EDI documents.

Specifying Processing Actions

You can use all of the Trading Networks processing actions for EDI documents.

- If you want to process the information in the transactions of an inbound EDI document, for example to use information from Transaction documents to form a documents that is to go to a back-end system, you will mainly use the Execute a Service action. You create the service that the Execute a Service processing action invokes.
In this situation, you would want to set the EDITPA variable *splitOption* to Transaction or Group.
 - For the Transaction and/or Group documents that result from the split, you would define processing rules that use the Execute a Service processing action to invoke a service that acts on the Transaction and/or Group documents. For more information about how to create the services, see "[Coding Services to Process Transaction and Group Documents](#)" below.
 - For the Interchange document, you could define a processing rule that has no processing actions selected, which in effect causes Trading Networks to ignore the document. There is no need to process this document. Alternatively, you could set up your processing rules to allow the Interchange document to fall through to the Default rule, which only sets the User Status system attribute of the document to IGNORED. If you use the Default rule, you do not need to define a processing rule for the Interchange document.
- If you are sending the inbound EDI document through Trading Networks to simply deliver it to a destination without processing individual transactions, you will mainly use the Deliver Document By action. In this situation, you set the EDITPA variable *splitOption* to Interchange. In this case, the only document sent to Trading Networks is the Interchange document. For more information about delivering documents, see [Chapter 16, "Forming EDI Documents to Send Outbound When Using Trading Networks"](#).

Coding Services to Process Transaction and Group Documents

The logic you create in the service that the Execute a Service action invokes depends on the types of documents that the EDI Module is splitting from the original EDI document (based on the EDITPA variable *splitOption*).

- If the *splitOption* is Transaction, the EDI Module creates Transaction, Group, and Interchange documents from the original EDI document.
 - For the Transaction document, create a service that processes the single transaction set contained in the Transaction document. For example form an internal-format document based on the transaction set information and send the internal-format document to an internal application, e.g., a back-end system. For information about how to create this service, see “[Logic to Process a Transaction Document](#)” on page 213.
 - For the Group and Interchange documents, you do not need to create a service. All processing for these EDI documents would be complete in the processing of the Transaction documents. You could set up processing rules for the Group and Interchange documents that have no processing actions selected or set up your processing rules so the Interchange document falls through to the Default rule.



Note: If you want to generate functional acknowledgments (FAs), see “[Automatically Generating Functional Acknowledgements](#)” on page 258 in Chapter 15, “Optional Inbound Processing When Using Trading Networks”.

- If the *splitOption* is Group, the EDI Module creates Group and Interchange documents from the original EDI document.
 - For the Group document, create a service that processes each transaction within the group. For information about how to create this service, see “[Logic to Process a Group Document when splitOption is Group](#)” on page 215.
 - For the Interchange document, you do not need to create a service. All processing for the EDI document would be complete in the processing of the Group document. You could set up a processing rule for the Interchange document that selects no processing actions or set up your processing rules so the Interchange document falls through to the Default rule.



Note: If you want to generate functional acknowledgments (FAs), see “[Automatically Generating Functional Acknowledgements](#)” on page 258 in Chapter 15, “Optional Inbound Processing When Using Trading Networks”.

- If the *splitOption* is Interchange, the EDI Module creates Interchange documents from the original EDI document. You would set the *splitOption* to Interchange if you wanted to simply deliver the EDI document to its receiver. For more information about how to deliver a document, see “[Delivering the EDI Document](#)” on page 291.

The services you create can use information that is in the pipeline. For information about the data that is in the pipeline when your service is invoked, see “[Information in the Pipeline that Your Service Can Access](#)” below.

Information in the Pipeline that Your Service Can Access

When an Interchange, Group, or Transaction document is passed to the processing rule, the following information is in the pipeline and is accessible by your service:

- BizDocEnvelope in the *bizdoc* variable. Use the BizDocEnvelope to retrieve information that Trading Networks maintains about the document. The BizDocEnvelope adhere to the `wm.tn.rec:BizDocEnvelope` IS document type. It is also an instance of `com.wm.app.tn.doc.BizDocEnvelope`. For details about the information in the BizDocEnvelope, see the *webMethods Trading Networks Built-in Services Reference*. The following lists some of the variables within the BizDocEnvelope:
 - *DocumentID* contains the EDI control number from the interchange, group, or transaction header as described in “[DocumentID](#)” on page 197.
 - *Attributes* contains the custom attributes that were set for the document. For information about the custom attributes that the EDI Module sets, see “[Custom Attributes that the EDI Module Sets](#)” on page 200.
 - *errors* contains the errors that Trading Networks and EDI Module encountered while processing the document.
- EDI document summary information in the *envelopeDocuments* variable. When Trading Networks originally receives the EDI document it passes it to the EDI recognizer, which parses the EDI document. This EDI document summary is the result of the parse. It is the entire original EDI document as an `IData` object.

You can use the EDI document summary information to derive a total document count or to determine when all of the documents in the original EDI document have completed processing. To do so, your service can perform a simple count of the original documents and a count of the documents processed. When these counts match, processing of the entire original EDI document is complete. You also can perform a query within Trading Networks that checks the status of each document using the Trading Networks internal ID for each document.

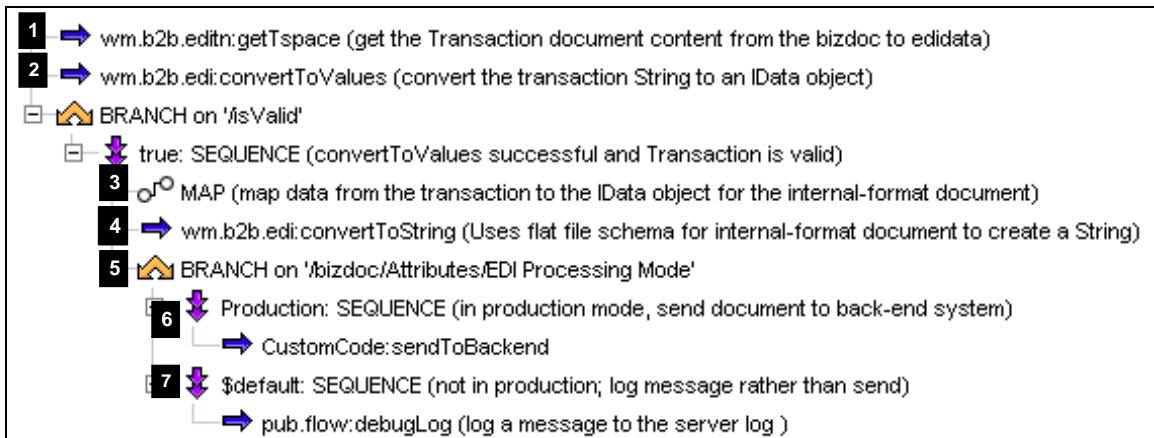
The following shows the structure of the EDI document summary information in the *envelopeDocuments* variable:

Variable	Description
<i>envelopeDocuments</i>	document list Summary of the interchange document including the groups and transaction sets that it contains.
Variable in <i>envelopeDocuments</i>	Description
<i>docID</i>	String Internal ID that Trading Networks generated for the Interchange document.
<i>docTypeID</i>	String Internal ID of the TN document type used for the Interchange document.
<i>docTypeName</i>	String Name of the TN document type used for the Interchange document, e.g., X12 Envelope.
<i>groupDocuments</i>	document list (optional) Summary of all the group documents in the interchange.
Variable in <i>groupDocuments</i>	Description
<i>docID</i>	String Internal ID that Trading Networks generated for the Group document.
<i>docTypeID</i>	String Internal ID of the TN document type used for the Group document.
<i>docTypeName</i>	String Name of the TN document type used for the Group document, e.g., X12 Group.
<i>transactionDocuments</i>	document list (optional) Summary of all the transaction documents in the group.
Variable in <i>transactionDocuments</i>	Description
<i>docID</i>	String Internal ID that Trading Networks generated for the Transaction document.
<i>docTypeID</i>	String Internal ID of the TN document type used for the Transaction document.
<i>docTypeName</i>	String Name of the TN document type used for the Transaction document, e.g., X12 4010 850.

Logic to Process a Transaction Document

The following shows sample code that includes logic you might want to include to process a Transaction document that contains a single transaction set from an inbound EDI document. The processing in the service below shows how to map information from the transaction set into an internal-format document and send the document to an internal application, i.e., a back-end system.

Sample code for processing a Transaction document



Flow operation	Description
1	<p>Invoke the <code>wm.b2b.edtn:getTspace</code> service to retrieve the content of the Transaction document from the BizDocEnvelope (in the <code>bizdoc</code> variable) and place it into the <code>edidata</code> variable.</p> <p>You can use the <code>getTspace</code> service whether the document is considered large or not. For more information about large document handling, see Chapter 7, “Handling Large Documents” and Chapter 19, “Handling Large Documents When Using Trading Networks”.</p> <p>For more information about the <code>getTspace</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>

Flow operation	Description
2	<p>Invoke the <code>wm.b2b.edi:convertToValues</code> service to:</p> <ul style="list-style-type: none"> ■ Convert the content of the Transaction document (in the <code>edidata</code> variable) from either String or InputStream format to an IData object. ■ Validate the structure of the EDI transaction.
	<p>The inputs to the <code>convertToValues</code> service includes the flat file schema for the EDI transaction. The <code>convertToValues</code> service uses the flat file schema to both determine how to parse the transaction set into an IData object and to validate its structure.</p>
	<p>For more information about the <code>convertToValues</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
3	<p>Map the data from the EDI transaction set into the internal-format document.</p> <p>Now that the transaction set contents is an IData object, you can access the data in the transaction set to map it to an IData object for the internal-format document. Depending on the complexity of your mapping requirements, you might need to add more logic than a MAP flow operation, or create a separate service to perform the mapping. For more information about how to map, see Chapter 5, “Mapping Data to Form New Documents”.</p>
4	<p>Invoke the <code>wm.b2b.edi:convertToString</code> service to convert the internal-format document from an IData object to String format.</p> <p>The inputs to the <code>convertToString</code> service include the IData object that contains the data for your internal-format document and the flat file schema for the internal-format document. The <code>convertToString</code> service uses the flat file schema to determine how to form the internal-format document. Alternatively, rather than use a flat file schema, the <code>convertToString</code> service also accepts an IS document type to define the structure of the internal-format document.</p>
	<p>For more information about the <code>convertToString</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
5	<p>Branch based on the value of the EDI Processing Mode custom attribute. This attribute is in the <code>bizdoc/Attributes/EDI Processing Mode</code> variable. The EDI Module sets the value of the EDI Processing Mode custom attribute based on the setting of the EDITPA <code>processingMode</code> variable. For more information, see “processingMode EDITPA Variable” on page 122.</p>
6	<p>If the value of the EDI Processing Mode custom attribute is <code>Production</code>, send the internal-format document to the back-end system. To send the document, add your own logic or invoke a service that you create to send the document.</p>

Flow operation	Description
7	If the value of the EDI Processing Mode custom attribute is <i>not</i> Production (e.g., if it is Testing), invoke the <code>wm.pub.flow:debugLog</code> service to log a message to the server log. For example, the message might be “Testing document was received but not sent to back-end system.”

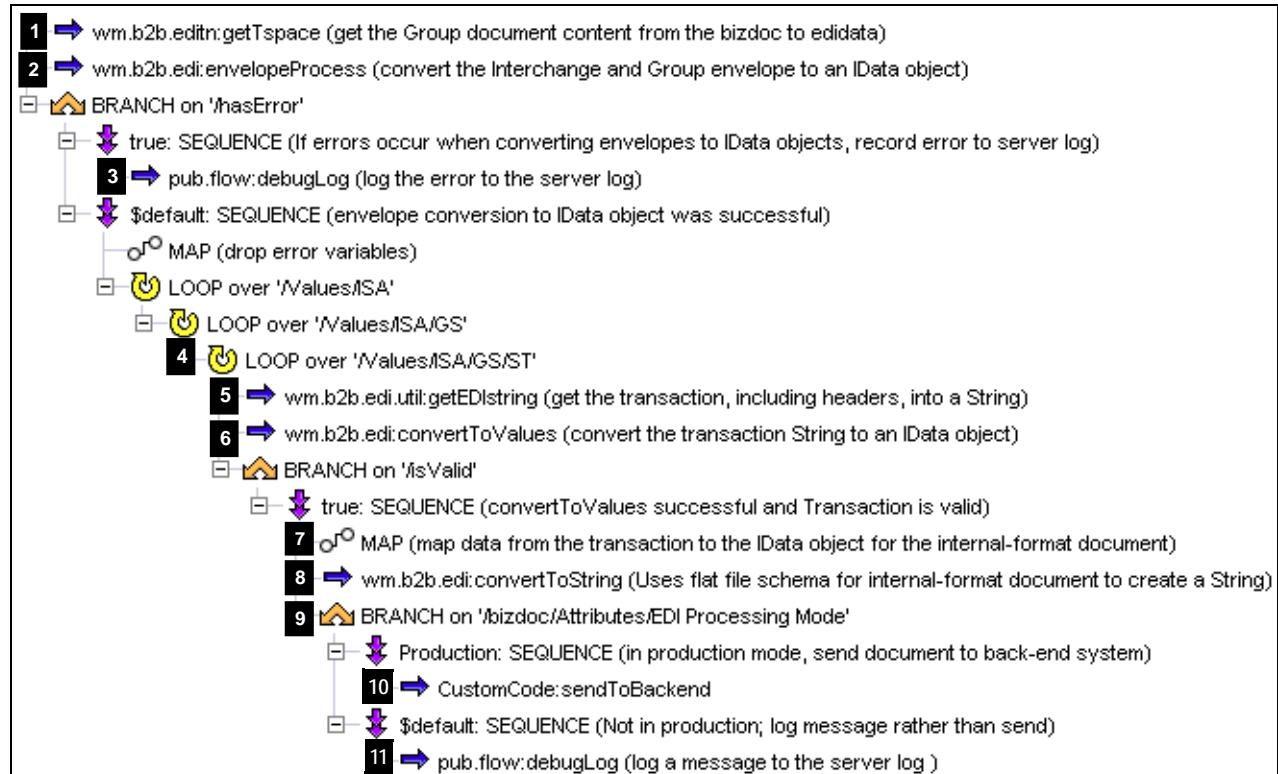
Logic to Process a Group Document when *splitOption* is Group

When the EDITPA variable *splitOption* is Group, the EDI Module does not create Transaction documents. If you want to process individual transactions, you can do so when you process the Group document. The following shows sample code that processes individual transactions within the group.



Note: If you want to generate functional acknowledgments (FAs), see “[Automatically Generating Functional Acknowledgements](#)” on page 258 in Chapter 15, “Optional Inbound Processing When Using Trading Networks”.

Sample code for processing a Group document to process transactions



Flow operation	Description
1	<p>Invoke the <code>wm.b2b.edith:getTspace</code> service to retrieve the content of the Group document from the <code>BizDocEnvelope</code> (in the <code>bizdoc</code> variable) and place it into the <code>edidata</code> variable.</p>
	<p>You can use the <code>getTspace</code> service whether the document is considered large or not. For more information about large document handling, see Chapter 7, “Handling Large Documents” and Chapter 19, “Handling Large Documents When Using Trading Networks”.</p>
	<p>For more information about the <code>getTspace</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
2	<p>Invoke the <code>wm.b2b.edi:envelopeProcess</code> service to process the envelopes in the Group document. This service converts the interchange, group, and transaction set headers into an <code>IData</code> object named <code>Values</code>. The contents of the transaction sets remain unparsed.</p>
	<ul style="list-style-type: none"> ■ For an ANSI X12 document, it creates an <code>IData</code> object of the ISA/IEA, GS/GE, and ST/SE headers/trailers.
	<ul style="list-style-type: none"> ■ For a UN/EDIFACT document, it creates an <code>IData</code> object of the UNB/UNZ, UNG/UNE, and UNH/UNT headers/trailers.
	<p>When setting the input variables to the <code>envelopeProcess</code> service, there is no need to validate or perform the compliance check on the Group document because the EDI Module performs this check when it initially processes the original EDI document that was sent to Trading Networks.</p>
	<p>For more information about the <code>envelopeProcess</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
3	<p>If the <code>wm.b2b.edi:envelopeProcess</code> service returns errors, invoke the <code>wm.pub.flow:debugLog</code> service to log a message to the server log.</p>
4	<p>Loop through the transaction set headers. Note that a Group document contains only a single interchange and a single group header. The data for the transaction set headers is within the <code>Values</code> <code>IData</code> object.</p>
	<ul style="list-style-type: none"> ■ For an ANSI X12 document, the transaction set headers are in <code>Values/ISA/GS/ST</code> as shown in the sample code
	<ul style="list-style-type: none"> ■ For a UN/EDIFACT document, the transaction set headers are in <code>Values/UNB/UNG/UNH</code>

Flow operation	Description
The remaining steps specify processing to perform for the content of each transaction.	
5	<p>Invoke the <code>wm.b2b.edi.util:getEDIstring</code> service to convert the transaction set header and trailer back to a String and concatenate them with the transaction set contents. The resulting transaction set with header and trailer can be either a String or <code>InputStream</code>. For more information about the <code>wm.b2b.edi.util:getEDIstring</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p> <p>You need to have an element that contains the entire transaction set, including header and trailer, before you can invoke the next service <code>wm.b2b.edi:convertToValues</code>. This is because the <code>convertToValues</code> service uses a flat file schema for the EDI transaction set that includes the transaction set header and trailer. If you input data without the header and trailer, the <code>convertToValues</code> service will return errors.</p>
6	<p>Invoke the <code>wm.b2b.edi:convertToValues</code> service to:</p> <ul style="list-style-type: none"> ■ Convert the content of the transaction set from either String or <code>InputStream</code> format to an <code>IData</code> object. ■ Validate the structure of the EDI transaction. <p>The inputs to the <code>convertToValues</code> service includes the flat file schema for the EDI transaction. The <code>convertToValues</code> service uses the flat file schema to both determine how to parse the transaction set into an <code>IData</code> object and to validate its structure.</p> <p>For more information about the <code>convertToValues</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
7	<p>Map the data from the EDI transaction set into the internal-format document.</p> <p>Now that the transaction set contents is an <code>IData</code> object, you can access the data in the transaction set to map it to an <code>IData</code> object for the internal-format document. Depending on the complexity of your mapping requirements, you might need to add more logic than a MAP flow operation, or create a separate service to perform the mapping. For more information about how to map, see Chapter 5, “Mapping Data to Form New Documents”.</p>

Flow operation	Description
8	<p>Invoke the <code>wm.b2b.edi:convertToString</code> service to convert the internal-format document from an <code>IData</code> object to String format.</p> <p>The inputs to the <code>convertToString</code> service include the <code>IData</code> object that contains the data for your internal-format document and the flat file schema for the internal-format document. The <code>convertToString</code> service uses the flat file schema to determine how to form the internal-format document. Alternatively, rather than use a flat file schema, the <code>convertToString</code> service also accepts an IS document type to define the structure of the internal-format document.</p> <p>For more information about the <code>convertToString</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
9	<p>Branch based on the value of the EDI Processing Mode custom attribute. This attribute is in the <code>bizdoc/Attributes/EDI Processing Mode</code> variable. The EDI Module sets the value of the EDI Processing Mode custom attribute based on the setting of the EDITPA <code>processingMode</code> variable. For more information, see “processingMode EDITPA Variable” on page 122</p>
10	<p>If the value of the EDI Processing Mode custom attribute is <code>Production</code>, send the internal-format document to the back-end system. To send the document, add your own logic or invoke a service that you create to send the document.</p>
11	<p>If the value of the EDI Processing Mode custom attribute is <i>not</i> <code>Production</code> (e.g., if it is <code>Testing</code>), invoke the <code>wm.pub.flow:debugLog</code> service to log a message to the server log. For example, the message might be “Testing document was received but not sent to back-end system.”</p>

Processing Inbound TRADACOMS Documents using Trading Networks

■ Overview	220
■ Before You Can Process Inbound EDI Documents	221
■ Specifying EDITPA Variables that Affect Inbound Processing	222
■ Trading Networks Attributes and EDI Documents	222
■ Defining Processing Rules to Process Inbound EDI Documents	227
■ Coding Services to Process File and Batch Documents	231

Overview

When webMethods Trading Networks (Trading Networks) receives an EDI document, it passes it to an EDI recognizer that performs the initial processing. The EDI recognizer is installed into Trading Networks when you install the webMethods EDI Module (EDI Module). You can tailor how the EDI recognizer processes an inbound EDI document. For more information, see [“Specifying EDITPA Variables that Affect Inbound Processing” on page 222](#).

To prepare the EDI document for the processing that you want to perform against it, the EDI recognizer splits the original inbound EDI document based on the *TRADACOMS/splitOption* variable that you specify in the EDITPA. The EDI recognizer can split the original EDI document into Transmission, Batch, and/or File documents. After forming the Transmission, Batch, and File documents, the EDI recognizer sends the documents into normal Trading Networks processing.

You define the processing that Trading Networks performs against the Transmission, Batch, and File documents by defining processing rules. Processing rules consist of:

- Criteria that Trading Networks uses to select the appropriate processing rule for a document
- Actions that Trading Networks is to take against the document

To set the criteria, you can use Trading Networks attributes along with other information that Trading Networks records about a document, for example, the TN document type used for the document. This allows you to set up criteria for processing rules, so Trading Networks selects the correct processing rule for your document. For example, you can set up criteria, so Trading Networks selects a processing rule only if it is from specific senders. For more information about the document attributes that are available for your use, see [“Trading Networks Attributes and EDI Documents” on page 222](#).

You will typically want to use the Execute a Service action in a processing rule to invoke a service that you create to process the Transmission, Batch, or File document. For more information about defining processing rules and creating a service to process a Transmission, Batch, or File document, see [“Defining Processing Rules to Process Inbound EDI Documents” on page 227](#).



Note: This chapter describes how to set up basic inbound processing when using Trading Networks. You can also set up to validate transmission and/or batch control numbers, as described in [Chapter 15, “Optional Inbound Processing When Using Trading Networks”](#).

To learn more about:

- Processing inbound EDI documents including an overview of the processing, see Chapter 3, "Using the EDI Module with Trading Networks", in the *webMethods EDI Module Concepts Guide*.
- Trading Networks document attributes and processing rules, see the *webMethods Trading Networks Concepts Guide* and the *webMethods Trading Networks User's Guide*.

Before You Can Process Inbound EDI Documents

Before you can set up inbound processing for EDI documents, do the following:

- Install the TN document types for the EDI documents that you want to process. For instructions on how to install TN document types for EDI documents, see "[Installing TN Document Types and Creating Flat File Schemas](#)" on page 104.
- Install the flat file schemas that define the structure of the EDI documents to process. The EDI Module uses the flat file schema for parsing, converting, and validating the structure of an inbound EDI document. The flat file schemas are installed when you install the TN document types for EDI documents.
- Optionally, create the flat file schema that defines the structure of an internal-format document. The service you create to process a Transmission, Batch, or File document might create an internal-format document (e.g., the format required by a back-end system). The EDI Module provides a service that you can invoke to create the internal-format document based on the flat file schema. Use the *webMethods Developer* to create the flat file schema. For more information, see the *Flat File Schema Developer's Guide*.
- Define profiles for the senders and receivers in the inbound EDI document. For instructions on how to create profiles, see "[Defining Trading Networks Profiles](#)" on page 147 and the chapter about profiles in the *webMethods Trading Networks User's Guide*.
- Define the default and optionally partner-specific EDITPAs for the sender/receiver pairs in the EDI document. For instructions on how to create the EDITPAs, see "[Defining EDI Trading Partner Agreements](#)" on page 150. For information about the variables in the EDITPAs that affect inbound processing, see "[Specifying EDITPA Variables that Affect Inbound Processing](#)" on page 222.

Specifying EDITPA Variables that Affect Inbound Processing

To tailor how you want the EDI Module to perform inbound processing, you use EDITPA variables. The EDI Module processes an inbound EDI document one transmission segment at a time.

For each transmission segment in an inbound EDI document, the EDI Module obtains the EDITPA values to use for transmission sender/receiver pair. The EDI Module uses those values when processing all documents (Transmission, Batch, and File) for the transmission segment.

For a complete list and description of the EDITPA variables, see [“wm.b2b.edtn.TPA:EDITPA IS Document Type” on page 152](#).

Trading Networks Attributes and EDI Documents

Trading Networks attributes identify selected content from a document that can be used for later processing. For example, you can set up processing rule criteria to select a processing rule based on the value of an attribute. Trading Networks supports two types of attributes: 1) system attributes that are defined by Trading Networks out-of-the-box and 2) custom attributes that are additional attributes that are added to Trading Networks. The EDI Module defines some custom attributes for EDI Module use. Additionally, you can define your own custom attributes and assign them values.

At run time when processing an EDI document, the EDI Module sets values for some of the system attributes and custom attributes using information from the EDI document being processed. If you want to define your own custom attributes, you must write code to set the values of those custom attributes.

You can use the attributes to:

- Process EDI documents using content-based processing rules; that is, select a processing rule for a document based on the value of the attributes extracted from the document
- Search for saved documents based on attribute values
- Report on documents based on attribute values

System Attributes that the EDI Module Sets

The EDI Module sets the **DocumentID** and **GroupID** system attributes using information from the EDI document. The EDI Module uses different values based on whether it is assigning the attribute to a Transmission, Batch, or File document. The sections below describe how the EDI Module sets each attribute for each type of document (Transmission, Batch, and File).

DocumentID

The **DocumentID** system attribute is an identifier of the document. The EDI Module sets the **DocumentID** as follows:

Type of Document	Value the EDI Module uses for DocumentID system attribute
Transmission	The sender's transmission reference (control number) specified in the STX segment.
Batch	The receiver's batch transmission reference (control number) specified in the BAT segment.
File	The file generation number in the message header's FIL segment.

GroupID

The EDI Module hierarchically assigns the **GroupID** system attribute, so that by viewing the **GroupID**, you can tell to which envelope an element originally belonged. The EDI Module sets the **GroupID** as follows:

Type of Document	Value the EDI Module uses for GroupID system attribute
Transmission	The sender's transmission reference (control number).
Batch	The sender's transmission reference/control number specified in the transmission's segment, so you can determine to which transmission the Batch document belongs.
File	The batch control number, so you can determine to which batch the File document belongs.

Custom Attributes that the EDI Module Sets

The EDI Module sets the value of the following custom attributes that you might want to use in processing rule criteria. These custom attributes are added to Trading Networks when you install the EDI Module. This table shows which attributes are stored for each document type (Transmission, Batch, and File).

Custom Attribute	Transmission Type	Batch Type	File Type
Application Reference	yes	derived from transmission	derived from transmission
Priority Code	yes	derived from transmission	derived from transmission
Detail Message Count	no	no	yes
Has VAT Message	no	no	yes
Has Reconciliation Message	yes	no	no
Is Multiple Envelope	yes	no	no
Version	yes (only valid value is 1)	derived from transmission	no (determined by document type)

You may also use these custom attributes:

Custom Attribute	Description
Envelope CntrlNum Status Group CntrlNum Status Transaction CntrlNum Status	<p>These attributes indicate the control number validation status of documents, thus enabling child documents to know the validation status of their parent documents.</p> <p>When validating document control numbers, the EDI Module sets this attribute to Valid, Duplicate, Out of Sequence, or Not Validated.</p> <ul style="list-style-type: none"> ■ For transmission documents, the EDI Module only sets the attribute Envelope CntrlNum Status. ■ For batch documents, the EDI Module sets the following attributes: <ul style="list-style-type: none"> ■ Envelope CntrlNum Status (indicates the status of the associated transmission document) ■ Group CntrlNum Status (indicates the status of the associated batch document)

Custom Attribute	Description
	<ul style="list-style-type: none"> ■ For file documents, the EDI Module sets the following attributes: <ul style="list-style-type: none"> ■ Envelope CntrlNum Status (indicates the status of the associated transmission document) ■ Group CntrlNum Status (indicates the status of the associated batch document; the EDI Module may also set this attribute to Not Present.) ■ Transaction CntrlNum Status (indicates the status of the associated transaction document)

Defining Your Own Custom Attributes for EDI Documents

You can create custom document attributes and assign them to TN document types for EDI documents. At run-time, services that you create must set the values of the attributes.

At Design Time Creating Attributes and Associating Them with TN Document Types

Perform the following procedure to associate custom attributes with EDI documents.

To associate a custom attribute with an EDI document

- 1 From the WmEDIforTN home page, install the TN document type for the type of EDI document with which you want to associate the custom attribute. For more information about how to complete this step, see “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104.
- 2 In the Trading Networks Console, create the custom attribute(s). For more information about how to complete this step, see the document attributes chapter in *webMethods Trading Networks User’s Guide*.
- 3 Invoke the `wm.b2b.editin:addAttributeTypeToBizDoc` service to associate the custom document attribute that you created in [step 2](#) with the TN document type you installed in [step 1](#). For more information about this service, see the *webMethods EDI Module Built-In Services Reference*.

Setting the Values of Attributes at Run Time

You must create a service to set the attribute values at run time. To have your service executed at run time, you can use it in the Execute a Service processing action of a Trading Networks processing rule. Note that your EDI document type or processing rule must use the Save Document to Database pre-processing action to save the document content and attributes to the database.

Create a service that performs the following logic:

- Obtain the values you want to use for each custom attribute.
- For each attribute, invoke the `wm.tn.doc:setAttribute` service to set the value of the attribute in the `BizDocEnvelope`.
- Invoke the `wm.tn.doc:updateAttributes` service to update the values of the attributes in the saved copy of the document that is in the Trading Networks database.

Example

When working with the File document type, you might want to associate the `PurchaseOrderNumber` attribute with an ORDERS document. To do so, you would:

- 1 Install the TN document type for the ORDERS document.
- 2 Create the `PurchaseOrderNumber` attribute.
- 3 Invoke the `wm.b2b.edtn:addAttributeTypeToBizDoc` service to associate the `PurchaseOrderNumber` attribute with the TN document type for the ORDERS document.
- 4 Create a processing rule that invokes a service that extracts the value for the `PurchaseOrderNumber` attribute from the ORDERS document. Your service should invoke the `wm.tn.doc:setAttribute` service to set the value for the attribute in the `BizDocEnvelope`, and then invoke the `wm.tn.doc:updateAttributes` service to update the attributes in the saved copy of the document in the Trading Networks database.

Defining a Processing Rule that Uses Your Custom Attribute as Criteria

To use the custom attributes you defined as criteria in a processing rule, you need to:

- Associate custom attributes with EDI document types (as described in [“At Design Time Creating Attributes and Associating Them with TN Document Types” on page 225](#) in this chapter).
- Create a processing rule to process the document. This processing rule does not use content-based processing. Leave the Extended Criteria tab of the Processing Rules screen blank. For this processing rule, use the Execute a Service processing rule to invoke a service that sets the values of the custom attributes (as described in [“Setting the Values of Attributes at Run Time” on page 225](#)). Additionally, this service should invoke the `wm.tn.reroute` service to have Trading Networks select another processing rule for the document. The next processing rule can use your custom attributes as criteria.
- Create a processing rule that uses specifies the custom attribute as criteria on the Extended Criteria tab.



Note: When you order your processing rules, be sure to list the processing rule that uses your custom attributes as criteria *before* the processing rule that sets the attribute values. In this way, Trading Networks will skip the processing rule that uses the custom attribute criteria the first time the document is processed, but select it the second time.

Defining Processing Rules to Process Inbound EDI Documents

To specify the processing that you want to perform against Transmission, Batch, and File documents that the EDI Module has split from the original inbound EDI document, you define processing rules.

When you define a processing rule, you define the criteria that you want Trading Networks to use to select the processing rule, and the pre-processing and processing actions that you want Trading Networks to perform against the document.

The processing rules you need to define are based on the *TRADACOMS/splitOption* that you specify in the EDITPA.

If you set <i>TRADACOMS/splitOption</i> to...	Then define processing rules for...
Transmission	<ul style="list-style-type: none">■ Transmission document
Batch	<ul style="list-style-type: none">■ Transmission document■ Batch documents
File	<ul style="list-style-type: none">■ Transmission document■ Batch documents■ Each type of File document (e.g., ORDHDR, INVHDR, etc.)

To define a processing rule

You define processing rules using the Trading Networks Console. For steps to define processing rules, see the chapter about processing rules in the *webMethods Trading Networks User's Guide*.

See the sections below for more information about setting the criteria, pre-processing actions, and processing actions for EDI documents.

Specifying Processing Rule Criteria

When you create a processing rule, you define criteria that Trading Networks uses to select the appropriate processing rule for a Transmission, Batch, or File document. The following lists the criteria you can specify.

Use this criteria	To have Trading Networks select a processing rule based on...	
Sender	The sender of the document.	
Receiver	The receiver of the document.	
Document Type	The TN document type for the document.	
	To have Trading Networks select a processing rule for... Set the Document Type criterion to this TN document type...	
	Transmission document	TRADACOMS Transmission
	Batch document	TRADACOMS Batch
	File document	A particular TRADACOMS File type and version, such as TRADACOMS 9 ORDHDR.
Recognition Errors	Whether Trading Networks or the EDI recognizer encountered any errors during the recognition process.	
Extended Criteria	The value of custom attributes. For example, on the Extended Criteria tab of the Processing Rules Detail screen, you might specify that the EDI Status attribute must have the value Duplicate Control Number for Trading Networks to use the processing rule. For more information about the custom attributes that the EDI Module sets, see “Custom Attributes that the EDI Module Sets” on page 224 . For information about how to set your own custom attributes, see “Defining Your Own Custom Attributes for EDI Documents” on page 225 .	



Note: The EDI Module does not set the User Status system attribute, so you should not use the User Status criterion.

Specifying Pre-Processing Actions

Default pre-processing actions are defined in TN document types. However, you can override the settings in the processing rule.

To learn more about how the pre-processing actions apply to EDI documents, see the section about pre-processing actions in Chapter 3, "Using the EDI Module with Trading Networks", in the *webMethods EDI Module Concepts Guide*.

The following table lists the pre-processing actions you can set in a processing rule and the default settings for each action as set in the TN document types for EDI documents.

Pre-processing action	Description	Default setting in TN document type for the EDI document
Validate Structure	Validates the structure of the EDI document.	Validate the structure of the envelope
Check for Duplicate Document	Determines if Trading Networks already has this document in its database.	Do not use Trading Networks check for duplication
Save Document to Database	Saves a copy of the document content, attributes, and/or activity log information to the Trading Networks database.	Save document content, attributes, and activity log

If a pre-processing action fails, Trading Networks records the error and continues processing. Trading Networks records the error in the *errors* variable in the *BizDocEnvelope*, which is in the *bizdoc* pipeline variable. Subsequent processing that you add, for example a service invoked by the Execute a Service processing action, can access the error information.

 Note: You cannot use the Verify Digital Signature pre-processing action because values for the SignedBody and Signature system attributes are not set for EDI documents.

Specifying Processing Actions

You can use all of the Trading Networks processing actions for EDI documents.

- If you want to process the information in the file of an inbound EDI document, for example to use information from File documents to form a document that is to go to a back-end system, you will mainly use the **Execute a Service** action. You create the service that the **Execute a Service** processing action invokes.

In this situation, you would want to set the EDITPA variable *TRADACOMS/splitOption* to **File** or **Batch**.

- For the File and/or Batch documents that result from the split, you would define processing rules that use the **Execute a Service** processing action to invoke a service that acts on the File and/or Batch documents. For more information about how to create the services, see [“Coding Services to Process File and Batch Documents”](#) below.
- For the Transmission document, you could define a processing rule that has no processing actions selected, which in effect causes Trading Networks to ignore the document. There is no need to process this document. Alternatively, you could set up your processing rules to allow the Transmission document to fall through to the **Default** rule, which only sets the User Status system attribute of the document to **IGNORED**. If you use the **Default** rule, you do not need to define a processing rule for the Transmission document.
- If you are sending the inbound EDI document through Trading Networks to simply deliver it to a destination without processing the file, you will mainly use the **Deliver Document By** action. In this situation, you set the EDITPA variable *TRADACOMS/splitOption* to **Transmission**. In this case, the only document sent to Trading Networks is the Transmission document. For more information about delivering documents, see [Chapter 16, “Forming EDI Documents to Send Outbound When Using Trading Networks”](#).

Coding Services to Process File and Batch Documents

The logic you create in the service that the Execute a Service action invokes depends on the types of documents that the EDI Module is splitting from the original EDI document (based on the EDITPA variable *TRADACOMS/splitOption*).

- If the *splitOption* is File, the EDI Module creates File, Batch, and Transmission documents from the original EDI document.
 - For the File document, create a service that processes the file contained in the File document. For example, form an internal-format document based on the file information and send the internal-format document to an internal application, e.g., a back-end system. For information about how to create this service, see “[Logic to Process a File Document](#)” on page 234.
 - For the Batch and Transmission documents, you do not need to create a service. All processing for these EDI documents would be complete in the processing of the File documents. You could set up processing rules for the Batch and Transmission documents that have no processing actions selected or set up your processing rules so the Transmission document falls through to the Default rule.
- If the *splitOption* is Batch, the EDI Module creates Batch and Transmission documents from the original EDI document.
 - For the Batch document, create a service that processes each file within the batch. For information about how to create this service, see “[Logic to Map a File Document To an Internal-Format Document](#)” on page 236.
 - For the Transmission document, you do not need to create a service. All processing for the EDI document would be complete in the processing of the Batch document. You could set up a processing rule for the Transmission document that selects no processing actions or set up your processing rules so the Transmission document falls through to the Default rule.
- eatmelf the *splitOption* is Transmission, the EDI Module creates Transmission documents from the original EDI document. You would set the *splitOption* to Transmission if you wanted to simply deliver the EDI document to its receiver. For more information about how to deliver a document, see “[Delivering the EDI Document](#)” on page 291.

The services you create can use information that is in the pipeline. For information about the data that is in the pipeline when your service is invoked, see “[Information in the Pipeline that Your Service Can Access](#)” below.

Information in the Pipeline that Your Service Can Access

When a Transmission, Batch, or File document is passed to the processing rule, the following information is in the pipeline and is accessible by your service:

- BizDocEnvelope in the *bizdoc* variable. Use the BizDocEnvelope to retrieve information that Trading Networks maintains about the document. The BizDocEnvelope adheres to the `wm.tn.rec:BizDocEnvelope` IS document type. It is also an instance of `com.wm.app.tn.doc.BizDocEnvelope`. For details about the information in the BizDocEnvelope, see the *webMethods Trading Networks Built-in Services Reference*. The following lists some of the variables within the BizDocEnvelope:
 - *DocumentID* contains the EDI transmission control number/file generation number from the transmission, batch, or file header as described in [“DocumentID” on page 223](#).
 - *Attributes* contains the custom attributes that were set for the document. For information about the custom attributes that the EDI Module sets, see [“Custom Attributes that the EDI Module Sets” on page 224](#).
 - *errors* contains the errors that Trading Networks and EDI Module encountered while processing the document.
- EDI document summary information in the *envelopeDocuments* variable. When Trading Networks originally receives the EDI document it passes it to the EDI recognizer, which parses the EDI document. This EDI document summary is the result of the parse. It is the entire original EDI document as an `IData` object.

You can use the EDI document summary information to derive a total document count or to determine when all of the documents in the original EDI document have completed processing. To do so, your service can perform a simple count of the original documents and a count of the documents processed. When these counts match, processing of the entire original EDI document is complete. You also can perform a query within Trading Networks that checks the status of each document using the Trading Networks internal ID for each document.

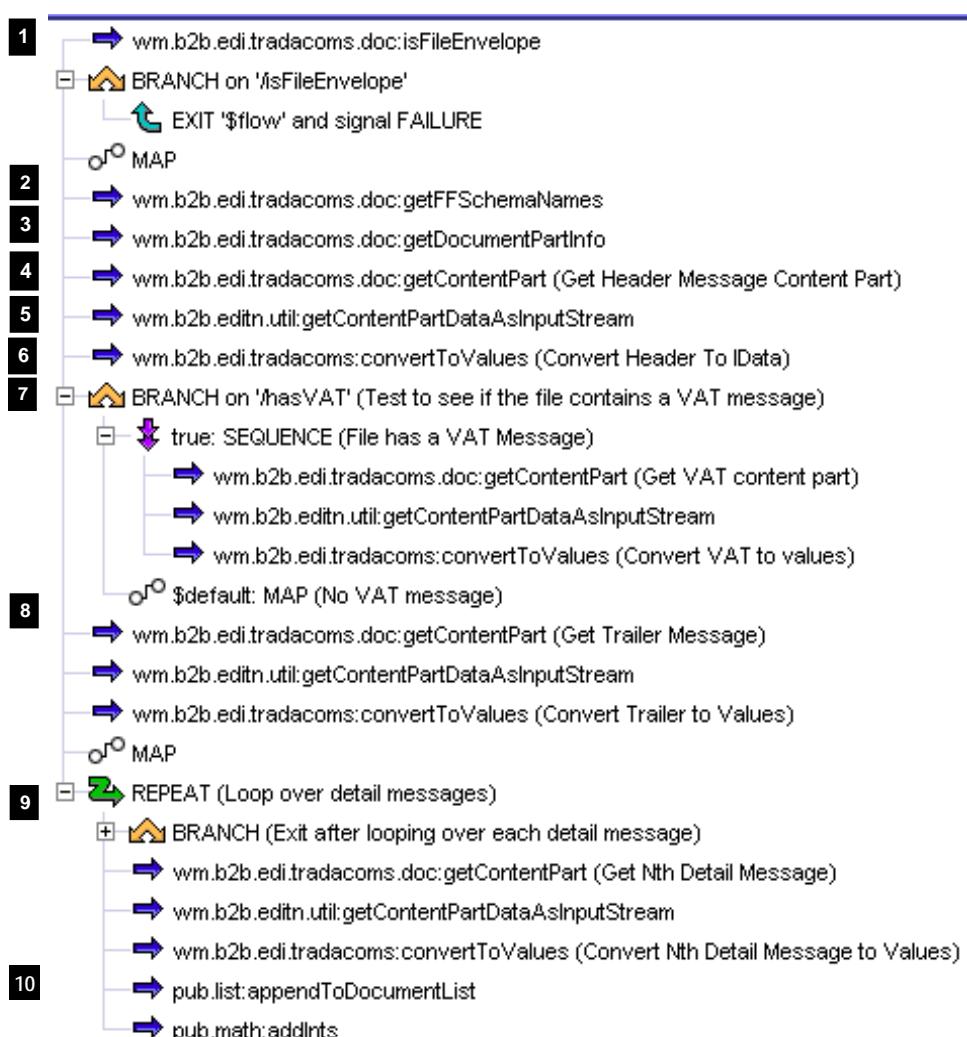
The following shows the structure of the EDI document summary information in the *envelopeDocuments* variable:

<u>Variable</u>	<u>Description</u>
<i>envelopeDocuments</i>	document list Summary of the transmission document including the batch and file that it contains.
<u>Variable in <i>envelopeDocuments</i></u>	<u>Description</u>
<i>docID</i>	String Internal ID that Trading Networks generated for the Transmission document.
<i>docTypeID</i>	String Internal ID of the TN document type used for the Transmission document.
<i>docTypeName</i>	String Name of the TN document type used for the Transmission document, e.g., TRADACOMS Transmission.
<i>groupDocuments</i>	document list (optional) Summary of all the batch documents in the transmission.
<u>Variable in <i>groupDocuments</i></u>	<u>Description</u>
<i>docID</i>	String Internal ID that Trading Networks generated for the Batch document.
<i>docTypeID</i>	String Internal ID of the TN document type used for the Batch document.
<i>docTypeName</i>	String Name of the TN document type used for the Batch document, e.g., Tradacoms Batch.
<i>transaction Documents</i>	document list (optional) Summary of all the file documents in the batch.
<u>Variable in <i>transactionDocuments</i></u>	<u>Description</u>
<i>docID</i>	String Internal ID that Trading Networks generated for the File document.
<i>docTypeID</i>	String Internal ID of the TN document type used for the File document.
<i>docTypeName</i>	String Name of the TN document type used for the File document, e.g., Tradacoms File.

Logic to Process a File Document

The following sample flow service can be invoked by a processing rule to process a File document. This service creates one File document type that contains all the detail messages contained in the file. Alternatively, you can write a service that creates a File document type for *each* detail message, as described in “[Storage Options For File Document Types](#)” on page 159.

Sample code for processing a File document



Flow operation	Description
1	Invoke the EDI built-in service <code>wm.b2b.edi.tradacoms.doc:isFileEnvelope</code> to determine whether a <code>BizDocEnvelope</code> contains a TRADACOMS File document.
2	Invoke the <code>wm.b2b.edi.tradacoms.doc:getFFSchemaNames</code> service to return the name of the flat file schemas that can be used to parse the parts of the TRADACOMS file. This service returns up to four flat file schemas: a header schema, a detail schema, a VAT schema (if present), and a trailer schema.
3	Invoke the <code>wm.b2b.edi.tradacoms.doc:getDocumentPartInfo</code> service to return the number of detail messages contained in the TRADACOMS file and to determine whether the file contains VAT information.
4	Invoke the <code>wm.b2b.edi.tradacoms.doc:getDocumentPart</code> service to return the content part object representing the header message.
5	Invoke the <code>wm.b2b.edi.util:getContentPartDataAsInputStream</code> service to obtain the data, as an input stream, from the content part. This service determines whether the document is stored in memory or on disk.
6	Invoke the <code>wm.b2b.edi.tradacoms:convertToValues</code> service to convert the content part input stream to an IS document (<code>IData</code> object) based on the input flat file schemas.
7	Determine whether the file contains a VAT message. If it contains a VAT message, parse the VAT message by invoking the <code>getContentPartInfo</code> , <code>getDocumentPartDataAsInputStream</code> , and <code>convertToValues</code> services, as you did for the header message.
8	Parse the trailer message by invoking the <code>getContentPart</code> , <code>getDocumentPartDataAsInputStream</code> , and <code>convertToValues</code> services, as you did for the header message.
9	Parse the detail messages by executing a loop and invoking the <code>getContentPart</code> , <code>getDocumentPartDataAsInputStream</code> , and <code>convertToValues</code> services for each detail message, as you did for the header message. In addition, invoke the <code>wm.b2b.edi.util.documentList:addDocToDocumentList</code> service to put all detail messages into a document list.
10	Append the documents to a document list by invoking the <code>pub.list:appendToDocumentList</code> service.

For more information about these built-in services, see the *webMethods EDI Module Built-In Services Reference*.

Logic to Map a File Document To an Internal-Format Document

The following sample code shows logic you might want to include to process a File document that contains a file from an inbound EDI document. The processing in the service below shows how to map information from the file to an internal-format document and send the document to an internal application, i.e., a back-end system.

Sample code for mapping a File document to an internal-format document

<table border="1" style="border-collapse: collapse; width: 100%;"> <tr><td style="width: 20px; height: 20px; background-color: black; color: white; text-align: center;">1</td><td>→ wm.b2b.edi.tradacoms.compose:startTradacomsTransmission</td></tr> <tr><td style="width: 20px; height: 20px; background-color: black; color: white; text-align: center;">2</td><td>→ WM (Map Header Message)</td></tr> <tr><td style="width: 20px; height: 20px; background-color: black; color: white; text-align: center;">3</td><td>→ WM (Map Detail Message)</td></tr> <tr><td style="width: 20px; height: 20px; background-color: black; color: white; text-align: center;">4</td><td>→ WM (Map Trailer Message)</td></tr> <tr><td style="width: 20px; height: 20px; background-color: black; color: white; text-align: center;">5</td><td>→ WM (Map Header Message)</td></tr> <tr><td style="width: 20px; height: 20px; background-color: black; color: white; text-align: center;">6</td><td>→ pub.io:streamToBytes</td></tr> <tr><td style="width: 20px; height: 20px; background-color: black; color: white; text-align: center;">7</td><td>→ pub.string:bytesToString</td></tr> </table>	1	→ wm.b2b.edi.tradacoms.compose:startTradacomsTransmission	2	→ WM (Map Header Message)	3	→ WM (Map Detail Message)	4	→ WM (Map Trailer Message)	5	→ WM (Map Header Message)	6	→ pub.io:streamToBytes	7	→ pub.string:bytesToString	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <th style="background-color: #e0e0e0;">Flow operation</th> <th>Description</th> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">1</td> <td>Invoke the <code>wm.b2b.edi.tradacoms.compose:startTradacomsTransmission</code> EDI built-in service to create an STX segment for a transmission. This service returns a TRADACOMS transmission object.</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">2</td> <td>Invoke the <code>wm.b2b.edi.tradacoms.compose:addToTradacomsTransmission</code> service to add the header message to the TRADACOMS transmission object that the <code>startTradacomsTransmission</code> service returned.</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">3</td> <td>Invoke the <code>addToTradacomsTransmission</code> service again to add the detail messages to the TRADACOMS transmission object.</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">4</td> <td>Invoke the <code>addToTradacomsTransmission</code> service again to add the trailer message to the TRADACOMS transmission object.</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">5</td> <td>Invoke the <code>wm.b2b.edi.tradacoms.compose:endTradacomsTransmission</code> service to create an END segment for the transmission.</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">6</td> <td>Invoke the Integration Server built-in service <code>pub.io:streamToBytes</code> service to convert the <code>InputStream</code> to bytes.</td> </tr> <tr> <td style="vertical-align: top; padding-right: 10px;">7</td> <td>Invoke the Integration Server built-in service <code>pub.string:bytesToString</code> service to convert the output of <code>pub.io:streamToBytes</code> to a string.</td> </tr> </table>	Flow operation	Description	1	Invoke the <code>wm.b2b.edi.tradacoms.compose:startTradacomsTransmission</code> EDI built-in service to create an STX segment for a transmission. This service returns a TRADACOMS transmission object.	2	Invoke the <code>wm.b2b.edi.tradacoms.compose:addToTradacomsTransmission</code> service to add the header message to the TRADACOMS transmission object that the <code>startTradacomsTransmission</code> service returned.	3	Invoke the <code>addToTradacomsTransmission</code> service again to add the detail messages to the TRADACOMS transmission object.	4	Invoke the <code>addToTradacomsTransmission</code> service again to add the trailer message to the TRADACOMS transmission object.	5	Invoke the <code>wm.b2b.edi.tradacoms.compose:endTradacomsTransmission</code> service to create an END segment for the transmission.	6	Invoke the Integration Server built-in service <code>pub.io:streamToBytes</code> service to convert the <code>InputStream</code> to bytes.	7	Invoke the Integration Server built-in service <code>pub.string:bytesToString</code> service to convert the output of <code>pub.io:streamToBytes</code> to a string.
1	→ wm.b2b.edi.tradacoms.compose:startTradacomsTransmission																														
2	→ WM (Map Header Message)																														
3	→ WM (Map Detail Message)																														
4	→ WM (Map Trailer Message)																														
5	→ WM (Map Header Message)																														
6	→ pub.io:streamToBytes																														
7	→ pub.string:bytesToString																														
Flow operation	Description																														
1	Invoke the <code>wm.b2b.edi.tradacoms.compose:startTradacomsTransmission</code> EDI built-in service to create an STX segment for a transmission. This service returns a TRADACOMS transmission object.																														
2	Invoke the <code>wm.b2b.edi.tradacoms.compose:addToTradacomsTransmission</code> service to add the header message to the TRADACOMS transmission object that the <code>startTradacomsTransmission</code> service returned.																														
3	Invoke the <code>addToTradacomsTransmission</code> service again to add the detail messages to the TRADACOMS transmission object.																														
4	Invoke the <code>addToTradacomsTransmission</code> service again to add the trailer message to the TRADACOMS transmission object.																														
5	Invoke the <code>wm.b2b.edi.tradacoms.compose:endTradacomsTransmission</code> service to create an END segment for the transmission.																														
6	Invoke the Integration Server built-in service <code>pub.io:streamToBytes</code> service to convert the <code>InputStream</code> to bytes.																														
7	Invoke the Integration Server built-in service <code>pub.string:bytesToString</code> service to convert the output of <code>pub.io:streamToBytes</code> to a string.																														

Optional Inbound Processing When Using Trading Networks

- Overview 238
- Performing Inbound Control Number Validation 238
- Automatically Generating Functional Acknowledgements 258

Overview

During inbound processing, you can have the webMethods EDI Module (EDI Module) perform the following optional processing:

- Validate inbound interchange and/or group control numbers (or TRADACOMS transmission and/or batch control numbers)
- Automatically generate functional acknowledgments (FAs)



Note: Functional acknowledgments (FAs) are not applicable to TRADACOMS.

Performing Inbound Control Number Validation

During inbound processing, the EDI Module can validate the interchange and/or group control numbers (or TRADACOMS transmission and/or batch control numbers) in the headers of the inbound EDI document. You can turn validation on or off. For information, see [“Turning Inbound Control Number Validation On and Off” on page 168](#).

The validation determines whether the control numbers are in order, and therefore whether the EDI documents arrived in order. The validation does *not* guarantee that EDI documents are processed in the same order in which documents are received.

To determine the order, the EDI Module determines whether a control number is valid or invalid by comparing the control number from the interchange or group header (or TRADACOMS transmission or batch header) to the next expected control number in the EDIControlNumber table. The EDI Module determines that the control number from the header is:

- Valid if the control number matches the next expected control number.
- Invalid if the control number does not match the next expected control number. When the control number is invalid, the EDI Module determines whether the invalid control number is:
 - Duplicate control number, which might indicate a duplicate document. A duplicate control number is one that the EDI Module believes has already been used, indicating a duplicate document.

-OR-

- Out-of-sequence control number, which might indicate that there was one or more preceding documents that should have already arrived, but have not. A control number is also considered out-of-sequence if the control number is not numerical.

For more information about how the EDI Module distinguishes between duplicate and out-of-sequence control numbers, see “[Using the Settings to Determine the Type of Invalid Control Number](#)” on page 175.



Note: Because the EDI Module only maintains the next expected control number (not a list of all previously used control numbers) and because the next expected control number can be manually set, the EDI Module might determine that a control number is a:

- Duplicate even though the control number was never before received.
- Out-of-sequence even though the control number has already been received.

You define the action that you want the EDI Module to take if a document with a duplicate or out-of-sequence control number arrives. For example, you can set up the EDI Module to prevent processing of documents with duplicate or out-of-sequence control numbers. For more information see “[Actions the EDI Module Can Take for Invalid Control Numbers](#)” on page 240. At a later time, you can force the processing of these documents. For more information, see “[Reprocessing EDI Documents with Invalid Control Numbers](#)” on page 250.

How the EDI Module Validates Inbound Control Numbers

The EDI Module uses the EDIControlNumber table to keep track of control numbers for each sender/receiver pair, EDI standard/version, production mode, and type (e.g., Envelope or Group type).

- The Envelope Type means you want to use the control number in the interchange header (ANSI X12 or UN/EDIFACT) or the transmission header (TRADACOMS).
- The Group Type means you want to use the control number in the group header (ANSI X12 or UN/EDIFACT) or the batch header (TRADACOMS).

The entry in the EDIControlNumber table contains the control number settings that are described in “[Control Number Cap, Minimum, Increment, and Window](#)” on page 172.

When the EDI Module receives an EDI document, it performs the following to validate control numbers:

- 1 The EDI Module first determines whether it is to validate the control number. To determine whether to validate control numbers in inbound EDI documents, the EDI Module uses the sender/receiver from the header and finds the settings you have defined for that sender/receiver pair. For information about how to allow validation, see “[Turning Inbound Control Number Validation On and Off](#)” on page 168.
If the EDI Module is to validate the control number, continue with the next step.
- 2 The EDI Module locates the entry in the EDIControlNumber table for the sender/receiver, EDI standard/version, production mode, and type (e.g., Envelope or

Group Type) identified in the interchange or group header (or the transmission or batch header).

- 3 The EDI Module determines whether the control number is valid (that is, in order) by comparing the control number from the header with the control number from the EDIControlNumber table entry. The EDI Module maintains the next expected control number in the EDIControlNumber table.
 - **Valid control number.** The control number value from the header matches the next expected control number in the EDIControlNumber table entry.

Note: If there is no EDIControlNumber entry for the sender/receiver, EDI standard/version, production mode, and type (e.g., “Envelope” or group type) identified in the interchange or group header, the EDI Module:

- Assumes the control number is valid.
 - Adds an entry to the EDIControlNumber entry for the sender/receiver, EDI standard/version, production mode, and type. In the EDIControlNumber entry, the EDI Module sets the control number cap, minimum, increment, and window to their defaults and calculates the next expected control number.
-

For a valid control number, the EDI Module calculates the next expected control and re-saves this new next expected control number in the entry in the EDIControlNumber table. For more information, see [“Using the Settings to Determine the Next Expected Control Number”](#) on page 175.

- **Invalid control number.** The control number value from the header does *not* match the next expected control number in the EDIControlNumber table entry. When the control number is not valid, the EDI Module determines whether the control number is a duplicate or out-of-sequence. For more information, see [“Using the Settings to Determine the Type of Invalid Control Number”](#) on page 175.

For an invalid control number, the EDI Module takes the action you define. For more information, see [“Actions the EDI Module Can Take for Invalid Control Numbers”](#) below.

Actions the EDI Module Can Take for Invalid Control Numbers

The following lists the actions that you can define for duplicate or out-of-sequence control numbers are described below.

- **Error & Continue,** for more information see [“Error & Continue”](#) on page 241.
- **Process Normally,** for more information see [“Process Normally”](#) on page 241.
- **Reject,** for more information, see [“Reject”](#) on page 242.

For information about how to define the action you want the EDI Module to take, see [“Defining Actions for Invalid Control Numbers”](#) on page 170.

Error & Continue

When you select **Error & Continue** and the EDI Module encounters an invalid control number, it:

- 1 Logs the error to the Trading Networks activity log.
- 2 Adds the error to the BizDocEnvelope (in the *bizdoc/Errors* pipeline variable). Because the error is logged to the BizDocEnvelope, you can use the **Recognition Errors** criterion in a processing rule to select documents have errors.
- 3 Sets the next expected control number for the sender/receiver by incrementing the control number in the appropriate EDIControlNumber table entry. The sender/receiver pair is either:
 - For ANSI X12 or UN/EDIFACT: The sender/receiver identified on an interchange header for an interchange control number or a group header for a group control number.
 - For TRADACOMS: The sender/receiver identified on a transmission header for a transmission control number or the sender/receiver identified on a batch header for a batch control number.

For more information, see [“Using the Settings to Determine the Next Expected Control Number” on page 175](#).

- 4 Continues normal processing of the document.

Process Normally

When you select **Process Normally** and the EDI Module encounters an invalid control number, it:

- 1 Logs the warning to the Trading Networks activity log.
- 2 Sets the next expected control number for the sender/receiver by incrementing the control number in the appropriate EDIControlNumber table entry. The sender/receiver pair is either:
 - For ANSI X12 or UN/EDIFACT: The sender/receiver identified on an interchange header for an interchange control number or the sender/receiver identified on a group header for a group control number.
 - For TRADACOMS: The sender/receiver identified on a transmission header for a transmission control number or the sender/receiver identified on a batch header for a batch control number.

For more information, see [“Using the Settings to Determine the Next Expected Control Number” on page 175](#).

- 3 Continues normal processing of the document.

Reject

When you select Reject and the EDI Module encounters an invalid control number, it does *not* perform normal processing on the document. Rather, the EDI Module:

- 1 Does *not* split the Interchange or Group document according to the setting of the EDITPA variable *splitOption* (or the Transmission or Batch document according to the setting of the EDITPA variable *TRADACOMS/splitOption*).

For ANSI X12 or UN/EDIFACT:

- If an interchange control number is invalid and the *splitOption* variable is set to *Group*, the EDI Module does not create Group documents.
- If an interchange control number is invalid and the *splitOption* variable is set to *transaction*, the EDI Module does not create Group or Transaction documents.
- If a group control number is invalid and the *splitOption* variable is set to *transaction*, the EDI Module does not create Group or Transaction documents for the group that has the invalid control number.

For more information about the *splitOption* variable, see [“splitOption EDITPA Variable” on page 119](#).

For TRADACOMS:

- If a transmission control number is invalid and the *TRADACOMS/splitOption* variable is set to *Batch*, the EDI Module does not create Batch documents.
- If a transmission control number is invalid and the *TRADACOMS/splitOption* variable is set to *File*, the EDI Module does not create Batch or File documents.
- If a batch control number is invalid and the *TRADACOMS/splitOption* variable is set to *File*, the EDI Module does not create Batch or File documents for the batch that has the invalid control number.

For more information about the *TRADACOMS/splitOption* variable, see [“TRADACOMS/splitOption EDITPA Variable” on page 153](#).

- 2 Logs the error to the Trading Networks activity log.
- 3 Adds the error to the BizDocEnvelope (in the *bizdoc/Errors* pipeline variable). Because the error is logged to the BizDocEnvelope, you can use the Recognition Errors criterion in a processing rule to select documents have errors.
- 4 Sets the Trading Networks custom attribute EDI Status based on whether the action is for a duplicate control number or out-of-sequence control number:
 - For a duplicate control number, sets the EDI Status attribute to Duplicate Control Number.
 - For an out-of-sequence control number, sets the EDI Status attribute to Out of Sequence Control Number.



Note: If the control number is valid, the EDI Module sets the EDI Status attribute to Processed.

- 5 Continues to process the unsplit document by passing it to Trading Networks processing rules. You can define a processing rule that has criteria that matches documents with the Trading Networks custom attribute EDI Status set to Duplicate Control Number or Out of Sequence Control Number. The processing rule should handle the rejected document, for example, by sending an administrator an e-mail notification. For more information, see [“Defining Processing Rules to Handle Documents with Invalid Control Numbers” on page 247](#).

Additionally, you can later force the processing of documents that contained invalid control numbers. For more information, see [“Reprocessing EDI Documents with Invalid Control Numbers” on page 250](#).



Note: The EDI Module does *not* increment the control number in the EDIControlNumber table.

Setting Up to Perform Inbound Control Number Validation

You set up whether to perform inbound control number validation and options for how the EDI Module performs inbound control number validation for partner pairs in your system. You can set up the EDI Module so it validates interchange and/or group control numbers (or transmission and/or batch control numbers) for all partners, some partners, or no partners.

For ANSI X12 or UN/EDIFACT

Action	Description
Turning inbound control number validation on or off:	<p>To turn control number validation on or off, you define EDITPA variables for partner pairs. If you want to turn control number validation on or off for <i>all</i> partners, set the EDITPA variables appropriately in the default EDITPA.</p> <p>The following lists the EDITPA variables that you set to turn validation on or off.</p> <ul style="list-style-type: none"> ■ For interchange control numbers, use the <i>ControlNumberManagement/validateInboundEnvelopeControlNumbers</i> variable. For more information, see “ControlNumberManagement/validateInboundEnvelopeControlNumbers EDITPA Variable” on page 124. <p>Non-standard</p> <p>When you are using non-standard processing, EDI Module does not use this EDITPA variable. Instead, it uses the Validate inbound envelope control numbers setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 390 in Appendix A, “Non-Standard Processing”.</p> <ul style="list-style-type: none"> ■ For group control numbers, use the <i>ControlNumberManagement/validateInboundGroupControlNumbers</i> variable. For more information, see “ControlNumberManagement/validateInboundGroupControlNumbers EDITPA Variable” on page 125.
Configure control number settings for sender/receiver pairs:	Configure the control number cap, control number minimum, control number increment, and control number window for sender/receiver pairs. For more information about these settings, see “Control Number Cap, Minimum, Increment, and Window” on page 172 . For more instructions for defining these settings, see “Defining Control Number Settings” on page 177 .

Action	Description
Set the action you want the EDI Module to take when it encounters an invalid control number:	<p>You define the actions to take for invalid control numbers using EDITPA variables. The following lists the EDITPA variables that you set:</p> <ul style="list-style-type: none"> ■ For duplicate control numbers, use the <i>ControlNumberManagement/duplicateControlNumberAction</i> variable. For more information, see “ControlNumberManagement/duplicateControlNumberAction EDITPA Variable” on page 125.
Non-standard	<p>When you are using non-standard processing, this EDITPA is only used for duplicate group control numbers. To set the action for duplicate interchange control numbers, use the Duplicate control number action setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 390 in Appendix A, “Non-Standard Processing”.</p> <ul style="list-style-type: none"> ■ For out-of-sequence control numbers, use the <i>ControlNumberManagement/outOfSequenceControlNumberAction</i> variable. For more information, see “ControlNumberManagement/useReverseRouting EDITPA Variable” on page 127.
Non-standard	<p>When you are using non-standard processing, this EDITPA is only used for out-of-sequence group control numbers. To set the action for out-of-sequence interchange control numbers, use the Out of sequence control number action setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 390 in Appendix A, “Non-Standard Processing”.</p>

For TRADACOMS

Action	Description
Turning inbound control number validation on or off:	<p>To turn control number validation on or off, you define EDITPA variables for partner pairs. If you want to turn control number validation on or off for <i>all</i> partners, set the EDITPA variables appropriately in the default EDITPA.</p> <p>The following lists the EDITPA variables that you set to turn validation on or off.</p> <ul style="list-style-type: none">■ For transmission control numbers, use the <i>ControlNumberManagement/validateInboundTransmissionControlNumbers</i> variable. For more information, see “TRADACOMS/ControlNumberManagement/validateInboundTransmissionControlNumbers EDITPA Variable” on page 154.■ For batch control numbers, use the <i>ControlNumberManagement/validateInboundBatchControlNumbers</i> variable. For more information, see “TRADACOMS/ControlNumberManagement/validateInboundBatchControlNumbers EDITPA Variable” on page 154.■ For file control numbers, use the <i>ControlNumberManagement/validateInboundFileControlNumbers</i> variable. For more information, see “TRADACOMS/ControlNumberManagement/validateInboundFileControlNumbers EDITPA Variable” on page 155.
Configure control number settings for sender/receiver pairs:	Configure the control number cap, control number minimum, control number increment, and control number window for sender/receiver pairs. For more information about these settings, see “Control Number Cap, Minimum, Increment, and Window” on page 172 . For more instructions for defining these settings, see “Defining Control Number Settings” on page 177 .

Action	Description
Set the action you want the EDI Module to take when it encounters an invalid control number:	<p>You define the actions to take for invalid control numbers using EDITPA variables. The following lists the EDITPA variables that you set:</p> <ul style="list-style-type: none"> ■ For duplicate control numbers, use the <i>ControlNumberManagement/duplicateControlNumberAction</i> variable. For more information, see “TRADACOMS/ControlNumberManagement/duplicateControl Number Action EDITPA Variable” on page 155. ■ For out-of-sequence control numbers, use the <i>ControlNumberManagement/outOfSequenceControlNumberAction</i> variable. For more information, see “TRADACOMS/ControlNumberManagement/outOfSequence Control NumberAction EDITPA Variable” on page 156.

Defining Processing Rules to Handle Documents with Invalid Control Numbers

If you set the action to take for an invalid control number to **Reject**, you should define a processing rule to handle the EDI document with the invalid control numbers. For information about how to define the action, see “[Defining Actions for Invalid Control Numbers](#)” on page 170 in Chapter 11, “[Defining Control Number Information for Partners](#)”. For more information about the processing that the EDI Module takes when you set the action to **Reject**, see “[Reject](#)” on page 242.

The EDI Module uses a Trading Networks custom attribute, EDI Status, to indicate whether a document has an invalid control number. You can use this status in the processing rule criteria.

Custom Attribute Set for Control Number Validation

The EDI Module sets the value of the following custom attribute for inbound EDI documents. For information about other Trading Networks document attributes that the EDI Module uses, see:

- For ANSI X12 or UN/EDIFACT users: “[Trading Networks Attributes and EDI Documents](#)” on page 196 in Chapter 13, “[Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks](#)”
- For TRADACOMS users: “[Trading Networks Attributes and EDI Documents](#)” on page 222 in Chapter 14, “[Processing Inbound TRADACOMS Documents using Trading Networks](#)”

Custom Attribute	Description
EDI Status	<p>The EDI Module sets this attribute to indicate whether a control number for a document is valid. If the EDI Module determines that a document contains:</p> <ul style="list-style-type: none"> ■ Valid control number, it sets this attribute to Processed. ■ Duplicate control number, it sets this attribute to Duplicate Control Number. ■ Out-of-sequence control number, it sets this attribute to Out of Sequence Control Number. <p>For more information about duplicate or out-of-sequence control numbers, see "Using the Settings to Determine the Type of Invalid Control Number" on page 175.</p>

Defining the Processing Rule

When you define a processing rule, you define the criteria that you want Trading Networks to use to select the processing rule, the pre-processing actions, and the processing actions. For more information about what to specify for each, see the sections below.

To define a processing rule to handle documents with invalid control numbers

You define processing rules using the Trading Networks Console. For steps to define processing rules, see the chapter about processing rules in the *webMethods Trading Networks User's Guide*. See the table below for more information about creating this processing rule.

Processing Rule tab:	Description						
Extended Criteria	<p>To select EDI documents that the EDI Module rejected because it detected an invalid control number and the action to take was set to Reject, use the Extended Criteria of the Trading Networks processing rules. You can use the Extended Criteria to select documents based on the value of custom attributes. In this case, you select documents based on the value of the EDI Status custom attribute, as shown below.</p> <ul style="list-style-type: none"> ■ To select documents with duplicate control numbers, specify: <table border="1"> <thead> <tr> <th>Attribute</th> <th>Operator</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>EDI Status</td> <td>Equals</td> <td>Duplicate Control Number</td> </tr> </tbody> </table>	Attribute	Operator	Value	EDI Status	Equals	Duplicate Control Number
Attribute	Operator	Value					
EDI Status	Equals	Duplicate Control Number					

Processing Rule tab:	Description				
	Attribute	Operator	Value		
	EDI Status	Equals	Out of Sequence Control Number		
Important! When specifying the value for extended criteria, be sure to use the exact combination of upper and lower case letters.					
Pre-Processing	You should override the default pre-processing actions with the following:				
	Pre-processing Action	Set to...			
	Verify Digital Signature	Do not verify digital signature			
	Validate Structure	Do not validate structure			
	Check for Duplicate Document	Do not use Trading Networks check for duplication			
	Save Document to Database	Save			
Note: If you do not select to save the content, EDI Module will automatically save the content so that you can reprocess the document. For more information about reprocessing, see "Reprocessing EDI Documents with Invalid Control Numbers" on page 250 .					
Action	Define the actions that you want to take to handle the EDI document containing the invalid control number. For example, you might select:				
	<ul style="list-style-type: none"> ■ Alert e-mail to send an e-mail message to notify an administrator of the problem. ■ Execute a service to invoke a service that you create to perform processing. You might choose to write your own service to send an e-mail message or perform some other type of notification. 				

Reprocessing EDI Documents with Invalid Control Numbers

If you set the action to take for a document with a duplicate or out-of-sequence control number to Reject, you can later reprocess the documents that have invalid control numbers. For information about how to define the action, see “[Defining Actions for Invalid Control Numbers](#)” on page 170 in Chapter 11, “[Defining Control Number Information for Partners](#)”. For more information about how the EDI Module processes documents when the action is Reject, see “[Reject](#)” on page 242.

The way you reprocess documents with invalid control numbers is different based on whether the invalid control number is a duplicate control number or out-of-sequence control number.

Reprocessing Documents with Duplicate Control Numbers

For documents with duplicate control numbers, first ensure the document is not a duplicate and that you have not already processed it. If you determine that you still need to process a document, you can force the reprocessing.

To reprocess a document with a duplicate control number

- 1 To determine the EDI documents that were rejected because they contained duplicate control numbers, from the webMethods Developer, invoke the `wm.b2b.editn.util.reprocess.listUnprocessedDocuments` service. When you invoke this service, specify `Duplicate` for the input variable `type`. The service returns a list of Trading Networks internal IDs of documents that have duplicate control numbers.
- 2 For each Trading Networks internal ID returned by the `wm.b2b.editn.util.reprocess.listUnprocessedDocuments` service, copy the Trading Networks internal ID into the Trading Networks Console to view the content of the document:
 - a From the Trading Networks Console, select **View ▶ Transaction Analysis**.
 - b If the query panels are not open on the Transaction Analysis screen, open them by selecting **Transactions ▶ Show/Hide Query**.
 - c Click the **Basic Criteria** tab if it is not already selected.
 - d Paste a Trading Networks internal ID in the **Internal ID** field.
 - e Click **Transactions ▶ Run Query**.
 - f Double click the row that Trading Networks returns to view the details.
 - g Click the **Content** tab to view contents of the document that had a duplicate control number.

- 3 If you decided you want to reprocess a document, from the webMethods Developer, invoke the `wm.b2b.editn.util.reprocess.reprocessDocument` service. When you invoke the document, specify the following:

<u>For this input variable...</u>	<u>Specify</u>
<code>internalID</code>	Trading Networks internal ID of the document you want to reprocess.
<code>generateFA</code>	For ANSI X12 and UN/EDIFACT only. Whether you want to generate functional acknowledgments (FAs) for the document you are reprocessing. Specify one of the following:
<u>Value of <code>generateFA</code></u>	<u>Meaning</u>
<code>true</code>	Generate FAs. Specify <code>true</code> when reprocessing a document with a duplicate interchange control number.
<code>false</code>	Do not generate FAs. This is the default. Specify <code>false</code> when reprocessing a document with a duplicate group control number.
<code>updateControlNumber</code>	Whether you want to update the next expected control number in the EDIControlNumber table. Specify <code>false</code> .

Reprocessing Documents with Out-of-Sequence Control Numbers

If you set the action to take for an out-of-sequence control number to `Reject`, you can later reprocess the out-of-sequence documents.

Out-of-sequence control numbers in EDI document might indicate that there were missing documents. Later if the missing EDI documents arrive, you might want to force the processing of the out-of-sequence document. This time because the missing EDI documents have arrived, the control number will no longer be out-of-sequence and the EDI Module will be able to process it normally.

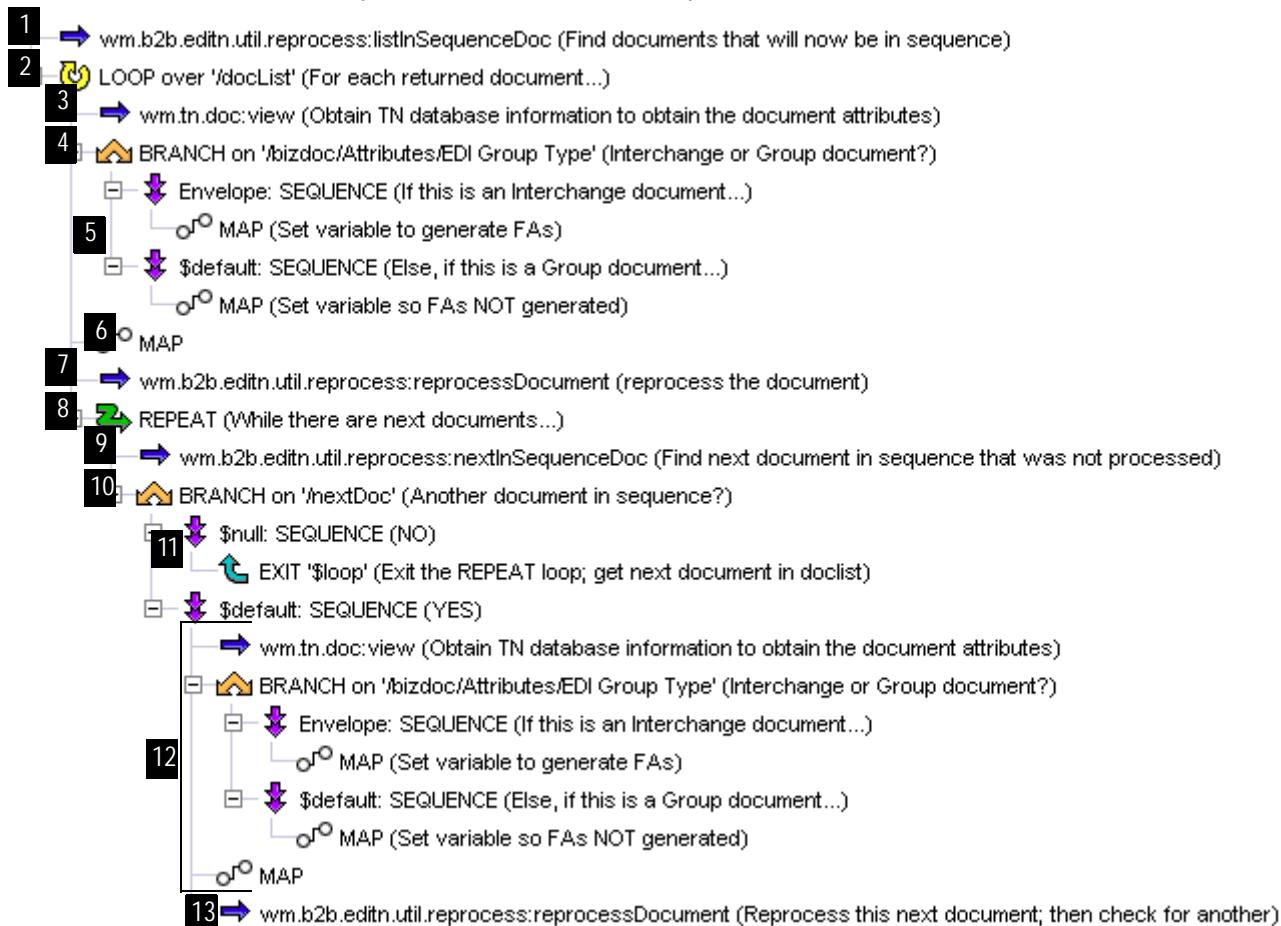
To force the processing of out-of-sequence documents, you can create a service that locates all out-of-sequence document and attempts to reprocess them. For information about how to create the service, see “[Creating a Service to Reprocess Documents with Out-of-Sequence Control Numbers](#)” on page 252.

Rather than invoke the service you create manually, you should use the Server Administrator to schedule a user task that causes the Integration Server to periodically invoke your service to reprocess any out-of-sequence documents. For more information about creating a user task to schedule a service to run periodically, see the chapter about managing services in the *webMethods Integration Server Administrator’s Guide*.

Creating a Service to Reprocess Documents with Out-of-Sequence Control Numbers

The following shows sample code that includes the logic you need to locate and reprocess out-of-sequence ANSI X12 documents. See the table below the diagram for more information.

Service to reprocess documents with out-of-sequence control numbers



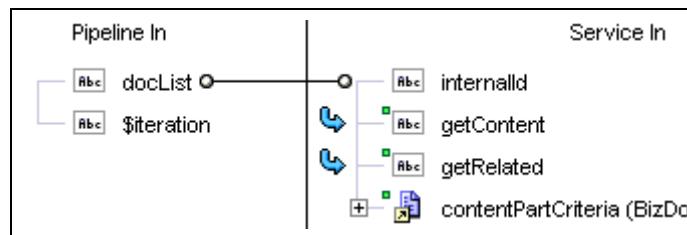
Flow operation	Description
1	<p>Invoke the <code>wm.b2b.edtn.util.reprocess:listInSequence</code> service to retrieve a list of EDI document that had out-of-sequence control numbers, but now due to missing documents arriving, are in sequence.</p> <p>For more information about the <code>listInSequence</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>

Flow operation	Description
2	The <code>wm.b2b.edtn.util.reprocess:listInSequence</code> service returns a list of Trading Networks internal IDs in the <code>docList</code> variable. This list represents EDI documents that are now in sequence. Loop to perform the following steps against each internal ID in the list.
3	Steps 3 through 6 are needed if you use automatic functional acknowledgment (FA) generation for ANSI X12 or UN/EDIFACT documents. If you use automatic FA generation, when reprocessing Interchange documents, you will want to generate FAs; however, when reprocessing Group documents, do not generate FAs because the FAs have already been generated for the document. For more about automatic FA generation, see "Automatically Generating Functional Acknowledgements" on page 258 .

Invoke the `wm.tn.doc:view` service to retrieve information about the document. Specifically, the value of the EDI Group Type attribute is needed to determine whether the document is an Interchange or Group document.

When you invoke `wm.tn.doc:view`, set the following input variables:

- Map the `docList` variable under Pipeline In to `internalID` under Service In.
- Set the value of `getContent` under Service In to `false`.
- Set the value of `getRelated` under Service In to `false`.

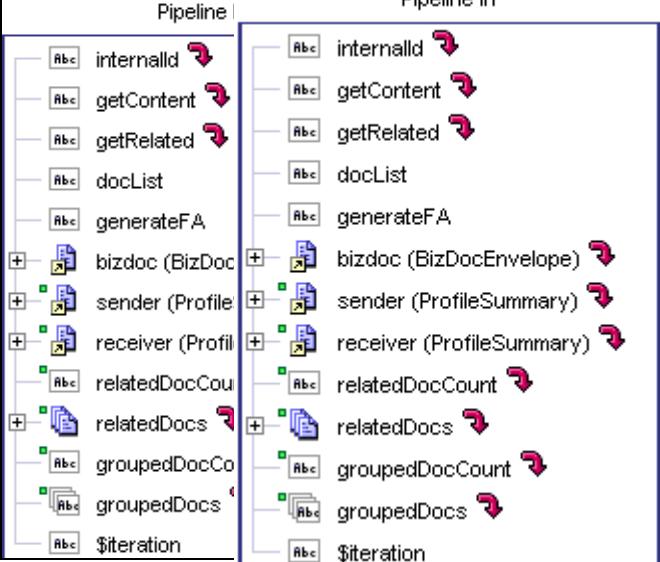


- 4 The value of the EDI Group Type attribute is in the `bizdoc/Attributes/EDI Group Type` pipeline variable. To branch based on its value, set the `Switch` property to the following:

`bizdoc/Attributes/EDI Group Type`

- When `bizdoc/Attributes/EDI Group Type` is `Envelope`, it is an Interchange document.
- When `bizdoc/Attributes/EDI Group Type` is another value (e.g., `PO`), it is a Group document.

Flow operation	Description
5	<p>Add a MAP flow operation to set the value of the <i>generateFA</i> variable, which is an input to the <code>wm.b2b.editn.util.reprocess:reprocessDocument</code> service.</p> <p>To do so, with the MAP flow operation selected on the Pipeline tab, add a <i>generateFA</i> variable under Pipeline Out and set the value to one of the following:</p> <ul style="list-style-type: none"> ■ In the SEQUENCE for when the EDI Group Type attribute is <code>Envelope</code>, set the value to <code>true</code>, which indicates the <code>wm.b2b.editn.util.reprocess:reprocessDocument</code> service should generate the FAs. ■ In the SEQUENCE for when the EDI Group Type attribute is another value, set the value to <code>false</code>, which indicates the <code>wm.b2b.editn.util.reprocess:reprocessDocument</code> service should not generate the FAs. <p>For example, below shows the pipeline for the MAP when the EDI Group Type attribute is <code>Envelope</code> (so setting the <i>generateFA</i> variable to <code>true</code>).</p>

Flow operation	Description
6	<p>Use a MAP flow operation to clean up the pipeline by dropping all variables related to invoking the <code>wm.ln.doc:view</code> service. Drop the following variables: <code>internalID</code>, <code>getContent</code>, <code>getRelated</code>, <code>bizdoc</code>, <code>sender</code>, <code>receiver</code>, <code>relatedDocCount</code>, <code>relatedDocs</code>, <code>groupDocCount</code>, and <code>groupedDocs</code>.</p> 
7	<p>Invoke the <code>wm.b2b.editn.util.reprocess:reprocessDocument</code> service to reprocess an EDI document that is now in sequence.</p> <p>To set the inputs for this service, with this flow operation selected, perform the following on the Pipeline tab:</p> <ul style="list-style-type: none"> ■ Map the <code>docList</code> variable under Pipeline In to <code>internalID</code> under Service In. ■ If you added the <code>generateFA</code> variable to the pipeline in the previous steps, the <code>generateFA</code> variable exists under Pipeline In and is automatically mapped to the input variable <code>generateFA</code> under Service In. If you are not using automatic FA generation, you can set the value of <code>generateFA</code> under Service In to <code>false</code>. ■ Set the value of the <code>updateControlNumber</code> variable under Service In to <code>true</code>. This is necessary so the service updates the next expected control number in the <code>EDIControlNumber</code> table. <p>For more information about the <code>reprocessDocument</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>

Flow operation	Description
8	<p>After you reprocess a document, there might be another out-of-sequence document that is now in sequence. This next document that is now in sequence would be one that has the same sender/receiver <i>and</i> group type as the one you just reprocessed. For example, if you reprocess a Group document of group type "PO" from sender A and receiver B, and the Group document had the control number 4. The Group document of group type "PO" from sender A and receiver B with control number 5 might also be available to process, and it is now in sequence.</p>
9	<p>To locate and reprocess the next document in sequence, use a REPEAT loop. You will exit this loop when a pipeline variable becomes null, indicating there are no more next documents in sequence. (See step 11 below.) When defining the properties for the REPEAT loop, do the following:</p> <ul style="list-style-type: none"> ■ Set the Repeat Interval property to -1. ■ Set the Repeat On property to SUCCESS. <p>Invoke the <code>wm.b2b.edtn.util.reprocess:nextInSequenceDoc</code> service to determine whether there is another document that had an out-of-sequence control number that is now in sequence due to the processing the document in step 2 above.</p> <p>The input to this service (<i>bizdoc</i>) is in the pipeline because it is an output of the <code>wm.b2b.edtn.util.reprocess:reprocessDocument</code> service. The Pipeline In variable will automatically map to the Service In variable. You should drop the <i>bizdoc</i> variable in Pipeline Out.</p> <pre> graph LR subgraph Pipeline_In [Pipeline In] direction TB P1[docList] --- > > S1(()) P2[bizdoc BizDocEnvelope] --- > > S1 P3[\$retries] --- > > S1 end subgraph Service_Box [nextInSequenceDoc] direction TB S1 --- > > Service_In[*] Service_Out[nextDoc] end </pre>
10	<p>If the <code>nextInSequenceDoc</code> service locates a next document, it returns the Trading Networks internal ID of the document in the <i>nextDoc</i> variable. Otherwise, the <i>nextDoc</i> variable will be null.</p> <p>For more information about the <code>nextInSequenceDoc</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
11	<p>Use a BRANCH flow operation to branch based on the value of the <i>nextDoc</i> variable, which is the result of the <code>nextInSequenceDoc</code> service.</p> <p>If the value of <i>nextDoc</i> variable is null, exit the REPEAT loop.</p>

Flow operation	Description
12	<p>If the <i>nextDoc</i> variable has a value, prepare to execute the <code>wm.b2b.editn.util.reprocess:reprocessDocument</code> service against this next document to reprocess it.</p> <p>The following flow operations are similar to steps 3 through 6 described above. Differences are noted below.</p> <ul style="list-style-type: none"> ■ INVOKE <code>wm.tn.doc:view</code> When setting the inputs to this service, map the <i>nextDoc</i> variable under Pipeline In to <i>internalID</i> under Service In. ■ BRANCH on <i>bizdoc/Attributes/EDI Group Type</i> ■ SEQUENCE for setting <i>generateFA</i> when EDI Group Type is Envelope ■ SEQUENCE for setting <i>generateFA</i> when EDI Group Type has another value ■ MAP to drop variables related to invoking <code>wm.tn.doc:view</code>
13	<p>Invoke the <code>wm.b2b.editn.util.reprocess:reprocessDocument</code> service against the next document to reprocess it.</p> <p>To set the inputs for this service, with this flow operation selected, perform the following on the Pipeline tab:</p> <ul style="list-style-type: none"> ■ Map the <i>nextDoc</i> variable under Pipeline In to <i>internalID</i> under Service In. You should drop the <i>nextDoc</i> variable in Pipeline In. ■ If you added the <i>generateFA</i> variable to the pipeline in the previous steps, the <i>generateFA</i> variable exists under Pipeline In and is automatically mapped to the input variable <i>generateFA</i> under Service In. If you are not using automatic FA generation, you can set the value of <i>generateFA</i> under Service In to false. ■ Set the value of the <i>updateControlNumber</i> variable under Service In to true. This is necessary so the service updates the next expected control number in the EDIControlNumber table.

Automatically Generating Functional Acknowledgements

During inbound processing, the EDI Module can automatically generate functional acknowledgments (FAs) for the inbound EDI document.



Note: Functional acknowledgments (FAs) are not applicable to TRADACOMS.

The following table shows the type of FA that the EDI Module generates based on the EDI standard of the inbound document.

EDI Standard	Type of FA the EDI Module Generates	Description
ANSI X12 UCS VICS	ANSI X12 997	The EDI Module generates an FA for each group in the inbound document.
UN/EDIFACT EANCOM ODETTE	UN/EDIFACT CONTRL	The EDI Module generates an FA for each interchange in the inbound document.

After generating the FAs, the EDI Module sends the FAs to Trading Networks, so they can be delivered to the sender of the inbound document. You need to define a processing rule to deliver the FAs.

To learn more about how to generate a report for FA reconciliation, see [Chapter 21, “Reconciling Functional Acknowledgments”](#).

Setting Up to Use Automatic FA Generation

The following table lists the actions you need to take to set up automatic FA generation and includes a cross reference to the section in this guide that provides more detail about how to perform the action.

Action	For more information see...
Ensure you have installed the TN document types and flat file schemas for the FA (997 or CONTRL).	“Before You Can Use Trading Networks to Process EDI Documents” on page 101
Turn on automatic FA generation.	“Turning Automatic FA Generation On or Off” on page 259
Define the level of detail to include in the generated FAs and how to generate the control numbers for the FA.	“Variables that Affect How the EDI Module Generates the FA” on page 260

Action	For more information see...
Define how the EDI Module is to set the FA status for an interchange, group, or transaction within an EDI document.	"How the EDI Module Reports the FA Status" on page 263
Define the action the EDI Module is to take for an Interchange, Group, or Transaction document based on its FA status.	"Actions the EDI Module Takes Based on FA Status" on page 272
Define one or more Trading Networks processing rules to process the generated FAs.	"Defining Processing Rules to Process FAs" on page 278
Define one or more Trading Networks processing rules to perform normal processing on documents that have FA statuses that you define as acceptable.	"Defining Processing Rules to Process Inbound EDI Documents" on page 206
For more information about acceptable FA statuses, see "Defining the FA Statuses that Are Acceptable and Unacceptable" on page 274.	
Define one or more Trading Networks processing rules to perform processing on documents that have FA statuses that you define as unacceptable.	"Defining Processing Rules for Documents With Unacceptable FA Statuses" on page 279
For more information about unacceptable FA statuses, see "Defining the FA Statuses that Are Acceptable and Unacceptable" on page 274.	

Turning Automatic FA Generation On or Off

You set the *FAGeneration/autoGenerateFA* EDITPA variable to indicate whether you want the EDI Module to automatically generate FAs for an interchange sender/receiver pair. Because you turn automatic FA generation on or off using an EDITPA variable, you can control whether the EDI Module automatically generates FAs for all sender/receiver pairs (by setting the variable in the default EDITPA) or for specific sender/receiver pairs (by using partner-specific EDITPAs).

To have the EDI Module:

Always automatically generate FAs

Set the *FAGeneration/autoGenerateFA* EDITPA variable to:

On

To have the EDI Module:	Set the <i>FAGeneration/autoGenerateFA</i> EDITPA variable to:
Never automatically generate FAs	Off
Automatically generate FAs based on the indicator flag in the interchange header (ISA14 or UNB09)	Per Document

Non-standard

When you are using non-standard processing, EDI Module does not use this EDITPA variable. Instead, it uses the Auto Generate FA setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in Appendix A, “Non-Standard Processing”.

Variables that Affect How the EDI Module Generates the FA

When generating FAs, the EDI Module uses the *FAGeneration/FALevel* and *FAGeneration/generateControlNumber* EDITPA variables:

- *FAGeneration/FALevel* defines the level of detail that the EDI Module is to acknowledge in the FAs that it generates.

To have the EDI Module acknowledge at:	Set the <i>FAGeneration/FALevel</i> EDITPA variable to:
Envelope level (group for ANSI X12 and interchange for UN/EDIFACT)	Default
Transaction set level	TransactionSet
Segment level	Segment
Element level	Element

Note: If you are generating FAs at the element level, be sure to configure the maximum number of errors to report per FA transaction. For more information, see “[Configuring the Maximum Number of Transaction Errors](#)” on page 58.

Non-standard

When you are using non-standard processing, EDI Module does not use this EDITPA variable. Instead, it uses the FA Level setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in Appendix A, “Non-Standard Processing”.

- *FAGeneration/generateControlNumber* to defines how the EDI Module is to generate the control numbers that it uses in the interchange and group headers of the generated FA.

To have the EDI Module:	Set the <i>FAGeneration/generateControlNumber</i> EDITPA variable to:
Use the control numbers from the inbound document.	FromInboundDocument
The EDI Module use the control number from the corresponding header of the inbound EDI document that the FA acknowledges. For example, if acknowledging a group, the EDI Module uses the control number from the corresponding group header in the inbound document.	
Randomly generate control numbers for the interchange and group headers of the FA.	Random
Look up the control number to use from the EDIControlNumber table.	FromControlNumberTable
The EDI Module searches the EDIControlNumber table for the entry that matches the sender/receiver, EDI standard/version, production mode, and type (e.g., "Envelope" or group type) identified in the corresponding interchange or group header of the inbound EDI document that the FA acknowledges. The EDI Module uses the next control number from this EDIControlNumber table entry for the FA. It also increments the value of the next control number in the table entry, so it reflects the new next control number.	



Note: The EDI Module always sets the control number for the transaction (997 or CONTRL) to 001.

Non-standard

When you are using non-standard processing, EDI Module does not use this EDITPA variable. Instead, it uses the Generate Control Number setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see ["Defining Interchange-Level Sender/Receiver Pair Information" on page 390](#) in Appendix A, "Non-Standard Processing".

Contents of the Generated Functional Acknowledgment

This section provides a table that summarizes the information in the generated FA. It describes items that the EDI Module sets in the:

- **BizDocEnvelope** that the EDI Module creates for the FA. When defining a Trading Networks processing rule to deliver the FA, you use the BizDocEnvelope information when defining the processing rule criteria.
- **Interchange header** of the FA
- **Group header** of the FA
- **Transaction header** (997 for ANSI X12 or CONTRL for UN/EDIFACT)

Values in the Generated FA		
Item	Setting	Value
BizDocEnvelope	sender	The receiver from the inbound document's interchange header
	receiver	The sender from the inbound document's interchange header
	TN document type	<ul style="list-style-type: none"> ■ For ANSI X12: X12 Envelope ■ For UN/EDIFACT: EDIFACT Envelope
Interchange header	EDI FA Outbound custom attribute	true
	sender	The receiver from the inbound document's interchange header
	receiver	The sender from the inbound document's interchange header
Group header	control number	The value is based on the <i>FAGeneration/generateControlNumber</i> EDITPA variable
	sender	The receiver from the inbound document's group header
	receiver	The sender from the inbound document's group header
Transaction header	control number	The value is based on the <i>FAGeneration/generateControlNumber</i> EDITPA variable
	control number	001
	control number	

How the EDI Module Reports the FA Status

The EDI Module reports an FA status for each transaction, group, and UN/EDIFACT interchange in an inbound EDI document. The FA status can be one of the following:

ANSI X12 997	UN/EDIFACT CONTRL	Description
N	N	Not Allowed
R	R	Rejected
P	R	Partially Accepted (for groups only)
E	R	Accepted, But Errors Were Noted
A	A	Accepted
FA	FA	EDI document is an FA (997 or CONTRL)

 Note: The rest of this chapter uses the FA statuses reported in an ANSI X12 997. Refer to this table to see the related FA status that is reported in a UN/EDIFACT CONTRL.

The following table lists the information that the EDI Module uses to determine the FA status for an interchange, group, and transaction in an inbound document.

Section	Information the EDI Module uses to determine the FA status
Transaction	<ul style="list-style-type: none"> ■ Whether the transaction is allowed in the group (or interchange) in which it resides ■ Whether the transaction is an FA (997 or CONTRL) ■ Syntax error status ■ Logical error status
Group (The EDI Module will report FA status for UN/EDIFACT groups if a UN/EDIFACT document contains groups.)	<ul style="list-style-type: none"> ■ Whether the group is “FA” or “CONTRL” ■ Syntax error status ■ Logical error status ■ Child transaction rejected status
Interchange (UN/EDIFACT only)	<ul style="list-style-type: none"> ■ Syntax error status ■ Logical error status ■ Child transaction rejected status

Child Transaction that is Not Allowed in its Group

When setting the FA status for a child transaction, the first check that the EDI Module makes is to determine whether the child transaction is allowed in its group. For example, it is not valid for an ANSI X12 997 transaction to be in an ANSI X12 PO group. Similarly, it is not valid for a UN/EDIFACT CONTRL to be in an ORDERS group.

If the child element is:

- **Allowed**, the EDI Module determines the rest of the statuses (e.g., syntax error status, logical error status, child transaction rejected status) for the child element.
- **Not allowed**, the EDI Module sets the FA status to “Not Allowed” and does not determine the rest of the statuses.

Transaction or Group is for an FA

The EDI Module reports the FA status “FA” when a transaction *is* an FA or a group that contains an FA. In other words, in the following circumstances:

- A transaction is an FA, that is either an ANSI X12 997 or UN/EDIFACT CONTRL
- A group is either an ANSI X12 “FA” or UN/EDIFACT “CONTRL” group.

Syntax Error Status

The syntax error status indicates whether there are syntax errors in the transaction, group, or UN/EDIFACT interchange, for example, missing mandatory elements, violation of syntax rules, invalid field lengths, code list violations, or segment repeat counts exceeded.

The EDI Module uses the syntax error status along with the logical error status and child transaction rejected status to determine the FA status. For more information, see [“How the EDI Module Determines Which FA Status to Use” on page 268](#).

You use the *FAGeneration/rejectionRules/syntaxErrorStatus* EDITPA variable to indicate how you want the EDI Module to report the syntax error status. The following table describes the settings of the *FAGeneration/rejectionRules/syntaxErrorStatus* EDITPA variable and their meanings:

Set <i>syntaxErrorStatus</i> to:	To have the EDI Module:
Rejected	<p>Report the syntax error status:</p> <ul style="list-style-type: none">■ “Accepted” if there are no syntax errors■ “Rejected” if there are syntax errors <p>Use this setting if you want to reject the element (e.g., transaction) because of the syntax errors.</p>

Set <i>syntaxErrorStatus</i> to:	To have the EDI Module:
Accepted, But Errors Were Noted	<p>Report the syntax error status:</p> <ul style="list-style-type: none"> ■ “Accepted” if there are no syntax errors ■ “Accepted, But Errors Were Noted” if there are syntax errors <p>Use this setting if you want to know whether there were syntax errors, but do not want to reject the element (e.g., transaction) because of the syntax errors.</p>
Accepted	<p>Always reports the syntax error status as “Accepted” regardless of whether there are any syntax errors.</p> <p>Use this setting if you do not want to check for syntax errors.</p>

Non-standard

When you are using non-standard processing, EDI Module does not use the *FAGeneration/rejectionRules/syntaxErrorStatus* EDITPA variable. Instead, it uses the **Syntax Error Status** setting that you set from the **Interchange Information Detail** screen of the EDI Module home page. For more information about accessing and using this screen, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 390](#) in Appendix A, “Non-Standard Processing”.

Logical Error Status

The logical error status indicates whether the transaction, group, or UN/EDIFACT interchange is malformed. For example:

- The control number in a header does not match the control number in the corresponding trailer.
- The segment count in a trailer does not have an accurate group, transaction, or segment count.

The EDI Module uses the logical error status along with the syntax error status and child transaction rejected status to determine the FA status. For more information, see “[How the EDI Module Determines Which FA Status to Use](#)” on page 268.

You use the *FAGeneration/rejectionRules/logicalErrorStatus* EDITPA variable to indicate how you want the EDI Module to report the logical error status. The following table describes the settings of the *FAGeneration/rejectionRules/logicalErrorStatus* EDITPA variable and their meanings:

Set <i>logicalErrorStatus</i> to:	To have the EDI Module:
Rejected	<p>Report the logical error status:</p> <ul style="list-style-type: none">■ “Accepted” if there are no logical errors■ “Rejected” if there are logical errors <p>Use this setting if you want to reject the element (e.g., transaction) because of the logical errors.</p>
Accepted, But Errors Were Noted	<p>Report the logical error status:</p> <ul style="list-style-type: none">■ “Accepted” if there are no logical errors■ “Accepted, But Errors Were Noted” if there are logical errors <p>Use this setting if you want to know whether there were logical errors, but do not want to reject the element (e.g., transaction) because of the logical errors.</p>
Accepted	<p>Always reports the logical error status as “Accepted” regardless of whether there are any logical errors.</p> <p>Use this setting if you do not want to check for logical errors.</p>

Non-standard

When you are using non-standard processing, EDI Module does not use the *FAGeneration/rejectionRules/logicalErrorStatus* EDITPA variable. Instead, it uses the Logical Error Status setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in Appendix A, “Non-Standard Processing”.

Child Transaction Rejected Status

The child transaction rejected status indicates whether child elements of a group or UN/EDIFACT interchange have an FA status of “Rejected”. That is:

- For a group, whether any transaction within a group has an FA status of “Rejected”
- For a UN/EDIFACT interchange, whether any transaction or group within the interchange has an FA status of “Rejected”

 Note: For information about how the EDI Module sets the FA status for a transaction and group based on the syntax error status and the logical error status, see [“Transaction FA Status” on page 268](#) and [“Group FA Status” on page 270](#).

The EDI Module uses the child transaction rejected status along with the syntax error status and logical error status to determine the FA status for the group and interchange. For more information, see [“How the EDI Module Determines Which FA Status to Use” on page 268](#).

You use the *FAGeneration/rejectionRules/childTransactionRejectedStatus* EDITPA variable to indicate how you want the EDI Module to report the child transaction rejected status. The following table describes how the EDI Module sets the child transaction rejected status based on the setting of the *FAGeneration/rejectionRules/childTransactionRejectedStatus* EDITPA variable and the FA statuses of the child transactions.

Setting of <i>childTransactionRejected Status</i>	When the FA statuses of the child transactions are:			
	All “Accepted”	“Rejected” -or- “Accepted, But Errors Were Noted” -and- no “Accepted”	“Rejected” -or- “Accepted, But Errors Were Noted” -and- at least one “Accepted”	All “Rejected”
	The child transaction rejected status is set to:			
Rejected	Accepted	Rejected	Rejected	Rejected
Partially Accepted	Accepted	Accepted, But Errors Were Noted	Partially Accepted	Rejected
Accepted, But Errors Were Noted	Accepted	Accepted, But Errors Were Noted	Accepted, But Errors Were Noted	Rejected

Non-standard

When you are using non-standard processing, EDI Module does not use the *FAGeneration/rejectionRules/childTransactionRejectedStatus* EDITPA variable. Instead, it uses the Child Transaction Rejected Status setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on [page 390](#) in [Appendix A, “Non-Standard Processing”](#).

How the EDI Module Determines Which FA Status to Use

After determining the syntax error status, logical error status, and child transaction rejected status (if applicable), the EDI Module can determine the FA status for a transaction, group, or UN/EDIFACT interchange.

Transaction FA Status

The EDI Module uses the most restrictive value of the following when setting the FA status for a transaction:

- Whether the transaction is allowed; for more information about transactions that are not allowed, see “[Child Transaction that is Not Allowed in its Group](#)” on [page 264](#).
- Whether the transaction is an FA (997 or CONTRL); for more information about transactions that are not allowed, see “[Transaction or Group is for an FA](#)” on [page 264](#).
- Syntax error status; for more information about how the EDI Module sets the syntax error status, see “[Syntax Error Status](#)” on [page 264](#).
- Logical error status; for more information about how the EDI Module sets the logical error status, see “[Logical Error Status](#)” on [page 265](#).

The following table shows the possible combinations of values and how the EDI Module sets the FA status for a transaction based on these values.

The EDI Module reports this FA status for a transaction...	When the transaction has the following statuses...			
	Is transaction allowed?	Is the transaction an FA?	Syntax Error Status	Logical Error Status
Not Allowed	Not Allowed	Regardless of whether it is an FA	Any syntax error status	Any logical error status
FA	Allowed	Is an FA	Any syntax error status	Any logical error status
Rejected	Allowed	Not an FA	■ At least one status is “Rejected”.	

The EDI Module reports this FA status for a transaction...	When the transaction has the following statuses...			
	Is transaction allowed?	Is the transaction an FA?	Syntax Error Status	Logical Error Status
Accepted, But Errors Were Noted	Allowed	<i>Not an FA</i>	<ul style="list-style-type: none"> ■ Statuses are either “Accepted, But Errors Were Noted” -or- “Accepted”. ■ At least one status is “Accepted, But Errors Were Noted”. 	
Accepted	Allowed	<i>Not an FA</i>	Accepted	Accepted

Group FA Status

The EDI Module uses the most restrictive value of the following when setting the FA status for a group:

- Whether the group is “FA” or “CONTRL”, see “[Transaction or Group is for an FA](#)” on [page 264](#).
- Syntax error status; for more information about how the EDI Module sets the syntax error status, see “[Syntax Error Status](#)” on [page 264](#).
- Logical error status; for more information about how the EDI Module sets the logical error status, see “[Logical Error Status](#)” on [page 265](#).
- Child transaction rejected status; for more information about how the EDI Module sets the child transaction rejected error status, see “[Child Transaction Rejected Status](#)” on [page 267](#).

The following table shows the possible combinations of values and how the EDI Module sets the FA status for a group based on these values.

The EDI Module reports this FA status for a group...	When the group has the following statuses...			
	Is group “FA” or “CONTRL”	Syntax Error Status	Logical Error Status	Child Transaction Rejected Status
FA	Is an “FA” or “CONTRL”	Any syntax error status	Any logical error status	Any child transaction rejected status
Rejected	Is <i>not</i> an “FA” or “CONTRL”	■ At least one status is “Rejected”.		
Partially Accepted	Is <i>not</i> an “FA” or “CONTRL”	Accepted, But Errors Were Noted Accepted	Accepted, But Errors Were Noted Accepted	Partially Accepted
Accepted, But Errors Were Noted	Is <i>not</i> an “FA” or “CONTRL”	<ul style="list-style-type: none"> ■ Statuses are either “Accepted, But Errors Were Noted” -or- “Accepted”. ■ At least one status is “Accepted, But Errors Were Noted”. 		
Accepted	Is <i>not</i> an “FA” or “CONTRL”	Accepted	Accepted	Accepted

UN/EDIFACT Interchange FA Status

The EDI Module uses the most restrictive value of the following when setting the FA status for a UN/EDIFACT interchange:

- Syntax error status; for more information about how the EDI Module sets the syntax error status, see “[Syntax Error Status](#)” on page 264.
- Logical error status; for more information about how the EDI Module sets the logical error status, see “[Logical Error Status](#)” on page 265.
- Child transaction rejected status; for more information about how the EDI Module sets the child transaction rejected error status, see “[Child Transaction Rejected Status](#)” on page 267.

The following table shows the possible combinations of values and how the EDI Module sets the FA status for a UN/EDIFACT interchange based on these values.

FA Status of Interchange	When the interchange has the following statuses...		
	Syntax Error Status	Logical Error Status	Child Transaction Rejected Status
Rejected	■ At least one status is “Rejected”.		
Partially Accepted	Accepted, But Errors Were Noted Accepted	Accepted, But Errors Were Noted Accepted	Partially Accepted
Accepted, But Errors Were Noted	■ Statuses are either “Accepted, But Errors Were Noted” -or- “Accepted”. ■ At least one status is “Accepted, But Errors Were Noted”.		
Accepted	Accepted	Accepted	Accepted

Actions the EDI Module Takes Based on FA Status

Based on the EDITPA *splitOption* variable, the EDI Module can be working with an Interchange document, Group document, or Transaction document. For more information, see [“splitOption EDITTPA Variable” on page 119](#).

- For ANSI X12, the Group documents and Transaction documents each have an FA status associated with them.
- For a UN/EDIFACT document, the Interchange, Group, and Transaction documents each have an FA status associated with them.

For more information about how each type of document is assigned an FA status, see [“How the EDI Module Reports the FA Status” on page 263](#).

You set how you want the EDI Module to process a document based on its FA status. The EDI Module can:

- Process the documents with FA statuses that you define as acceptable normally. For how to designate acceptable FA statuses, see [“Defining the FA Statuses that Are Acceptable and Unacceptable” on page 274](#).

To process documents with acceptable FA statuses, the EDI Module:

- 1 Sets the EDI Status custom attribute for the Interchange, Group, or Transaction document to Processed.
- 2 Splits the document, if appropriate:

For an Interchange document	If the <i>splitOption</i> is Group or Transaction and the UN/EDIFACT document contains groups, the EDI Module creates Group documents and processes the Group documents.
	If the <i>splitOption</i> is Group or Transaction and the UN/EDIFACT document does <i>not</i> contain groups, the EDI Module creates Transaction documents and processes the Transaction documents.
For a Group document	If the <i>splitOption</i> is Transaction, the EDI Module creates the Transaction documents and processes the Transaction documents.
- 3 Sends the Interchange, Group, or Transaction document to Trading Networks processing rules.

- Handle documents with FA statuses you define as unacceptable differently. For how to designate unacceptable FA statuses, see “[Defining the FA Statuses that Are Acceptable and Unacceptable](#)” on page 274.

To process documents with unacceptable FA statuses, the EDI Module:

- 1 Sets the EDI Status custom attribute for the Interchange, Group, or Transaction document based on the documents FA status.

<u>When the FA status is:</u>	<u>The EDI Module sets the “EDI Status” custom attribute to:</u>
Not Allowed	Generate FA - Not Allowed
Rejected	Generate FA - Rejected
Partially Accepted	Generate FA - Partially Accepted
Accepted, But Errors Were Noted	Generate FA - Accepted, But Errors Were Noted

- 2 For Interchange and Group documents, saves the document to the Trading Networks database regardless of the setting of the Trading Networks Save Document to Database pre-processing option.

For Transaction documents, does *not* do an automatic save. The document is saved to the Trading Networks database only if instructed to do so by the setting of the Save Document to Database pre-processing option.

- 3 Does *not* split an Interchange or Group document regardless of the setting of the EDITPA *splitOption* variable.

For an Interchange document	If the <i>splitOption</i> is Group or Transaction, the EDI Module does <i>not</i> create Group or Transaction documents.
-----------------------------	--

- 4 Sends the Interchange, Group, or Transaction document that has the unacceptable FA status to Trading Networks processing rules. You should create a processing rule to handle these documents. For more information, see “[Defining Processing Rules for Documents With Unacceptable FA Statuses](#)” on page 279.

Defining the FA Statuses that Are Acceptable and Unacceptable

You set the *FAGeneration/processDocument* EDITPA variable to indicate the acceptable and unacceptable FA statuses. The EDI Module processes documents that have acceptable FA statuses normally and handles documents that have unacceptable FA statuses in a different manner. For more information about how the EDI Module processes the documents based on their FA statuses, see “[Actions the EDI Module Takes Based on FA Status](#)” on page 272.

The following table describes the settings for the *FAGeneration/processDocument* EDITPA variable, when to use each one, and the acceptable and unacceptable FA statuses based on the *processDocument* setting.

Set <i>processDocument</i> to:	When you want to:	Acceptable FA statuses	Unacceptable FA statuses
All	Process <i>all</i> Interchange, Group, and Transaction documents regardless of their FA statuses.	Not Allowed Rejected Partially Accepted Accepted, But Errors Were Noted Accepted	No unacceptable FA status
Only Accepted	Process only Interchange, Group, and Transaction documents that have the FA status “Accepted”.	Accepted	Not Allowed Rejected Partially Accepted Accepted, But Errors Were Noted
Not Rejected	Process all Interchange, Group, and Transaction documents unless they have the FA status “Rejected”.	Not Allowed Partially Accepted Accepted, But Errors Were Noted Accepted	Rejected

The following table describes the settings for the *FAGeneration/processDocument* EDITPA variable and how the EDI Module processes a document based on the document's FA status and the setting of the *FAGeneration/processDocument* EDITPA variable.

Setting of <i>processDocument</i>	FA status of a document	The EDI Module
All	Not Allowed Rejected Partially Accepted Accepted, But Errors Were Noted Accepted	Processes the document with the acceptable FA status normally.
Only Accepted	Accepted	Processes the document with the acceptable FA status normally.
	Not Allowed Rejected Partially Accepted Accepted, But Errors Were Noted	Handles the processing different for these unacceptable FA statuses.
Not Rejected	Accepted Not Allowed Partially Accepted Accepted, But Errors Were Noted	Processes the document with the acceptable FA status normally.
	Rejected	Handles the processing different for these unacceptable FA statuses.

For more information about how the EDI Module processes the documents normally for acceptable FA statuses or handles a document differently due to an unacceptable FA status, see “[Actions the EDI Module Takes Based on FA Status](#)” on page 272.

Non-standard

When you are using non-standard processing, EDI Module does not use the *FAGeneration/processDocument* EDITPA variable. Instead, it uses the Process Document setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in Appendix A, “Non-Standard Processing”.

Steps the EDI Module Takes to Automatically Generate FAs

When the EDI Module receives an EDI document, it performs the following to automatically generate FAs.

- 1 The EDI Module parses an interchange from the EDI document.
- 2 The EDI Module determines whether it is to validate control numbers.
 - If the interchange control number is *not* valid, the EDI Module sets the EDI Status custom attribute for the Interchange document based on the type of invalid control number (duplicate or out-of-sequence) and does *not* continue with generating the FAs.
 - If the interchange control number *is* valid, the EDI Module continues with the next step.
- 3 The EDI Module determines whether it is to automatically generate FAs for the sender/receiver pair. For more information, see [“Turning Automatic FA Generation On or Off” on page 259](#). If automatically generating FAs, continue with the next step.
- 4 If automatic FA generation is on, generate the FAs.
 - For an ANSI X12 document, generate the FAs for the groups in the interchange.
 - For a UN/EDIFACT document, generate the FAs for the interchange.

For information about the format of the generated FAs, see [“Variables that Affect How the EDI Module Generates the FA” on page 260](#). The FAs are *not* sent to Trading Networks processing rules at this time.

In addition to generating the FAs, during this step the EDI Module also obtains information about:

- Whether any child element is not allowed in its envelope
- Syntax errors for the interchange, groups, and transactions
- Logical errors for the interchange, groups, and transactions

Because the EDI Module has information about the syntax errors and logical errors for transactions, it can also determine the child rejection error status for groups and interchanges.

- 5 If the interchange contains groups, the EDI Module works with each Group document that was split from the interchange. The EDI Module:
 - a Determines the FA status for the group. For more information, see [“How the EDI Module Reports the FA Status” on page 263](#).
 - b Based on the FA status for a group, the EDI Module:
 - Processes the Group document normally if it has an acceptable FA status.
 - Handles the Group document differently for unacceptable FA statuses.

For more information, see “[Actions the EDI Module Takes Based on FA Status](#)” on page 272.

- c If the group has an acceptable FA status so the EDI Module is processing the Group document normally, -and- if the EDITPA *splitOption* is Transaction, the EDI Module works with each Transaction document that was split from the group. The EDI Module:
 - 1 Determines the FA status for the transaction. For more information, see “[How the EDI Module Reports the FA Status](#)” on page 263.
 - 2 Based on the FA status for a transaction, the EDI Module:
 - Processes the Transaction document normally if it has an acceptable FA status.
 - Handles the Transaction document differently for unacceptable FA statuses.
- For more information, see “[Actions the EDI Module Takes Based on FA Status](#)” on page 272.
- 3 Sends the Transaction document to Trading Networks processing rules.
- d Sends the Group document to Trading Networks processing rules.
- 6 For a UN/EDIFACT document, if the *splitOption* is Transaction, the EDI Module works with each Transaction document that was split from the interchange. (That is, transactions that are not within group envelopes). The EDI Module:
 - a Determines the FA status for the Transaction. For more information, see “[How the EDI Module Reports the FA Status](#)” on page 263.
 - b Based on the FA status for a Transaction, the EDI Module:
 - Processes the Transaction document normally if it has an acceptable FA status.
 - Handles the Transaction document differently for unacceptable FA statuses.
- For more information, see “[Actions the EDI Module Takes Based on FA Status](#)” on page 272.
- c Sends the Transaction document to Trading Networks processing rules.
- 7 The EDI Module sends the generated FAs to Trading Networks processing rules.

When the FAs are processed through Trading Networks, the EDI recognizer updates the EDITRACKING table for the FAs. The EDI Module uses EDITRACKING table to track FAs and related groups/transactions for FA reconciliation. For more information, see [Chapter 21, “Reconciling Functional Acknowledgments”](#).
- 8 The EDI Module sends the Interchange document to Trading Networks processing rules.

Defining Processing Rules to Process FAs

To specify the processing that you want to perform against an outbound FA that the EDI Module automatically generated, you define a processing rule. Typical processing is to deliver the FA.

When you define a processing rule, you define the criteria that you want Trading Networks to use to select the processing rule and processing actions that you want Trading Networks to perform against the document.

To define a processing rule to deliver outbound FAs

You define processing rules using the Trading Networks Console. For steps to define processing rules, see the chapter about processing rules in the *webMethods Trading Networks User's Guide*. See the table below for more information about creating a processing rule for an outbound FA.

Processing Rule tab:	Description
Criteria	<p>To select outbound FAs that the EDI Module generated, use the following criteria:</p> <ul style="list-style-type: none">■ Receiver. If you want to perform different processing based on the receiver, select specific receivers in the Selected box. For information about the receiver set in the BizDocEnvelope for an outbound FA, see "Contents of the Generated Functional Acknowledgment" on page 262.■ Document Type. The TN document type used for an outbound FA will be one of the following:<ul style="list-style-type: none">■ For an ANSI X12: X12 Envelope.■ For a UN/EDIFACT: EDIFACT EnvelopeSelect the appropriate TN document type in the Selected box.

Processing Rule tab:	Description		
	Attribute	Operator	Value
	EDI Outbound FA	Equals	true
Action	To specify how to deliver the outbound FA, use one of the following processing rule actions:		
<ul style="list-style-type: none"> ■ Execute a Service processing action to invoke the service you created to deliver the FA. <p>-OR-</p> <ul style="list-style-type: none"> ■ Deliver Document By processing action to batch the outbound FA or deliver it to a VAN. For more information, see Chapter 17, “Batching EDI Documents” and Chapter 18, “Retrieving and Delivering EDI Documents from and to VANs”. 			

Defining Processing Rules for Documents With Unacceptable FA Statuses

To set up processing rules for documents with unacceptable FA statuses, use the EDI Status custom attribute in the criteria. The EDI Module uses a Trading Networks custom attribute, EDI Status, to indicate whether a document has an unacceptable FA status. For more information about how the EDI Status custom attribute is set, see information about handling documents with unacceptable FA statuses in [“Actions the EDI Module Takes Based on FA Status” on page 272](#).

When you define a processing rule, you define the criteria that you want Trading Networks to use to select the processing rule and processing actions that you want Trading Networks to perform against the document.

 To define a processing rule to process a document with an unacceptable FA status

You define processing rules using the Trading Networks Console. For steps to define processing rules, see the chapter about processing rules in the *webMethods Trading Networks User’s Guide*. See the table below for more information about creating a processing rule for documents with unacceptable FA statuses.

Processing Rule tab:	Description						
Extended Criteria	<p>Use the Extended Criteria to select documents based on the value of custom attributes. In this case, you select documents based on the value of the EDI Status custom attribute. When a document has an unacceptable FA status, the EDI Status custom attribute will be set to <i>one</i> of the following based on the unacceptable FA status:</p> <ul style="list-style-type: none"> ■ Generate FA - Not Allowed ■ Generate FA - Rejected ■ Generate FA - Partially Accepted ■ Generate FA - Accepted, But Errors Were Noted <p>The actual values that the EDI Status custom attribute might be are based on how you set up automatic FA generation, specifically how you define acceptable and unacceptable FA statuses. For more information, see "Defining the FA Statuses that Are Acceptable and Unacceptable" on page 274.</p> <p>The following shows an example of how to set the extended criteria to select the processing rule for documents that have an unacceptable FA status of "Rejected"; that is, to select the processing rule when the value of the EDI Status custom Attribute is set to Generate FA - Rejected. When specifying the Value, be sure to use the exact combination of upper and lowercase characters.</p> <table border="1"> <thead> <tr> <th>Attribute</th> <th>Operator</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>EDI Status</td> <td>Equals</td> <td>Generate FA - Rejected</td> </tr> </tbody> </table> <p>Important! When you set up processing rules, you must set up a different processing rule for each unacceptable FA status. Do <i>not</i> list all possible values for the EDI Status custom attribute on the Extended Criteria tab. Trading Networks performs an AND operation for all custom attribute criteria. As a result, a document must exhibit <i>all</i> custom attribute criteria that you specify for Trading Networks to select the processing rule. If you define multiple rows for the EDI Status custom attribute, Trading Networks will never use the processing rule because a document will never have the EDI Status custom attribute set to more than one value.</p>	Attribute	Operator	Value	EDI Status	Equals	Generate FA - Rejected
Attribute	Operator	Value					
EDI Status	Equals	Generate FA - Rejected					

Processing Rule tab:	Description
Action	<p>Define the actions that you want to take to handle the EDI document that have unacceptable FA statuses. For example, you might select:</p> <ul style="list-style-type: none">■ Alert e-mail to send an e-mail message to notify an administrator of the problem.■ Execute a Service to invoke a service that you create to perform processing. You might choose to write your own service to send an e-mail message or perform some other type of notification.

Forming EDI Documents to Send Outbound When Using Trading Networks

■ Overview	284
■ Creating the Service to Form the EDI Document	284
■ Setting up Trading Networks to Process Internal-Format Document	291
■ Delivering the EDI Document	291

Overview

You can send documents to webMethods Trading Networks (Trading Networks) from internal applications (e.g., back-end systems) and map information from them into a standard EDI format document. When you use Trading Networks with the webMethods EDI Module (EDI Module), when forming the EDI documents, you can access information for delimiters and headers that the EDI Module maintains in the Trading Networks database.

This chapter describes how to:

- Create a service to form the EDI document from an internal-format document
- Set up Trading Networks to process the internal-format document
- Deliver the outbound EDI document

To learn more about forming outbound EDI documents to send outbound when you are using Trading Networks with the EDI Module, see Chapter 3, "Using the EDI Module with Trading Networks", in the *webMethods EDI Module Concepts Guide*.

Creating the Service to Form the EDI Document

You can send documents to Trading Networks from internal applications (e.g., back-end systems) and create a service that maps information from the internal-format documents into an EDI document. The EDI Module provides built-in services that you can use to form the EDI document.

Additionally, because you are using Trading Networks, you can have the EDI Module maintain the following information that you can access and use when forming the EDI documents:

Information	Where the information is stored
Delimiters to use for the outbound EDI document	EDITPA
The following information that you use for the interchange header: <ul style="list-style-type: none"> ■ Authorization information qualifier ■ Authorization information ■ Security information qualifier ■ Security information 	EDITPA
Control numbers to use in group and interchange headers (or batch, transmission, and file) headers	EDIControlNumber table

Non-standard

For ANSI X12 and UN/EDIFACT users: When using non-standard processing, you obtain the delimiters, Authorization information and qualifier, -and- Security information and qualifier from the interchange sender/receiver pair information that you define using the EDI Module home page. For more information about defining this information, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in Appendix A, “Non-Standard Processing”. For more information about the difference between standard and non-standard processing, see “[Using Standard or Non-Standard Processing](#)” on page 110.

Before Creating the Service to Form an EDI Document

Before you create a service to form an EDI document from an internal-format document, be sure you have the following:

- The flat file schema that defines the structure of the EDI document that you are forming. If you installed the TN document type for this EDI document, the EDI Module installed the flat file schema as well. If you did not install the TN document type for this EDI document, see “[Creating Flat File Schemas for EDI Documents](#)” on page 20.
- Optionally, the flat file schema that defines the structure of the internal-format document. This is needed if your client passes the internal-format document to your service in String format or as an InputStream. Your service uses the flat file schema as input to the `wm.b2b.edi:convertToValues` service (or the `wm.b2b.edi.tradacoms:convertToValues` service) to convert the internal-format document to an IData object and optionally validate the document’s structure. If your service receives the document as an IData object, a flat file schema is not needed. Use the webMethods Developer to create the flat file schema. For more information, see the *Flat File Schema Developer’s Guide*.
- Optionally, an IS document type for the structure of the internal-format document. This is needed if 1) your client passes the internal-format document to your service as an IData object, *and* 2) you want to validate the internal-format document before forming the EDI document. Your service uses the IS document type as input to the `pub.schema:validate` service, which performs the structure validation. For instructions about how to create an IS document type, see the *webMethods Developer User’s Guide*.
- Trading Networks profiles for sender and receiver of the internal-format document. For instructions on how to create profiles, see the *webMethods Trading Networks User’s Guide*.
- Trading Networks profiles for sender and receiver of the outbound EDI document. For instructions on how to create profiles for senders and receivers of EDI documents, see the “[Defining Trading Networks Profiles](#)” on page 113.
- Default and optionally partner-specific EDITPAs for the sender/receiver pairs in the outbound EDI document. For instructions on how to create the EDITPAs, see “[Defining EDI Trading Partner Agreements](#)” on page 116.

Non-standard

For ANSI X12 and UN/EDIFACT users: When using non-standard processing, you also have to define:

- Interchange sender/receiver pair information. For instructions on defining this information, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390 in Appendix A, “Non-Standard Processing”.
- Group-level sender/receiver pair associations. For instructions on defining this information, see “[Defining Group-Level Sender/Receiver Pair Associations](#)” on page 404.

For more information about the difference between standard and non-standard processing, see “[Using Standard or Non-Standard Processing](#)” on page 110.

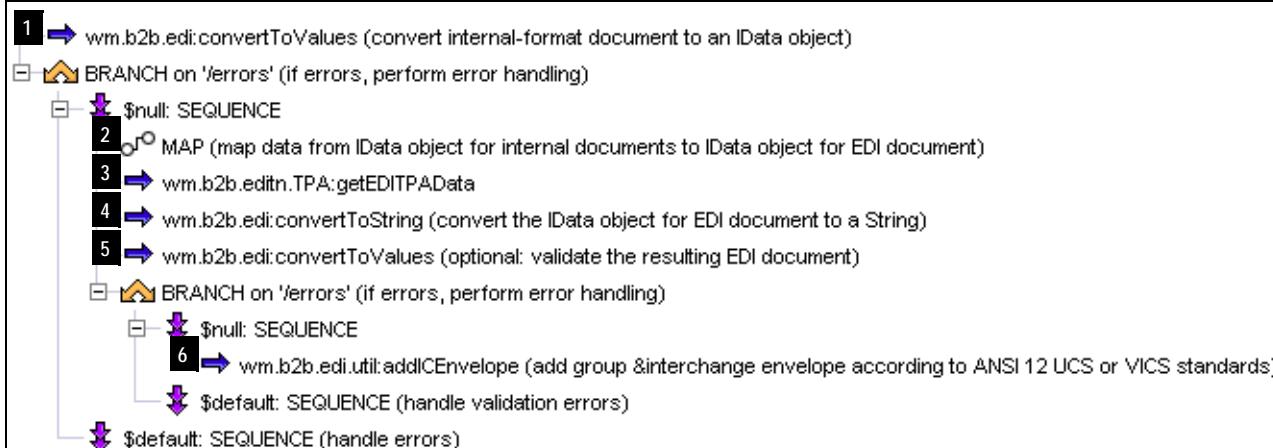
Creating the Service

Non-standard

For ANSI X12 and UN/EDIFACT users: This section describes how to create a service to form an outbound EDI document when you are using standard processing. If you are using non-standard processing, see “[Creating the Service When Using Non-Standard Processing](#)” on page 413 in Appendix A, “Non-Standard Processing”.

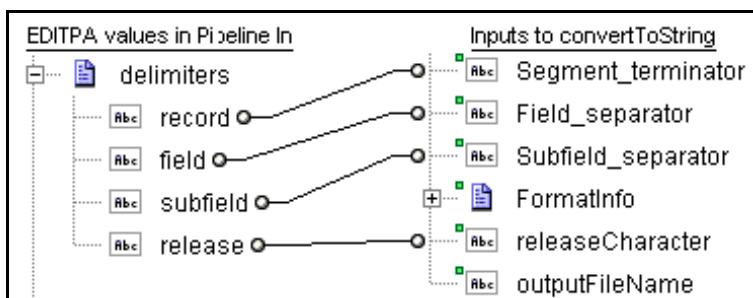
The following sample code illustrates how to form an ANSI X12 EDI document from an internal-format document when using Trading Networks with the EDI Module.

Sample code for forming an ANSI X12 EDI document



Flow operation	Description
1	<p>Invoke the <code>wm.b2b.edi:convertToValues</code> service (or for TRADACOMS documents, invoke the <code>wm.b2b.edi.tradacoms:convertToValues</code> service) to convert the incoming internal-format document that is either a String or <code>InputStream</code> into an <code>IData</code> object. If you want, you can set the input variables of the <code>convertToValues</code> service to have it validate the structure of the internal-format document.</p>
	<p>The inputs to the <code>convertToValues</code> service include the internal-format document and the flat file schema that defines the structure for the internal-format document. For backward compatibility, you can use an <code>IS</code> document type to define the structure of internal-format document rather than a flat file schema. However, it is recommended that you use flat file schemas. For more information about the <code>convertToValues</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
	<p>Note: If the internal-format document is passed to your service as an <code>IData</code> object, you can still validate its structure before forming the EDI document. See “Validating the Input Internal-Format Document When it is an <code>IData</code> Object” on page 91.</p>
2	<p>Map data from the internal-format document <code>IData</code> object into the EDI document <code>IData</code> object. Depending on the complexity of your mapping requirements, you might need to add more logic than a MAP flow operation, or create a separate service to perform the mapping. For more information about how to map, see Chapter 5, “Mapping Data to Form New Documents”.</p>
3	<p>Invoke the <code>wm.b2b.editn.TPA:getEDITPAData</code> service to obtain the EDITPA data that contains delimiters and header information you can use for the outbound EDI document. For more information about the <code>getEDITPAData</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>

Flow operation	Description
4	<p>Invoke the <code>wm.b2b.edi:convertToString</code> service (or the <code>wm.b2b.edi.tradacoms:convertToString</code> service) to convert the EDI document from an <code>IData</code> object to String format.</p> <p>The inputs to the <code>convertToString</code> service include:</p> <ul style="list-style-type: none"> ■ The <code>IData</code> object that contains the data for your EDI document. Map this <code>IData</code> object to the input variable <i>Values</i> of the <code>convertToString</code> service. ■ The flat file schema for the EDI document. The <code>convertToString</code> service uses the flat file schema to determine how to form the EDI document. <p>For backward compatibility, you can use an <code>IS</code> document type as input to the <code>convertToString</code> service rather than a flat file schema for files with delimited fields and records.</p> <ul style="list-style-type: none"> ■ The delimiters to use when creating the EDI document. Map the delimiters from the EDITPA to the inputs of the <code>convertToString</code> service:



-
- | | |
|---|--|
| 5 | Optionally, invoke the <code>convertToValues</code> service against the EDI document to validate the structure of your final EDI document. |
|---|--|

The inputs to the `convertToValues` service include:

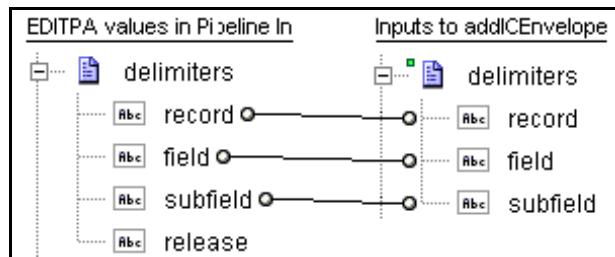
- The EDI document. The output variable *string* from the `convertToString` service contains the EDI document. Map this to the input variable *edidata* of the `convertToValues` service.
- The flat file schema that defines the structure for the EDI document.

-
- | | |
|---|---|
| 6 | Invoke the <code>wm.b2b.edi.util:addICEnvelope</code> service to add the interchange and group envelope to the EDI document. If you are creating a UN/EDIFACT EDI document, use the <code>wm.b2b.edi.util:addICEnvelopeEDIFACT</code> service. For more information, see “Adding UN/EDIFACT Envelopes” on page 91 . |
|---|---|

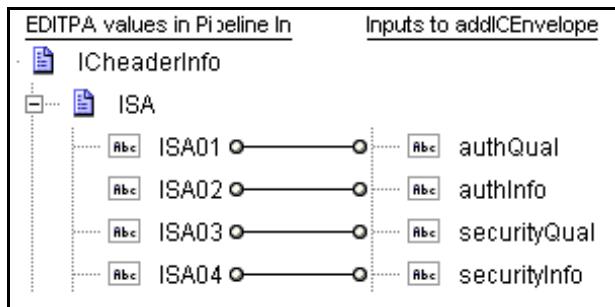
Flow operation	Description
<p>Note: If you are creating a TRADACOMS EDI document, use the services in the <code>wm.b2b.edi.tradacoms.compose</code> folder to add a transmission envelope and a batch envelope (if specified) and to obtain control numbers. For details, see “Adding TRADACOMS Envelopes and Obtaining Control Numbers” on page 290.</p>	

When setting the inputs to the `addICEnvelope`:

- Map the delimiters from the `EDITPA` to the inputs of the `addICEnvelope` service:



- Map the information from the `EDITPA` to the inputs of the `addICEnvelope` service:



- Set the `ctlFromTable` variable to `true` to have the service obtain the control number. See [“Obtaining Control Numbers for Outbound Processing \(ANSI X12 and UN/EDIFACT\)” on page 290](#) for more information.

For information about delivering the document, see [“Delivering the EDI Document” on page 291](#).

Obtaining Control Numbers for Outbound Processing (ANSI X12 and UN/EDIFACT)

The EDI Module maintains information about control numbers in the EDIControlNumber table, which is an EDI Module-specific table in the Trading Networks database.

For ANSI X12 and UN/EDIFACT documents, when you create services to form outbound EDI documents, you can invoke the following services that webMethods provides to add group and interchange headers to the outbound EDI document:

ANSI X12	UN/EDIFACT
wm.b2b.edi.util:addGroupEnvelope	wm.b2b.edi.util:addGroupEnvelopeEDIFACT
wm.b2b.edi.util:addICEnvelope	wm.b2b.edi.util:addICEnvelopeEDIFACT

When invoking one of the above services to add a header, use the *ctlFromTable* or *grpCtlNumber* input variable to indicate whether you want to use the control numbers from the EDIControlNumber table for group and interchange headers. If the EDIControlNumber table does not have an entry for the group or interchange sender/receiver pair, a row is added to the table for the sender/receiver pair and the control number is set to 1.

You can set the initial values you want to use for sender/receiver pairs using the EDI Module home page. For instructions, see “[Defining Control Number Settings](#)” on [page 177](#).

Adding TRADACOMS Envelopes and Obtaining Control Numbers

The WmEDI package provides the following built-in services you can use to add TRADACOMS transmission and batch envelopes:

- `wm.b2b.edi.tradacoms.compose:startTradacomsTransmission`

Invoke this service to create an STX segment for a transmission. This service returns a *tradacomsTransmission* object.

- `wm.b2b.edi.tradacoms.compose:addToTradacomsTransmission`

Invoke this service to add the header message to the TRADACOMS transmission object that the `startTradacomsTransmission` service returned.

Invoke the `addToTradacomsTransmission` service again to add the detail messages to the TRADACOMS transmission object.

Invoke the `addToTradacomsTransmission` service again to add the trailer message to the TRADACOMS transmission object.

- `wm.b2b.edi.tradacoms.compose:endTradacomsTransmission`

Invoke this service to create an END segment for the transmission.

- `wm.b2b.edi.tradacoms.compose:startTradacomsBatch`

Invoke this service to create a batch (BAT) segment for the transmission.

- Finally, invoke the built-in service `wm.b2b.edtn.batch:getControlNumber` (which is provided in the `WmEDIforTN` package) to obtain the current control number from the `EDIControlNumber` table. This table is an EDI Module-specific table in the Trading Networks database

For more information about these services, see the *webMethods EDI Module Built-In Services Reference*.

Setting up Trading Networks to Process Internal-Format Document

To set up Trading Networks to receive an internal-format document and use information in it to form an EDI document, define the following in Trading Networks:

- Profiles for the sender and receiver of the internal-format document. For instructions about how to define profiles, see the *webMethods Trading Networks User's Guide*.
- TN document type for the internal-format document. For instructions about how to define TN document type, see the *webMethods Trading Networks User's Guide*.
- Processing rule for the internal-format document. For instructions about how to define processing rules, see the *webMethods Trading Networks User's Guide*. When you define the processing rule, use the Execute a Service processing action to invoke the service you created to form the EDI document. For more information about how to create the service to form the EDI document, see ["Creating the Service to Form the EDI Document" on page 284](#).

After forming the EDI document, you can deliver it. For more information about delivering it, see ["Delivering the EDI Document" on page 291](#).

Delivering the EDI Document

This section describes three methods you can use to deliver the EDI document:

- Add logic to the service that forms the EDI document to deliver it. For more information, see ["Deliver the EDI Document Directly from the Service that Forms It"](#) below.
- Submit the EDI document back to Trading Networks document recognition to have the EDI recognizer within Trading Networks recognize the EDI document, and then use a processing rule to deliver the EDI document. For more information, see ["Submitting the EDI Document to Trading Networks" on page 293](#).

- Route the EDI document back to Trading Networks processing rule to use a processing rule to deliver the document. For more information, see ["Routing the Outbound EDI Document to Trading Networks" on page 301](#).

To learn more about these three methods for delivering outbound EDI documents, see Chapter 3, "Using the EDI Module with Trading Networks" in the *webMethods EDI Module Concepts Guide*.

Deliver the EDI Document Directly from the Service that Forms It

In the service that you create to form the EDI document, you can add your own logic to deliver the outbound EDI document.

When you deliver a document directly from the service that forms the EDI document, if you want to use the functional acknowledgment (FA) reconciliation feature of the EDI Module, you must also invoke the `wm.b2b.edtn:trackEDIdocs` service.



Note: Functional acknowledgments (FAs) are not applicable to the TRADACOMS standard.

This service updates the EDITRACKING table that keeps track of group documents and their corresponding FAs. To use the FA reconciliation feature, you must enable FA reconciliation on a per partner pair basis in the EDITPA. For more information, see ["FAReconciliation EDITPA Variable" on page 136](#). For more information about how to generate FA reconciliation reports, see ["Creating FA Reconciliation Reports" on page 368](#) in [Chapter 21, "Reconciling Functional Acknowledgments"](#).

The following code sample shows a portion of the service to form EDI documents that is described in ["Creating the Service to Form the EDI Document" on page 284](#) with additional flow operations added to invoke:

- The `wm.b2b.edtn:trackEDIdocs` service to do FA reconciliation.
- A service that you create to deliver the EDI document.

Sample code that delivers the outbound EDI document from the service that forms it

- `wm.b2b.edi.util:addICEnvelope` (add group &interchange envelope according to ANSI 12 UCS or VICS standards)
- `wm.b2b.edtn:trackEDIdocs` (optional: update info for FA reconciliation)
- `CustomCode:SendEDIDocument`

Submitting the EDI Document to Trading Networks

To submit the outbound EDI document back to Trading Networks, the document goes back through Trading Networks document recognition. This means that the EDI Module-specific recognizer, the *EDI recognizer*, receives control to perform processing on the EDI document.

You define a processing rule to indicate how to deliver the outbound EDI document.

Specifying Variables that Affect Outbound Processing (ANSI X12 and UN/EDIFACT)

Because the EDI document goes through document recognition, the EDITPA variables that affect inbound processing also affect the outbound EDI document. For more information, see “[Specifying EDITPA Variables that Affect Inbound Processing](#)” on page 196.

Note that the EDI recognizer will automatically perform FA reconciliation if enabled by the *FAReconciliation* EDITPA variable. For more information, see “[FAReconciliation EDITPA Variable](#)” on page 136. For more information about how to generate FA reconciliation reports, see [Chapter 21, “Reconciling Functional Acknowledgments”](#).

The following table lists the EDITPA variables that you should set differently for EDITPAs used for outbound EDI documents.

EDITPA variable	Description
<i>splitOption</i>	<p>The EDI recognizer will split the EDI document based on the value of the EDITPA <i>splitOption</i> variable. For more information, see “splitOption EDITTPA Variable” on page 119. For an outbound document, you should set the <i>splitOption</i> variable to one of the following:</p> <ul style="list-style-type: none"> ■ <i>Interchange</i> if you only want to deliver the EDI document and do not want to perform FA reconciliation. ■ <i>Group</i> if you want to perform FA reconciliation in addition to delivering the document. Note if the <i>splitOption</i> variable is set to <i>Interchange</i>, but the <i>FAReconciliation</i> variable is <i>true</i> to enable FA reconciliation, the EDI recognizer will split at the group level.

EDITPA variable	Description
<i>ControlNumberManagement/validateInboundEnvelopeControlNumbers</i>	<p>For an outbound document, you should set this EDITPA variable to <code>false</code>, so the EDI Module does not attempt to validate the interchange control numbers in the outbound document. For more information about this EDITPA variable, see “ControlNumberManagement/validateInboundEnvelopeControlNumbers EDITPA Variable” on page 124.</p>
Non-standard	<p>When you are using non-standard processing, EDI Module does not use this EDITPA variable. Instead, it uses the Validate inbound envelope control numbers setting that you set from the Interchange Information Detail screen of the EDI Module home page. For more information about accessing and using this screen, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 390 in Appendix A, “Non-Standard Processing”.</p>
<i>ControlNumberManagement/validateInboundGroupControlNumbers</i>	<p>For an outbound document, you should set this EDITPA variable to <code>false</code>, so the EDI Module does not attempt to validate the group control numbers in the outbound document. For more information about this EDITPA variable, see “ControlNumberManagement/validateInboundGroupControlNumbers EDITPA Variable” on page 125.</p>
<i>FAGeneration/autoGenerateFA</i>	<p>For an outbound document, you should set this EDITPA variable to <code>off</code>, so the EDI Module does not automatically generate FAs for your outbound document.</p>

Specifying Variables that Affect Outbound Processing (TRADACOMS)

Because the EDI document goes through document recognition, the EDITPA variables that affect inbound processing also affect the outbound EDI document. For more information, see [“Specifying EDITPA Variables that Affect Inbound Processing” on page 222](#).

The following table lists the EDITPA variables that you should set for EDITPAs used for outbound EDI documents.

TRADACOMS/splitOption EDITTPA Variable

The EDI recognizer will split the EDI document based on the value of the *TRADACOMS/splitOption* EDITTPA variable.

<i>splitOption</i> Value	Description
Transmission	The EDI Module delivers only the Transmission document.
Batch	The EDI Module delivers the Transmission document and a Batch document for each batch segment in the transmission segment.

TRADACOMS/Outbound/createReconciliationMessage EDITTPA Variable

Optional. Use this variable to create a reconciliation (RSGRSG) message in outbound batched EDI documents.

<i>createReconciliationMessage</i> Value	Description
true	Creates an RSGRSG message at the end of the transmission for the batched TRADACOMS document.
false	(Default) Does not create an RSGRSG message.

TRADACOMS/Outbound/recipientTransmissionReference EDITTPA Variable

Optional. Use this variable to return the receiver's transmission reference in outbound EDI documents.

<i>recipientTransmissionReference</i> Value	Description
true	The EDI Module returns the receiver's transmission reference in outbound EDI documents.
false	(Default) The EDI Module does not return the receiver's transmission reference.

TRADACOMS/Outbound/applicationReference EDITTPA Variable

Optional. Use this variable to return the value of the receiver's application reference field (the APRF field in the STX segment) in outbound EDI documents.

Note: This variable is ignored if the *TRADACOMS/Outbound/applicationReferenceFromFile* variable is set to true.

TRADACOMS/Outbound/applicationReferenceFromFile EDITTPA Variable

Optional. Use this variable to return the value of the receiver's application reference field (the APRF field in the STX segment) in outbound EDI documents. In this case, the value of the APRF field is the file type specified in the transmission. For example, if the transmission contains ORDERS files, the value of the APRF field will be ORDHDR.

<i>applicationReference</i> <i>FromFILE</i> Value	Description
true	The EDI Module returns the value of the APRF field, which is the file type specified in the transmission, for example ORDHDR.
false	(Default) The EDI Module uses the value specified for the <i>TRADACOMS/Outbound/applicationReference</i> variable.

Setting up Trading Networks to Deliver the EDI Document

The Trading Networks objects that you need to create vary based on whether you set the *splitOption* variable to Interchange (or Transmission) or Group (or Batch).

- If the *splitOption* variable is Interchange (or Transmission), define the following in Trading Networks:
 - Profiles for the senders and receivers in the outbound EDI document. For instructions about how to define profiles, see:
 - For ANSI X12 or UN/EDIFACT: “[Defining Trading Networks Profiles](#)” on page 113
 - For TRADACOMS: “[Defining Trading Networks Profiles](#)” on page 147

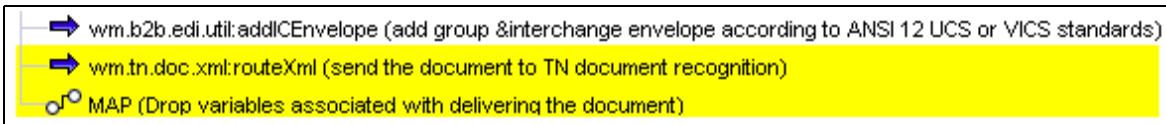
- TN document type for the Interchange (or Transmission) document that the EDI recognizer creates from the outbound EDI document. For instructions about how to define TN document types, see the *webMethods Trading Networks User's Guide*.
 - Processing rule for the Interchange (or Transmission) document. For instructions about how to define processing rules, see the *webMethods Trading Networks User's Guide*. When you define the processing rule, use the:
 - Execute a Service processing action to invoke the service you created to deliver the EDI document.
- OR-
- Deliver Document By processing action to batch the outbound EDI document or deliver it to a VAN. For more information, see [Chapter 17, "Batching EDI Documents"](#), and [Chapter 18, "Retrieving and Delivering EDI Documents from and to VANs"](#).
 - If the *splitOption* variable is Group (or Batch), define the following in Trading Networks:
 - Profiles for the senders and receivers in the outbound EDI document. For instructions about how to define profiles, see:
 - For ANSI X12 or UN/EDIFACT: ["Defining Trading Networks Profiles" on page 113](#)
 - For TRADACOMS: ["Defining Trading Networks Profiles" on page 147](#)
 - TN document type for the Interchange *and* Group (or Transmission *and* Batch) documents that the EDI recognizer creates from the outbound EDI document. For instructions about how to define TN document types, see the *webMethods Trading Networks User's Guide*.
 - Processing rule for the Interchange *and* Group (or Transmission *and* Batch) documents. For instructions about how to define processing rules, see the *webMethods Trading Networks User's Guide*. When you define the processing rule, for the Interchange (or Batch) document, use the:
 - Execute a Service processing action to invoke the service you created to deliver the EDI document.
- OR-
- Deliver Document By processing action to batch the outbound EDI document or deliver it to a VAN. For more information, see [Chapter 17, "Batching EDI Documents"](#), and [Chapter 18, "Retrieving and Delivering EDI Documents from and to VANs"](#).

Adding Logic to Service that Forms the EDI Document to Submit It to Trading Networks

To submit a document to Trading Networks document recognition, in the service that you created to form the EDI document, invoke the `wm.tn.doc.xml:routeXml` service.

The following code sample shows a portion of the service to form EDI documents that is described in [“Creating the Service to Form the EDI Document” on page 284](#) with an additional **Invoke** flow operation added to invoke the `wm.tn.doc.xml:routeXml` service.

Sample code that submits the outbound EDI document back to Trading Networks



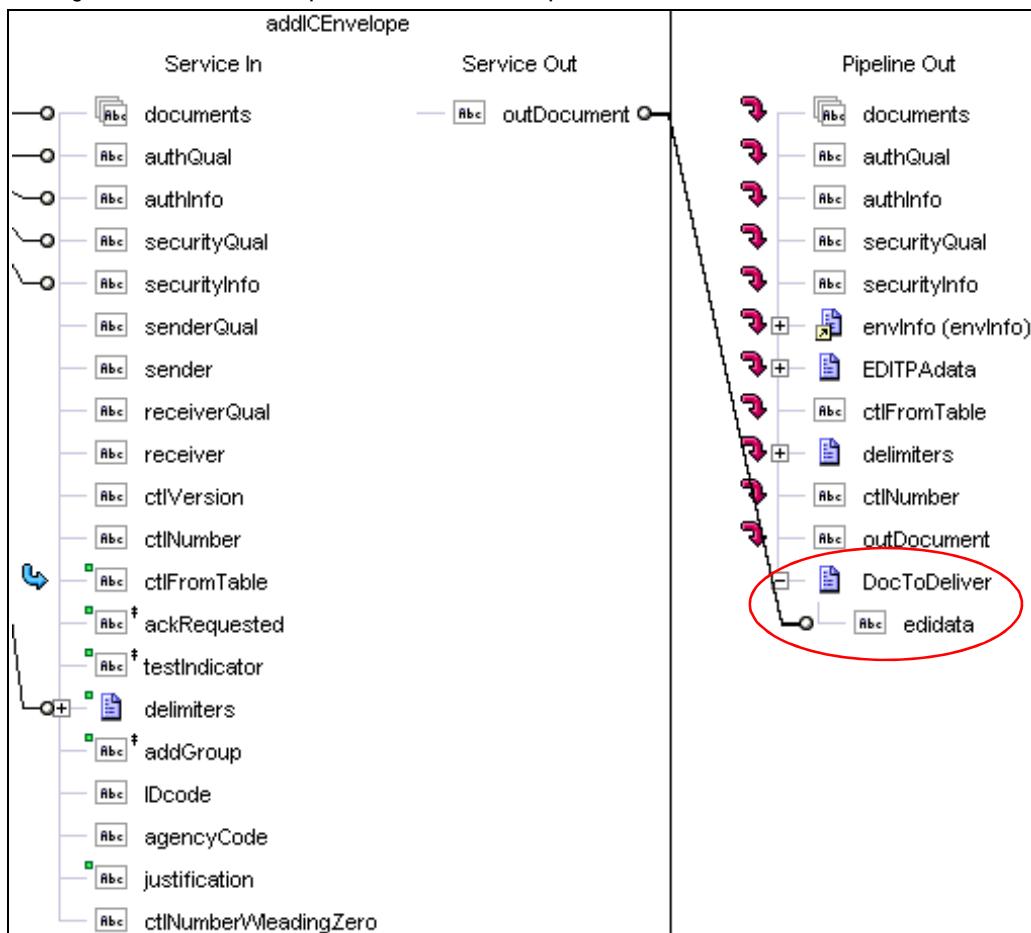
Preparing the Pipeline to Invoke the `wm.tn.doc.xml:routeXml` Service

When you invoke the `wm.tn.doc.xml:routeXml` service, you should have the document in the `edidata` variable in Pipeline In. Previous operations in this service placed data in an `edidata` variable. The following describes how to map an ANSI X12 document to deliver into the `edidata` variable without losing the value of the current `edidata` variable.

First, select the **Invoke** `wm.b2b.edi.util:addlCEnvelope` flow operation.

Then, in Pipeline Out, create a document. In the illustration below, the new document is named `DocToDeliver`. Within the document, create a String named `edidata`. Then map the output from the `addlCEnvelope` service, which is in the `outDocument` variable, to your new `edidata` variable, as shown below:

Creating a Document to hold output from the addICEnvelope service



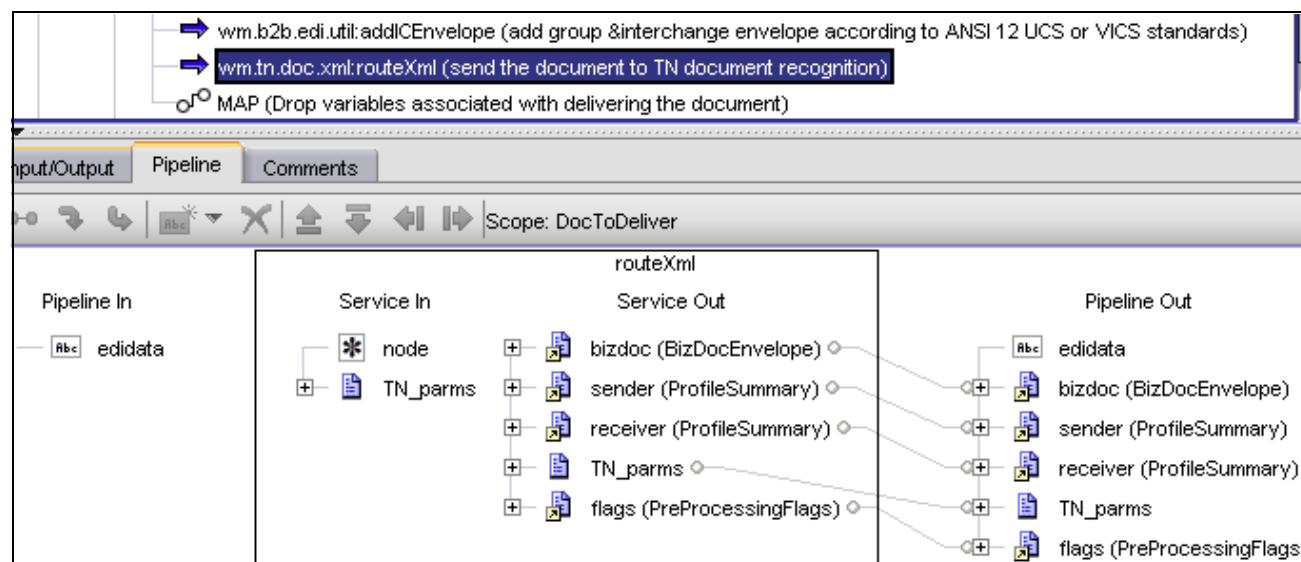
Click on the **Invoke** `wm.tn.doc.xml:routeXml` flow operation and view its properties. In the Properties panel, set the **Scope** property to the name of the document that you created to hold the output from the addICEnvelope service (e.g., *DocToDeliver*). This causes the **Invoke** `wm.tn.doc.xml:routeXml` flow operation to execute using only the pipeline variables within the specified scope document.

Set the Scope property to limit the elements in the pipeline when the routeXml service executes

Properties	
	wm.tn.doc.xml:routeXml
Property	Value
General	
Comments	send the document to TN document recognition
Scope	DocToDeliver
Timeout	
Label	
Service	wm.tn.doc.xml:routeXml
Validate input	False
Validate output	False

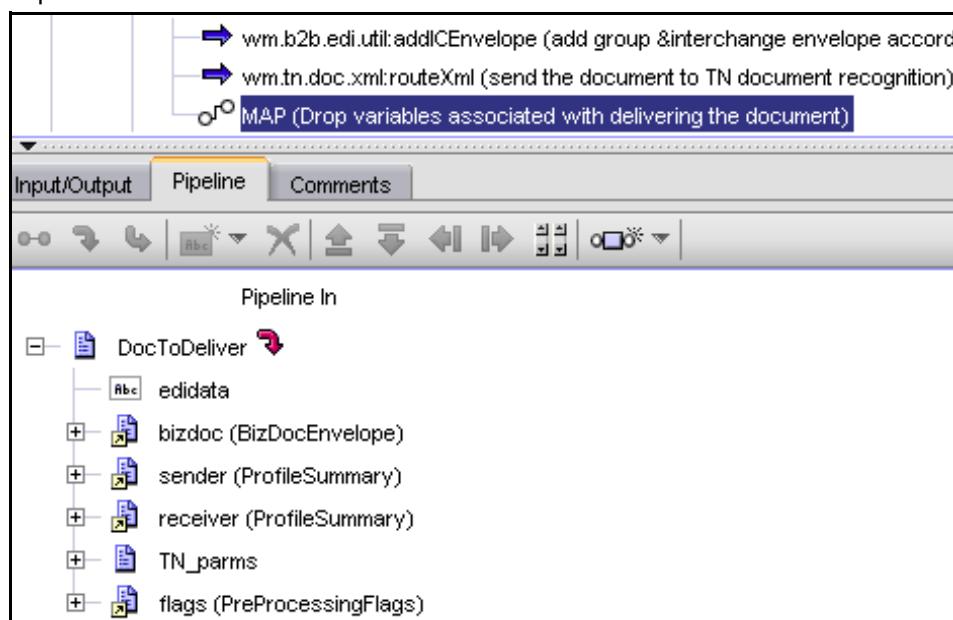
After setting the scope, when you select the **INVOKE** `wm.tn.doc.xml:routeXml` flow operation, Pipeline In contains only the *edidata* variable for:

Pipeline after setting scope for the INVOKE routeXml flow operation



When you use scope, the output from the `wm.tn.doc.xml:routeXml` service also goes into the document that you used for the scope (e.g., `DocToDeliver`). After invoking the `wm.tn.doc.xml:routeXml` service, you can drop all variables in the document that you created, as shown below:

Drop variables



Routing the Outbound EDI Document to Trading Networks

To route the EDI document through Trading Networks, the document goes back to Trading Networks processing rules, bypassing document recognition. The result of document recognition is a BizDocEnvelope that is sent to the processing rules. Because you are bypassing document recognition, you must form the BizDocEnvelope that you send to the processing rule. You send the BizDocEnvelope to the processing rules by invoking the `wm.tn.route:routeBizdoc` service. For more information about how to form the BizDocEnvelope, see [“Creating a BizDocEnvelope for the Outbound EDI Document” on page 303](#).

Additionally, when you route a document through Trading Networks, if you want to use the functional acknowledgment (FA) reconciliation feature of the EDI Module, you must also invoke the `wm.b2b.editn:trackEDIdocs` service. This service updates the EDITRACKING table that keeps track of group documents and their corresponding FAs. To use the FA reconciliation feature, you must enable FA reconciliation on a per partner pair basis in the EDITPA. For more information, see [“FAReconciliation EDITPA Variable” on page 136](#). For more information about how to generate FA reconciliation reports, see [Chapter 21, “Reconciling Functional Acknowledgments”](#).

Setting up Trading Networks to Deliver the EDI Document

To set up Trading Networks to route the outbound EDI document, define the following in Trading Networks:

- Profiles for the senders and receives in the outbound EDI document. For instructions about how to define profiles, see:
 - For ANSI X12 or UN/EDIFACT: “[Defining Trading Networks Profiles](#)” on [page 113](#)
 - For TRADACOMS: “[Defining Trading Networks Profiles](#)” on [page 147](#)
 - Processing rule for the outbound EDI document. For instructions about how to define processing rules, see the *webMethods Trading Networks User’s Guide*. When you define the processing rule, use the:
 - Execute a Service processing action to invoke the service you created to deliver the EDI document.
- OR-
- Deliver Document By processing action to batch the outbound EDI document or deliver it to a VAN. For more information, see [Chapter 17, “Batching EDI Documents”](#), and [Chapter 18, “Retrieving and Delivering EDI Documents from and to VANs”](#).

Adding Logic to Service that Forms the EDI Document to Route It through Trading Networks

To route a document through Trading Networks, in the service that you created to form the EDI document, invoke the `wm.tn.route:routeBizdoc` service.

The following code sample shows a portion of the service to form EDI documents that is described in “[Creating the Service to Form the EDI Document](#)” on [page 284](#) with additional flow operations added to invoke:

- The `wm.b2b.edith:trackEDIdocs` service to do FA reconciliation.
- A service that you define to create a `BizDocEnvelope` for the outbound EDI document (the sample invokes the `createEDIBizDoc` service). For more information about how to create this service, see “[Creating a BizDocEnvelope for the Outbound EDI Document](#)” on [page 303](#).
- The `wm.tn.route:routeBizdoc` service to send the `BizDocEnvelope` to the Trading Networks processing rules.

Sample code that routes the outbound EDI document through Trading Networks

- ▶ `wm.b2b.edi.util:addICEnvelope` (add group &interchange envelope according to ANSI 12 UCS or VICS standards)
- ▶ `wm.b2b.edtn:trackEDIdocs` (optional: update info for FA reconciliation)
- ▶ `CustomCode:createEDIBizDoc` (create a BizDocEnvelope for the outbound EDI document)
- ▶ `wm.tn.route:routeBizdoc` (send the BizDocEnvelope with EDI document to TN processing rules)

Creating a BizDocEnvelope for the Outbound EDI Document

This section describes how to create the `createEDIBizDoc` service that is used in the code sample in [“Adding Logic to Service that Forms the EDI Document to Route It through Trading Networks”](#) on page 302.

The service creates a BizDocEnvelope and adds the outbound EDI document to the BizDocEnvelope as a content part.

Input Variables for the `createEDIBizDoc` Service

Input Variable	Description
<i>OutboundEDIDoc</i>	The outbound EDI document in String format.
<i>sender</i>	The sender ID for the sender of the outbound EDI document. This is the sender at the interchange or transmission level.
<i>senderQualifier</i>	The EDI ID qualifier for the sender, for example, <code>01</code> if you specify a D-U-N-S number for <i>sender</i> .
<i>receiver</i>	The receiver ID for the receiver of the outbound EDI document. This is the receiver at the interchange or transmission level.
<i>receiverQualifier</i>	The EDI ID qualifier for the receiver, for example, <code>01</code> if you specify a D-U-N-S number for <i>receiver</i> .
<i>standard</i>	The EDI standard of the outbound EDI document; <i>standard</i> should be one of the following: <ul style="list-style-type: none"> ■ <code>x12</code> for an ANSI X12 document ■ <code>UNEDIFACT</code> for a UN/EDIFACT document ■ <code>TRADACOMS</code> for a TRADACOMS document

Input Variable	Description
<i>controlNumber</i>	The interchange or transmission control number to use for the Trading Networks DocumentID and GroupID system attributes.
<i>encoding</i>	Optional. The name of a registered, IANA character set (e.g., ISO-8859-1) that is used for input to the pub.string:stringToBytes service, which is used to convert the EDI document into byte[] format. To use the default encoding, set this value to autoDetect. If you specify an unsupported encoding, an exception will be thrown.

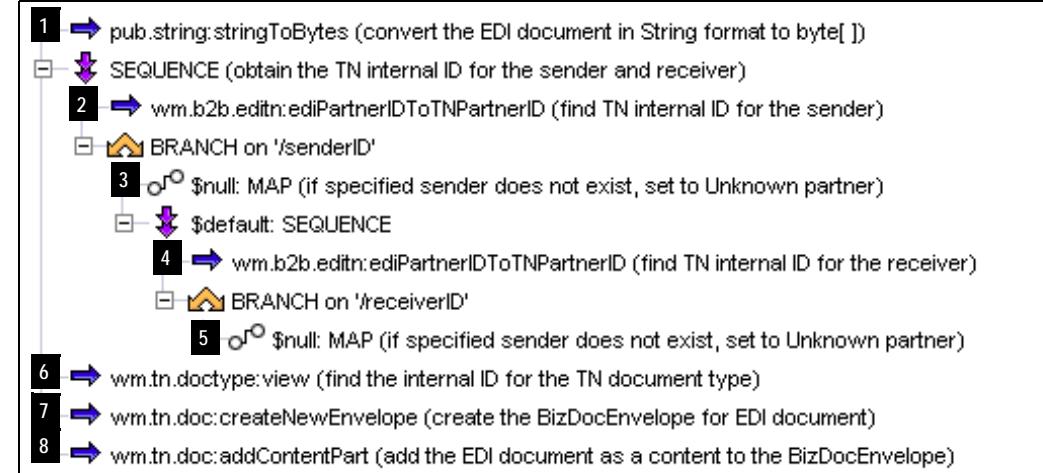
Output Variable for the createEDIBizDoc Service

Output Variable	Description
<i>bizdoc</i>	The BizDocEnvelope for the outbound EDI document. This is a Document Reference to the IS document type, <code>wm.tn.rec:BizDocEnvelope</code> .

Logic for the Service to Create a BizDocEnvelope for the Outbound EDI Document

The following shows the flow operations to use for the service.

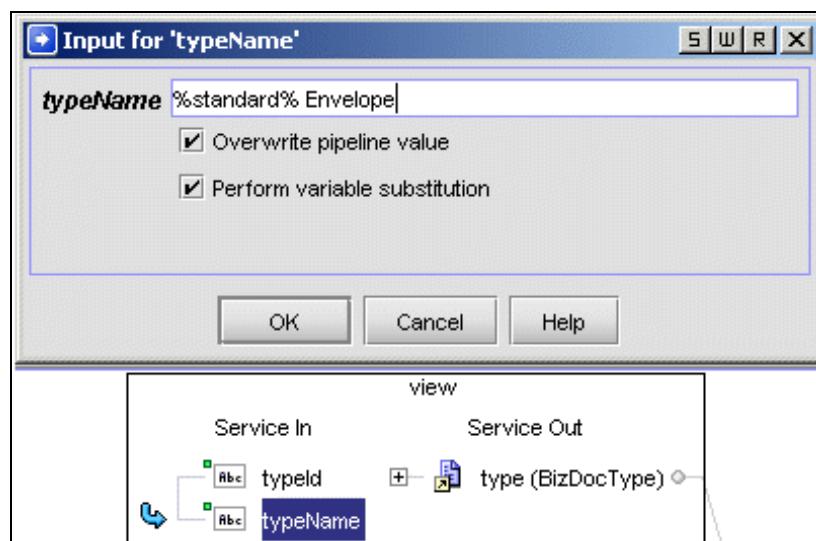
Service to create a BizDocEnvelope for the outbound EDI document



Flow operation	Description
1	<p>Invoke the pub.string:stringToBytes service to convert the outbound EDI document from String format to a byte[]. Map the input variables of the createBizDoc service, <i>OutboundEDIDoc</i> and <i>encoding</i> to the input variables of the stringToBytes service, <i>string</i> and <i>encoding</i>, respectively.</p>
	<p>Map the output of the stringToBytes service, bytes</p>
	<p>Convert the format to a byte[] because the service to add the content part to the BizDocEnvelope requires the content part to be in byte[] format.</p>
2	<p>Invoke the wm.b2b.edtn:ediPartnerIDToTNPartnerID service to retrieve the Trading Networks internal ID for the sender. Map the input variables of the createBizDoc service, <i>sender</i> and <i>senderQualifier</i> to the input variables of the ediPartnerIDToTNPartnerID service, <i>identifier</i> and <i>qualifier</i>, respectively.</p>
	<p>Map the output of the ediPartnerIDToTNPartnerID service (<i>id</i>), to <i>senderID</i>, which is a variable that you define in Pipeline Out.</p>
3	<p>Use a BRANCH operation to branch based on whether the <i>senderID</i> is null. If <i>senderID</i> is null, use a MAP flow operation to set the value of <i>senderID</i> to 000000000000000000000000 (24 zeros), which indicates the Unknown partner.</p>
4	<p>Invoke the wm.b2b.edtn:ediPartnerIDToTNPartnerID service to retrieve the Trading Networks internal ID for the receiver. Map the input variables of the createBizDoc service, <i>receiver</i> and <i>receiverQualifier</i> to the input variables of the ediPartnerIDToTNPartnerID service, <i>identifier</i> and <i>qualifier</i>, respectively.</p>
	<p>Map the output of the ediPartnerIDToTNPartnerID service, <i>id</i>, to <i>receiverID</i>, which is a variable that you define in Pipeline Out.</p>
5	<p>Use a BRANCH operation to branch based on whether the <i>receiverID</i> is null. If <i>receiverID</i> is null, use a MAP flow operation to set the value of <i>receiverID</i> to 000000000000000000000000 (24 zeros), which indicates the Unknown partner.</p>

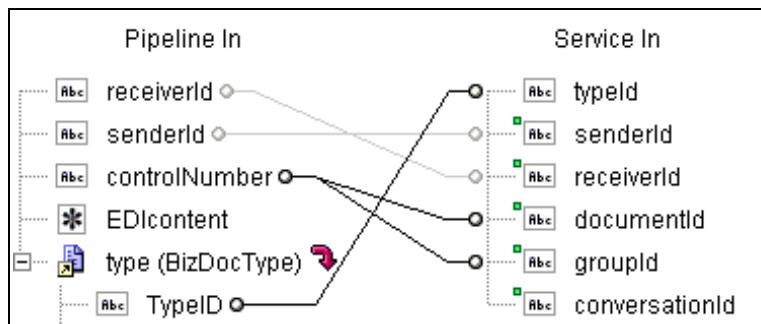
Flow operation	Description
6	Invoke the <code>wm.tndoctype:view</code> service to determine the Trading Networks internal ID for the TN document type to use for the outbound EDI document. Specify the name of the TN document type as input, and you can retrieve the internal ID from the output.

The name of the TN document type will be either 1) X12 Envelope, 2) UNEDIFACT Envelope, or 3) TRADACOMS Transmission. For example, to set this value at run time, set the value of the input variable `typeName` for the view service as follows:

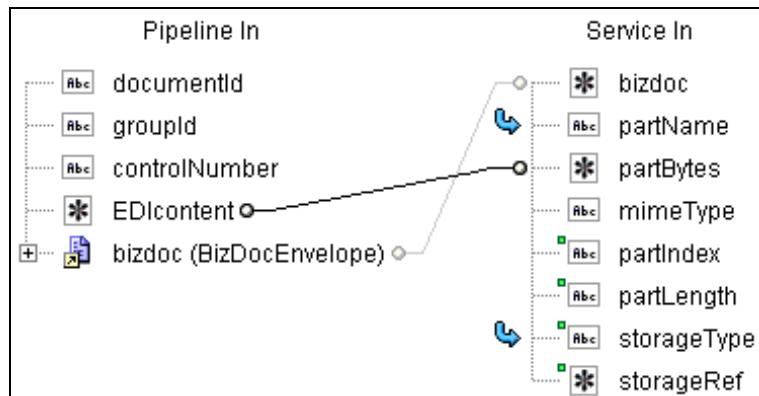


The variable substitution `%standard%` picks up the value of the input variable of the `createBizDoc` service, `standard`, which should be `x12`, UNEDIFACT, or TRADACOMS.

Flow operation	Description
7	<p>Invoke the <code>wm.tn.doc:createNewEnvelope</code> service to create the <code>BizDocEnvelope</code> for the outbound EDI document. To set the inputs to the <code>createNewEnvelope</code> service, map:</p> <ul style="list-style-type: none"> ■ The Trading Networks internal IDs for the sender and receiver, which are in Pipeline In in the <code>senderID</code> and <code>receiverID</code> variables, to the Service In variables, <code>senderID</code> and <code>receiverID</code>, respectively. ■ The input variable of the <code>createBizDoc</code> service, <code>controlNumber</code>, which is in Pipeline In, to the Service In variables, <code>documentID</code> and <code>groupId</code>. ■ The Trading Networks internal ID for the TN document type, which is in Pipeline In in the <code>type/TypeID</code> variable to the Service In variable, <code>typeID</code>.



Flow operation	Description
8	<p>Invoke the <code>wm.tn.doc:addContentPart</code> service to add the outbound EDI document (in <code>byte[]</code> format) to the <code>BizDocEnvelope</code>. To set the inputs to the <code>addContentPart</code> service:</p> <ul style="list-style-type: none"> ■ Map the content of the outbound EDI document that is in <code>byte[]</code> format and which is in Pipeline In in the <code>EDIcontent</code> variable, to the Service In variable, <code>partBytes</code>. ■ Set the value of the Service In variable, <code>partName</code>, to <code>EDIdata</code>. ■ Set the value of the Service In variable, <code>storageType</code>, to <code>database</code>.



Batching EDI Documents

■ Overview	310
■ Indicating How Many Batch EDI Documents to Create	310
■ How Documents are Combined to Create the Batch EDI Document	311
■ Extracting Transactions and Sorting the Batched Documents	311
■ Recombining the Transactions into the Batch EDI Document	320
■ Before You Can Batch EDI Documents	326
■ Controlling How the batchProcess Service Forms the Batch Document	326
■ Setting Up to Batch EDI Documents	329
■ Delivering the Batch EDI Document	332

Overview

Non-standard

Important! You *cannot* use the EDI batching feature if you are using non-standard processing. For more information about processing levels, see “[Using Standard or Non-Standard Processing](#)” on page 110.

The EDI batching feature uses webMethods Trading Networks (Trading Networks) scheduled delivery. That is, you set up a public scheduled delivery queue that will receive the EDI documents that you want to batch into a batch EDI document. To get the EDI documents into the queue, you define processing rules that use the *Deliver Document By* processing action to place EDI documents into the scheduled delivery queue.

When you define the scheduled delivery, you associate it with a service. To batch EDI documents, you use the `wm.b2b.editn.batch:batchProcess` service, which webMethods provides with the webMethods EDI Module (EDI Module). The `batchProcess` service acts on the EDI documents in the scheduled delivery queue to create a batch EDI document.

To learn more about the basics of batching EDI documents, see Chapter 3, “[Using the EDI Module with Trading Networks](#)”, in the *webMethods Trading Networks Concepts Guide*.

Indicating How Many Batch EDI Documents to Create

Use the `batchProcess` service input variable, `oneBatchQueue`, to indicate how many output batch EDI documents you want the `batchProcess` service to create. You can set the `oneBatchQueue` variable to one of the following:

- `SINGLEOUTPUT`, which indicates that you want all the EDI documents in the scheduled delivery queue to be combined into one batch EDI document that has multiple interchanges or TRADACOMS transmissions.
- `MULTIPLEOUTPUTS`, which indicates that you want the EDI documents in the queue to be combined into multiple output batch EDI documents, where each batch EDI document contains one interchange or TRADACOMS transmission.



Important! If you set `oneBatchQueue` to `NONE`, this signals that you want to use backward compatibility and have the `batchProcess` service combine the documents in the method available in version 6.0.1 of the EDI Module. For more information, see [Appendix B, “Using the 6.0.1 Version of the Batching Feature”](#).

How Documents are Combined to Create the Batch EDI Document

When you define a scheduled delivery queue for batching documents, you associate a schedule with the queue. When the schedule indicates, Trading Networks invokes the batchProcess service to act on all EDI documents in the queue to combine them into one or more batch EDI documents.

To create the batch EDI document(s), the batchProcess service:

- 1 Extracts the transactions or messages (or TRADACOMS Files) from the EDI documents in the queue and sorts them.
- 2 Recombines the transactions or messages (or TRADACOMS Files) into the batch EDI document(s).

Extracting Transactions and Sorting the Batched Documents

For each EDI document in the queue, the batchProcess service extracts the transactions in the EDI document and sorts them. The following describes how the batchProcess service processes the EDI documents in the queue when the documents have interchange headers or TRADACOMS transmission headers.



Note: For information about how the batchProcess service sorts EDI documents in the queue that do *not* have interchange or transmission headers, see ["When EDI Documents Have No Interchange or Transmission Header" on page 315](#).

For each transaction in an EDI document:

- 1 The batchProcess service extracts a transaction from the EDI document.
- 2 The batchProcess service uses the sender/receiver from the interchange or transmission header of the EDI document to locate the EDITPA variables to use. For more information about the EDITPA variables that the batchProcess service uses, see ["Controlling How the batchProcess Service Forms the Batch Document" on page 326](#).
- 3 The batchProcess service sorts the transactions. For more information, see ["Sorting Transactions in the Queued Documents" on page 312](#).
- 4 Each queued EDI document is associated with a Trading Networks delivery task. After all transactions for an EDI document have been extracted, the batchProcess service updates the task status. For more information, see ["Updating the Task Status and Publishing Documents in the Case of Failure" on page 319](#).

Sorting Transactions in the Queued Documents

The EDI Module sorts the transactions in EDI documents in the batch queue. This section describes how the EDI Module sorts the EDI documents when they have interchange or transmission headers. For information about how the EDI Module sorts documents when they do not have interchange or transmission headers, see ["When EDI Documents Have No Interchange or Transmission Header" on page 315](#).

Collection Areas (which represent an interchange in the final batch EDI document)

You can think of the sorting as if the batchProcess service was placing each transaction from an EDI document into a *collection area*. Each collection area is associated with the following items, which the batchProcess service obtains from the interchange or transmission header of the document:

Each collection area has the same:

- EDI standard and version
- Interchange or transmission sender/receiver
- Production mode (e.g., Production or Testing)

Each collection area becomes an interchange or transmission segment in the final batch EDI document.

Subcollections (which represent a group in the final batch EDI document)

Within a collection area are *subcollections* that are associated with the following items, which the batchProcess service obtains from the group header (or TRADACOMS batch header) of the document:

Each subcollection area has the same:

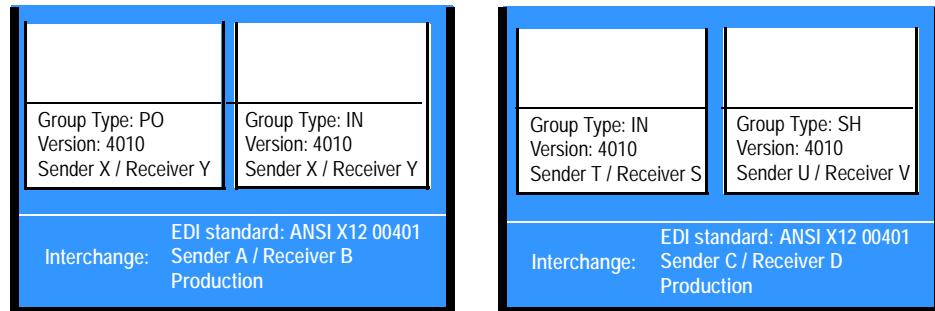
- Group type (or TRADACOMS batch type)
- Version of the EDI standard (e.g., 4010)
- Group sender/receiver (or TRADACOMS batch sender/receiver)

Each subcollection becomes a group segment (or TRADACOMS batch segment) within the interchange (or TRADACOMS transmission) in the final batch EDI document.



Note: The batchProcess service provides an input parameter, *createGroup*, which enables you to add a group or a TRADACOMS batch.

Example of collection and subcollection areas used for batching EDI documents



Delimiters Associated with Collection Areas

Each collection area is also associated with a set of *delimiters* that all transactions in that collection use.

Delimiters for the collection area

First, the batchProcess service uses the values from its *delimiters* input variable.

Second, if you do not specify a value for the *delimiters* input variable, the batchProcess service uses the sender and receiver from the interchange or transmission header to locate the EDITPA values, and uses the EDITPA *delimiters* variables.

Third, if there are no values for the EDITPA *delimiters* variables in the partner-specific and default EDITPAs, the batchProcess service uses its own defaults.

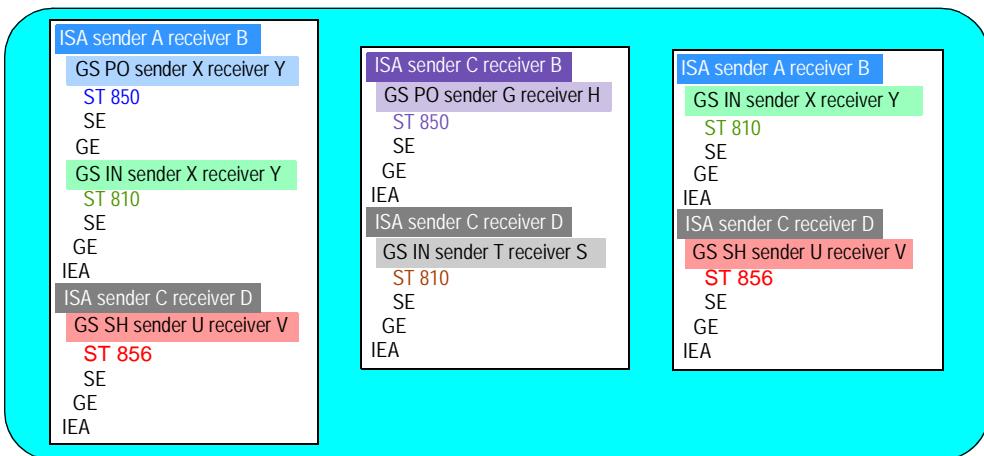
For more information, see “[Delimiters Used for the Batch EDI Document](#)” on page 318.

Using the interchange or transmission header from the original EDI document, the batchProcess service determines whether the transaction uses the same delimiters as those associated with the collection area. If the transaction uses the same delimiters, the batchProcess service places the document in the collection area. If the transaction uses different delimiters, the batchProcess service replaces the delimiters in the transaction with those used by the collection area before it places the transaction into the collection area.

Example of Sorting Documents

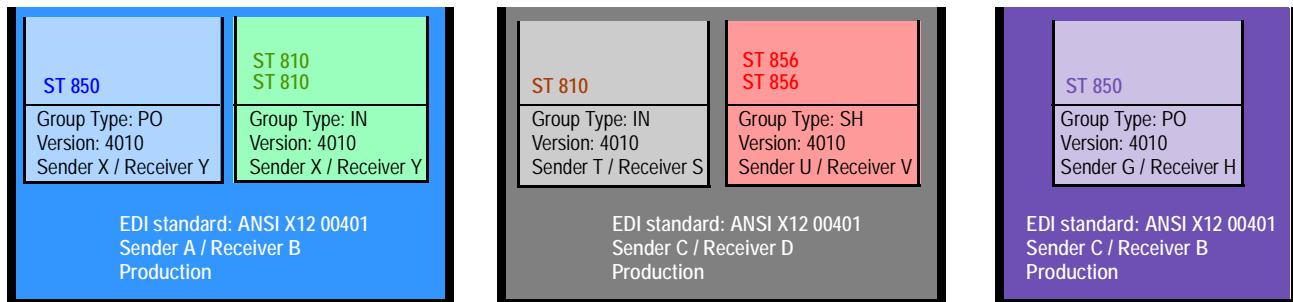
The following diagram shows three EDI documents in a scheduled delivery queue.

Example scheduled delivery queue holding EDI documents to batch



The following shows how the transactions from the above three documents are sorted into collection areas.

Transactions sorted into collection areas



When EDI Documents Have No Interchange or Transmission Header

You can place EDI documents that do not have interchange or TRADACOMS transmission headers in the queue. As described above in [“How Documents are Combined to Create the Batch EDI Document” on page 311](#), the batchProcess service typically uses information from the interchange or transmission headers when sorting the documents. The following describes how the batchProcess service behaves when the interchange or transmission headers are *not* available:

For each transaction in an EDI document that does not have an interchange or transmission header:

- 1 The batchProcess service extracts a transaction from the EDI document.
- 2 The batchProcess service uses its *senderIDQualifier*, *senderID*, *receiverIDQualifier*, and *receiverID* input variables to obtain the EDITPA to use. For more information about the EDITPA variables that the batchProcess service uses, see [“Controlling How the batchProcess Service Forms the Batch Document” on page 326](#).
- 3 The batchProcess service sorts the transactions. For more information, see [“Sorting Transactions When Queued Documents Have No Interchange Headers” on page 315](#).
- 4 Each queued EDI document is associated with a Trading Networks delivery task. After all transactions for an EDI document have been extracted, the batchProcess service updates the task status. For more information, see [“Updating the Task Status and Publishing Documents in the Case of Failure” on page 319](#).

Sorting Transactions When Queued Documents Have No Interchange Headers

This section describes how the EDI Module sorts the transactions of queued EDI documents when they have no interchange or transmission headers. For information about how the EDI Module sorts transactions when the EDI document have interchange or transmission headers, see [“Sorting Transactions in the Queued Documents” on page 312](#).

Default Collection Area (which will be an interchange in the final batch EDI document)

First the batchProcess service sorts the transactions into the default collection area. The default collection area will become an interchange or transmission in the final batch EDI document. The default collection area is associated with the following items, which the batchProcess service obtains from its input variables:

Default collection area to use when no interchange or transmission header	
EDI standard	Obtained from the input variable, <i>standard</i> , of the batchProcess service. In the final batch EDI document, the interchange or transmission that corresponds to the default collection area will use this EDI standard.
Version of the EDI standard	Obtained from the input variable, <i>version</i> , of the batchProcess service. In the final batch EDI document, the interchange or transmission that corresponds to the default collection area will use this version.
Interchange or transmission sender/receiver	Obtained from the following inputs of the batchProcess service: <i>senderIDQualifier</i> , <i>senderID</i> , <i>receiverIDQualifier</i> , and <i>receiverID</i> . In the final batch EDI document, the interchange or transmission that corresponds to the default collection area will use this sender and this receiver.
Production mode (e.g., Production or Testing)	Obtained from the input variable, <i>environment</i> , of the batchProcess service. In the final batch EDI document, the interchange or transmission that corresponds to the default collection area will use this production mode.

Subcollections When There are No Headers (which will be groups in the final batch EDI document)

The batchProcess further sorts the transaction into a subcollection. Each subcollection becomes a group (or a TRADACOMS batch) in the final batch EDI document. If the EDI document has group headers, the batchProcess service sorts the transaction into the subcollection as described in [“Subcollections \(which represent a group in the final batch EDI document\)” on page 312](#). If the EDI document does *not* have group headers, it uses the following.

Subcollection to use when no group header	
Group type	Determined based on the type of transaction; for example, if the transaction is an ANSI X12 850 (which is a purchase order), the batchProcess service uses the group type PO

Subcollection to use when no group header

Version of the EDI standard (e.g., 4010)	Obtained from the input variable, <i>version</i> , of the batchProcess service
Group sender/receiver	Obtained from the following inputs of the batchProcess service: <i>senderID</i> and <i>receiverID</i>

Delimiters Associated with the Default Collection Area

The default collection area is associated with a set of delimiters.

Delimiters for the default collection area

First, the batchProcess service uses the values from its *delimiters* input variable.

Second, if you do not specify a value for the *delimiters* input variable, the batchProcess service uses its *senderIDQualifier*, *senderID*, *receiverIDQualifier*, and *receiverID* input variables to locate the EDITPA values, and uses the EDITPA *delimiters* variables.

Third, if there are no values for the EDITPA *delimiters* variables in the partner-specific and default EDITPAs, the batchProcess service uses its own defaults.

For more information, see “[Delimiters Used for the Batch EDI Document](#)” on page 318.

The batchProcess service *cannot* replace delimiters before placing transactions in a collection area. When the interchange or transmission header *is* available, the batchProcess service can determine the delimiters used by the transaction from the queue, and as a result, can determine whether the transaction uses delimiters that are different from those used by the collection area. When the interchange or transmission header *is not* available, the batchProcess service is unable to determine the delimiters used by a transaction, and therefore cannot replace delimiters if they do not match.



Important! If you place EDI documents into a scheduled delivery queue for batching and the EDI documents do not have interchange or transmission headers, you *must* ensure that the delimiters that the EDI documents use match the delimiters of the default collection area. If there is a delimiter mismatch, the batchProcess service will create a batch EDI document that is not valid and that will not be able to be processed.

Delimiters Used for the Batch EDI Document

For ANSI X12 and UN/EDIFACT documents, each collection area is associated with a set of delimiters:

- Record delimiter
- Field delimiter
- Subfield delimiter
- Release character (used by UN/EDIFACT)



Note: For TRADACOMS documents, the EDI Module provides built-in support for the following TRADACOMS delimiters: segment terminator, data element separator, sub-element separator, and segment code separator. You cannot modify this list of delimiters. TRADACOMS users should skip this section and continue reading ["Updating the Task Status and Publishing Documents in the Case of Failure" on page 319](#).

The batchProcess service uses these delimiters when it combines the transactions in the collection area into the final batch EDI document(s). The batchProcess service determines the delimiters to use for a collection in the following order:

- 1 The batchProcess service *delimiters* input variable. If you specify the batchProcess service *delimiters* input variable, the batchProcess service uses the delimiters you specify for all collection areas. As a result, all interchanges in the output batch EDI document(s) will all have the same delimiters.
- 2 The *delimiters* EDITPA variables. If you leave the batchProcess service *delimiters* input variable null, the batchProcess service uses the EDITPA delimiters variable to determine the delimiters to use for a collection area. Leave the batchProcess service *delimiters* input variable if you want the interchange segments of the output batch EDI document(s) to use different delimiters.

How the batchProcess service locates the EDITPA varies based on whether the EDI document in the batch queue has interchange headers.

- If the EDI document has interchange headers, the batchProcess service uses the sender and receiver from the interchange header to locate the EDITPA.
 - If the EDI document does not have interchange headers, the batchProcess service uses its *senderIDQualifier*, *senderID*, *receiverIDQualifier*, and *receiverID* input variables to locate the EDITPA.
- 3 The EDI Module defaults for delimiters. If the batchProcess service cannot obtain delimiters from either its *delimiters* input variable or the EDITPA *delimiters* variables, it uses the following default for delimiters:

Type of delimiter	Default to use when cannot obtain delimiter value from other sources For ANSI X12	For UN/EDIFACT
Record	\n (newline character)	' (apostrophe)
Field	*	+(plus sign)
Subfield	:	:
Release character	(not used by ANSI X12)	? (question mark)

Updating the Task Status and Publishing Documents in the Case of Failure

Each EDI document in the queue of documents that are being batched is associated with a Trading Networks delivery task. You can view tasks from the **Tasks** screen of the Trading Networks Console. The `batchProcess` service updates the status of the delivery tasks to one of the following as it sorts the transactions.

Task Status	Description
success	If the <code>batchProcess</code> service successfully extracts and sorts the transactions from an EDI document, the <code>batchProcess</code> service sets the delivery task status to success . The transactions from the document will be included in a final batch EDI document.
fail	If the <code>batchProcess</code> service is unable to extract and sort the transactions from an EDI document, the <code>batchProcess</code> service sets the delivery task status to fail . This can occur, for example, if the document in the queue is not a properly formed EDI document. The transactions from the document will <i>not</i> be included in the final batch EDI document. If a task fails, you can have the <code>batchProcess</code> service publish an IS document to notify you of the failure. The <code>batchProcess</code> service publishes the document if the <code>publishBatchFailEvent</code> EDITPA variable is set to <code>true</code> . For more information, ANSI X12 and UN/EDIFACT users should see “ “publishBatchFailEvent EDITPA Variable” on page 140 and TRADACOMS users should see “ “publishBatchFailEvent EDITPA Variable” on page 159 .

Task Status	Description
	The format of the IS document is defined by the <code>wm.b2b.editn.rec:batchFailDocument</code> IS document type. To view the format of this IS document, see the description of the <code>wm.b2b.editn.rec:batchFailDocument</code> IS document type in the <i>webMethods EDI Module Built-In Services Reference</i> .
	To handle the failure, you can use the webMethods Developer to create Integration Server triggers that subscribe to the published documents. For information on creating an Integration Server trigger, see the <i>Publish-Subscribe Developer's Guide</i> .

Recombining the Transactions into the Batch EDI Document

When recombining the transactions to create the final batch EDI document(s), the `batchProcess`:

- Adds interchange or TRADACOMS transmission headers with the following information:
 - Sender associated with the collection area
 - Receiver associated with the collection area
 - EDI standard associated with the collection area
 - Production mode associated with the collection area
 - Delimiters associated with the collection area
 - Other interchange values (e.g., ISA01, ISA02, UNB01, UNB07, etc.) that the `batchProcess` service obtains from the EDITPA *ICheaderInfo* variables. If the value of an *ICheaderInfo* variable is not specified, the `batchProcess` service uses the value from the interchange header in the original EDI document. Information about the *ICheaderInfo* variables starts at "["ICheaderInfo/ISA/ISA01 EDITPA Variable" on page 142.](#)
- Adds group or TRADACOMS batch headers with the following information:
 - Sender associated with the subcollection
 - Receiver associated with the subcollection
 - Group or batch type associated with the subcollection
 - EDI version associated with the subcollection
 - GS07 value that the `batchProcess` service obtains from the EDITPA *ICheaderInfo/GS/GS07* variable. If the value of an *ICheaderInfo/GS/GS07* variable is not specified, the `batchProcess` service uses the value from the group header in the

original EDI document. Information about the *ICheaderInfo* variables starts at “[ICheaderInfo/GS/GS07 EDITPA Variable](#)” on page 143.

- **Performs FA reconciliation.** The batchProcess service tracks groups in the batch EDI document(s) for FA reconciliation. That is, for each group in the final batch EDI document(s) and for each functional acknowledgement (i.e., ANSI X12 997 or UN/EDIFACT CONTRL), the batchProcess service updates the EDITRACKING table, honoring the *FAReconciliation* EDITPA variable. For more information about the settings for the *FAReconciliation* EDITPA variable, see “[FAReconciliation EDITPA Variable](#)” on page 136. For more information about FA reconciliation, see Chapter 21, “[Reconciling Functional Acknowledgments](#)”.



Note: Functional acknowledgments (FAs) are not applicable to TRADACOMS.

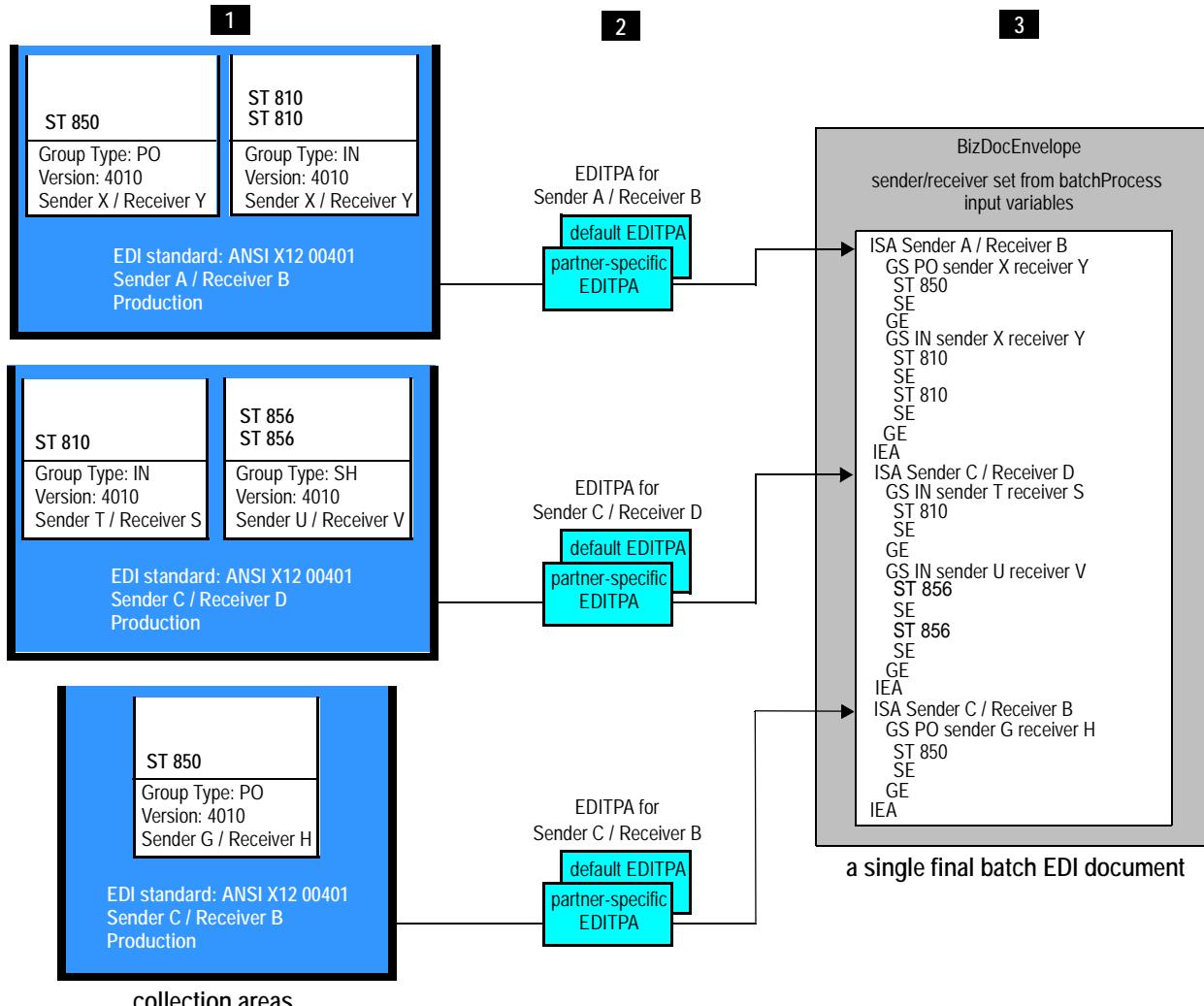
- **Creates one or more output batch EDI documents.** The batchProcess service determines how many batch EDI documents to create based on the *oneBatchQueue* input variable. That is, if *oneBatchQueue* is **SINGLEOUTPUT**, the batchProcess service creates a single output batch EDI document. If *oneBatchQueue* is **MULTIPLEOUTPUTS**, the batchProcess service creates multiple output batch EDI documents, each containing a single interchange or transmission. For more information, see “[Indicating How Many Batch EDI Documents to Create](#)” on page 310.

After creating the final output batch EDI document(s), the batchProcess service creates a BizDocEnvelope for each batch EDI document that it created and sends each BizDocEnvelope to Trading Networks processing rules. You create a processing rule that specifies the action to deliver the batch EDI document. For more information, see “[Delivering the Batch EDI Document](#)” on page 332.

Creating the Batch EDI Document when using SINGLEOUTPUT

The following diagram illustrates how the batchProcess service combines the transactions from the collection areas when the *oneBatchQueue* input variable is **SINGLEOUTPUT**. The batchProcess service combines all transactions from all collection areas into a single batch EDI document. For more information, see the table following the diagram.

Transactions sorted into collection areas and the final EDI document when using **SINGLEOUTPUT**



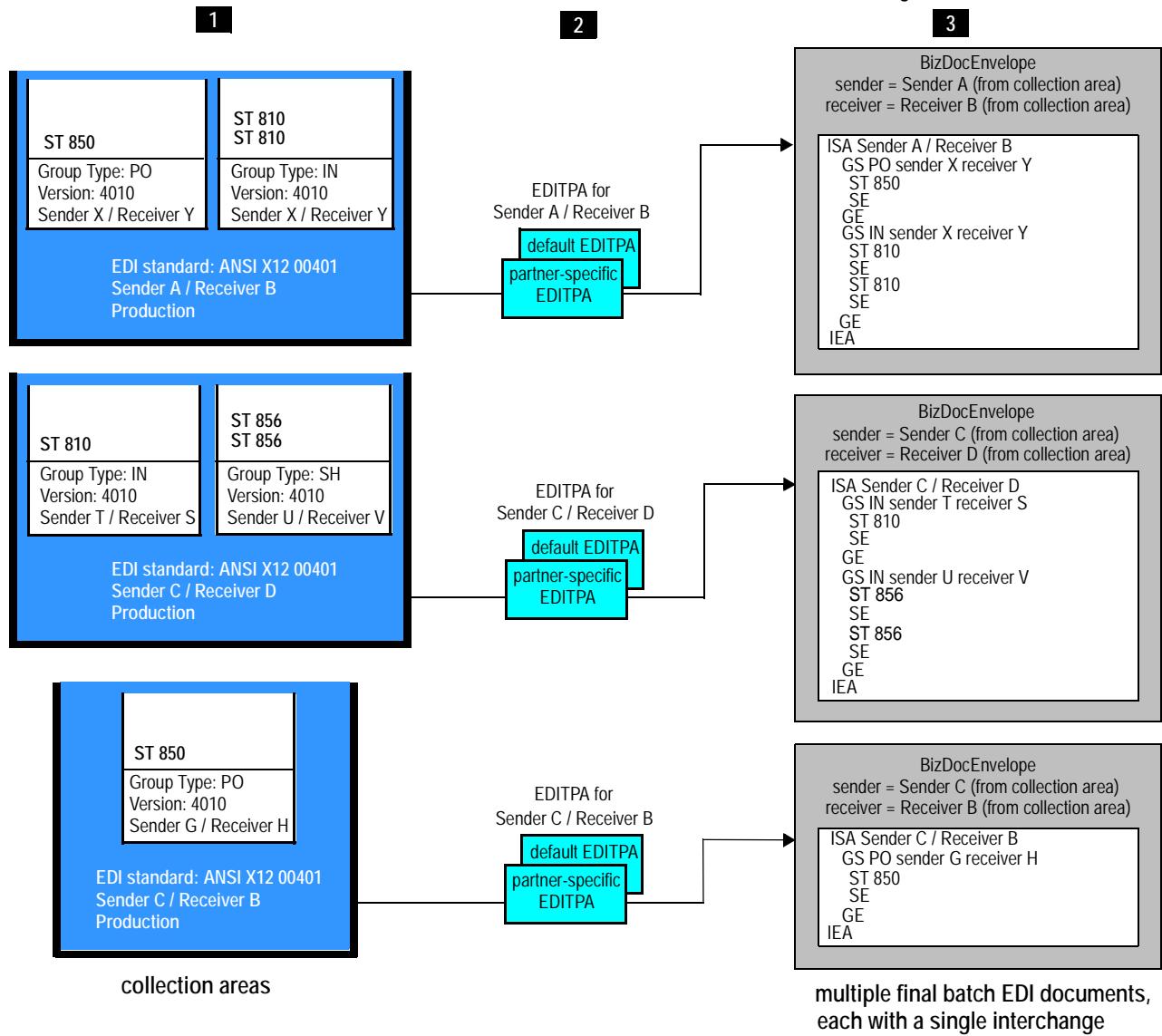
Step	Description
1	For each collection area, the batchProcess service combines the transactions in the collection area. Each collection becomes an interchange in the final EDI batch document, with each subcollection a group within the interchange.
2	<p>The batchProcess service obtains the EDITPA information for the sender/receiver that are associated with the collection area. It uses information associated with the collection area, the EDITPA information, and information associated with the subcollections to create the interchange and group headers:</p> <ul style="list-style-type: none"> ■ Uses the following information associated with the collection area for the interchange headers: sender/receiver, EDI standard, production mode, and delimiters. ■ Obtains fields for the interchange header from the EDITPA <i>ICHeaderInfo</i> variables. If an EDITPA <i>ICHeaderInfo</i> variable does not have a value, the batchProcess service uses the corresponding value from the interchange header of the original document. For more information, see descriptions of the <i>ICheaderInfo</i> variables that start with ""ICheaderInfo/ISA/ISA01 EDITPA Variable" on page 142. ■ Uses the following information associated with a subcollection for a group header: sender/receiver, group type, version of the EDI standard ■ Obtains the value of the GS07 field (for ANSI X12) from the EDITPA <i>ICHeaderInfo/GS/GS07</i> variable. If this variable does not have a value, it uses the GS07 from the original EDI document. For more information, see ""ICheaderInfo/GS/GS07 EDITPA Variable" on page 143.
3	After recombining the information for all collection areas, the batchProcess service creates a BizDocEnvelope for the final batch EDI document. The batchProcess service sets the sender and receiver for the BizDocEnvelope to the sender/receiver identified in its <i>senderIDQualifier</i> , <i>senderID</i> , <i>receiverIDQualifier</i> , and <i>receiverID</i> input variables. It then sends the BizDocEnvelope to Trading Networks processing rules for routing.

Note: If you use Sender or Receiver criteria in the processing rule to route the final batch EDI document, identify the final batch EDI document using the sender/receiver of the BizDocEnvelope.

Creating the Batch EDI Document when using MULTIPLEOUTPUTS

The following diagram illustrates how the batchProcess service combines the transactions from the collection areas when the *oneBatchQueue* input variable is **MULTIPLEOUTPUTS**. The batchProcess service combines the transactions from each collection areas into a single batch EDI document that contains a single interchange. The result is multiple output batch EDI documents.

Transactions sorted into collection areas and the final EDI documents when using **MULTIPLEOUTPUTS**



Step	Description
1	For each collection area, the batchProcess service combines the transactions in the collection area. Each collection becomes a single batch EDI document that contains a single interchange with each subcollection a group within the interchange.
2	The same as when <i>oneBatchQueue</i> is SINGLEOUTPUT, when <i>oneBatchQueue</i> is MULTIPLEOUTPUTS, the batchProcess service obtains the EDITPA information for the sender/receiver that are associated with the collection area. It uses information associated with the collection area, the EDITPA information, and information associated with the subcollections to create the interchange and group headers. For more information, see step 2 in “Creating the Batch EDI Document when using SINGLEOUTPUT” on page 322 .
3	For each collection area, after recombining the transactions in the collection area, the batchProcess service creates a BizDocEnvelope for the final batch EDI document. The batchProcess service sets the sender and receiver for the BizDocEnvelope to the sender/receiver associated with the collection area. It then sends the BizDocEnvelope to Trading Networks processing rules for routing.

Note: If you use Sender or Receiver criteria in the processing rule to route the final batch EDI document, identify the final batch EDI document using the sender/receiver of the BizDocEnvelope.

Before You Can Batch EDI Documents

Before the EDI Module can batch EDI documents, you must do the following:

- Install the TN document types for the EDI documents that you will place in the scheduled delivery queues. For instructions on how to install TN document type for EDI documents, see “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104.
- Install the TN document types for the final batch EDI document. You need to install the TN document types for the envelope (e.g., the TN document type X12 Envelope) and the group or TRADACOMS batch (e.g., the TN document type X12 Group). For instructions on how to install TN document type for EDI documents, see “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104.
- Define the default and optionally partner-specific EDITPAs for the interchange or transmission sender/receiver pairs. For more information, ANSI X12 and UN/EDIFACT users should see “[Defining EDI Trading Partner Agreements](#)” on page 116 and TRADACOMS users should see “[Defining EDI Trading Partner Agreements](#)” on page 150. For more information about the variables in the EDITPAs that affect batching EDI documents, see “[EDITPA Variables that the batchProcess Service Uses](#)” on page 327.

Controlling How the batchProcess Service Forms the Batch Document

You control how the batchProcess service combines the EDI documents in a schedule delivery queue by:

- Using the input variables of the batchProcess service

You define the input variables to the batchProcess service when you define the scheduled delivery queues in Trading Networks. Trading Networks invokes the service to act on documents that are placed in the scheduled delivery queue using the input variables that you specify. For more information, see “[Defining the Scheduled Delivery Queues](#)” on page 329.

- Defining EDITPA variables that are used during batch processing.

For more information, see “[EDITPA Variables that the batchProcess Service Uses](#)” on page 327. For more information about EDITPAs in general, ANSI X12 and UN/EDIFACT users should see “[Defining EDI Trading Partner Agreements](#)” on page 116 and TRADACOMS users should see “[Defining EDI Trading Partner Agreements](#)” on page 150.

EDITPA Variables that the batchProcess Service Uses

The batchProcess service uses EDITPA variable when sorting transactions in the EDI documents in the scheduled delivery queue and when recombining the transactions into the final batch EDI document(s).

EDITPA Variables used When Sorting Transaction

The following table lists the EDITPA variables that the batchProcess service uses when sorting the transactions.

EDITPA variables that the batchProcess service uses	Description
<i>publishBatchFailEvent</i>	The batchProcess service uses the <i>publishBatchFailEvent</i> variable to determine whether you want it to publish an IS document to notify you when it is unable to sort the transactions from a queued EDI document into a collection area, and therefore will not be able to include those transactions in the final batch EDI document. For more information, see “publishBatchFailEvent EDITPA Variable” on page 140 and “Updating the Task Status and Publishing Documents in the Case of Failure” on page 319 .
<i>delimiters</i>	ANSI X12 and UN/EDIFACT only. The batchProcess service uses the EDITPA <i>delimiters</i> variables when the batchProcess service <i>delimiters</i> input variable is null. In this situation, the batchProcess service uses the EDITPA delimiters for collection areas and therefore in the interchanges (or TRADACOMS transmissions) in the final batch EDI document(s). For more information, see “Delimiters Used for the Batch EDI Document” on page 318 .

EDITPA Variables used When Recombining Transactions

The following table lists the EDITPA variables that the batchProcess service uses when recombining the transactions.

 Note: These variables are applicable only to ANSI X12 and UN/EDIFACT documents.	
EDITPA variables that the batchProcess service uses	Description
<i>UNAmode</i>	The batchProcess service uses the <i>UNAmode</i> variable to determine whether to create a UNA segment prior to the interchange (or TRADACOMS transmission) in the final batch EDI document. The <i>UNAmode</i> variable is specific to UN/EDIFACT. For more information, see “UNAmode EDITPA Variable” on page 138 .
<i>ICheaderInfo</i>	The batchProcess service uses the <i>ICheaderInfo</i> variables to construct the interchange headers (or TRADACOMS transmission headers) in the final batch EDI document. For an X12 group header, the batchProcess service also uses the <i>ICHeaderInfo/GS/GS07</i> variable for the group header. Descriptions of the <i>ICheaderInfo</i> variables start with “ICheaderInfo/ISA/ISA01 EDITPA Variable” on page 142 .

Note: If the *ICheaderInfo* variables do not have a value, the batchProcess service uses the values from the headers in the original document.

Setting Up to Batch EDI Documents

To batch EDI documents, you need to:

- Determine the number of scheduled delivery queues you need to define. See “[Determining the Number of Queues You Need to Define](#)” below.
- Define the scheduled delivery queues for batching EDI documents. See “[Defining the Scheduled Delivery Queues](#)” on page 329.
- Define the processing rules to deliver documents to the scheduled delivery queues. See “[Defining Processing Rules to Batch EDI Documents](#)” on page 331.

Determining the Number of Queues You Need to Define

The number of scheduled delivery queues you need depends on the number of destinations to which you want to send batch EDI documents. Create one scheduled delivery queue for each destination. For example, you might create one queue to batch data destined for the GXS VAN, a second queue for batch data to go to the ICC.NET VAN, and a third queue for batch data directed to a specific company.

Defining the Scheduled Delivery Queues

You define the scheduled delivery queues using the Trading Networks Console.

To define a scheduled delivery queue for batching EDI documents

To create a scheduled delivery queue, define a public queue in Trading Networks. For steps to create a public queue, see the chapter about queues in the *webMethods Trading Networks User’s Guide*.

The following table describes information that you need to supply when defining the queue:

Public queue setting	Description
Queue Name	The name you want to give the scheduled delivery queue. For example, if you are defining a queue for EDI documents for which the receiver is CYG Company, you might set the Queue Name to "CYG Queue."
Delivery Service	The delivery service you want to associate with the queue. Select EDI Batch. Note: When you install the EDI Module, the EDI Module automatically registers the <code>wm.b2b.edtn:batchProcess</code> service with Trading Networks as a scheduled delivery service and assigns it the name EDI Batch. You assign the <code>batchProcess</code> service to a scheduled delivery queue by selecting EDI Batch for Delivery Service.
Set Inputs	The inputs for the <code>batchProcess</code> service. Click the Set Inputs button to open a dialog box that allows you to set the input variables for the <code>batchProcess</code> service. When Trading Networks invokes the <code>batchProcess</code> service, Trading Networks passes the inputs that you specify. For more information, see the description of the <code>wm.b2b.edtn:batch:batchProcess</code> service in the <i>webMethods EDI Module Built-In Services Reference</i> .
Schedule	When you want Trading Networks to invoke the <code>batchProcess</code> service to act on the EDI documents in the queue you are defining. When defining a schedule, consider how often your trading partners, VAN, etc., should receive the batch EDI documents.



Note: webMethods recommends that you do not set up private queues for EDI batching because of the limitations that private queues pose. EDI batching with private queues results in EDI documents always going to a specific receiver's queue because you define private queues in a trading partner profile.

Defining Processing Rules to Batch EDI Documents

You define processing rules that instruct Trading Networks to place an EDI document into a scheduled delivery queue that is being used for batching EDI documents.

To define a processing rule to batch EDI documents

To define a processing rule in Trading Networks, use the Trading Networks Console. For steps to create a processing rule, see the chapter about processing rules in the *webMethods Trading Networks User's Guide*.

The following table lists details about how to define the processing rules.

On this Processing Rules tab...	Specify...
Criteria	The standard criteria you want to use. Specify criteria that describes the EDI documents you want to place in a scheduled delivery queue.
Extended Criteria	You will not need to use extended criteria.
Action	<p>What you want Trading Networks to do with the EDI document it is processing.</p> <p>To place the document into a scheduled delivery queue that is used for batching EDI documents, do the following:</p> <ul style="list-style-type: none">■ Select Deliver Document By processing action.■ For the Deliver Document By processing action, select Scheduled Delivery.■ Select the appropriate scheduled delivery queue that you associated with the EDI Batch delivery service from the list of registered queues.

Delivering the Batch EDI Document

After the batchProcess service creates the final batch EDI document, it sends the batch EDI document to Trading Networks processing rules. You create another Trading Networks processing rule to deliver the final batch EDI document.

When defining the processing rule to deliver a batch EDI document, use the EDI Batch custom attribute. For a batch EDI document, the EDI Batch custom attribute is set to Interchange. You can use the EDI Batch custom attribute in the extended criteria of the processing rule, so the processing rule is only used for a batch EDI document.

To deliver a batch EDI document, the processing rule can use one of the following actions:

- Execute a Service processing action to invoke a service that you create to deliver the final batch EDI document.
- Deliver Document By processing action to send the document to a VAN

To define a processing rule to deliver a final batch EDI document

To define a processing rule in Trading Networks, use the Trading Networks Console. For steps to create a processing rule, see the chapter about processing rules in the *webMethods Trading Networks User's Guide*.

The following table lists details about how to define the processing rules.

On this Processing Rules tab...	Specify...
Criteria	<p>The standard criteria you want to use to select the processing rule. Specify criteria that describes the final batch EDI document that the batchProcess service creates.</p> <p>Note: If you use the Sender and/or Receiver criteria, the sender/receiver you need to specify is that of the BizDocEnvelope for the final batch EDI document. The sender/receiver are different based on whether the batchProcess service input variable <i>oneBatchQueue</i> is SINGLEOUTPUT or MULTIPLEOUTPUTS. For more information, see step 3 in “Creating the Batch EDI Document when using SINGLEOUTPUT” on page 322 and “Creating the Batch EDI Document when using MULTIPLEOUTPUTS” on page 324.</p>

On this Processing Rules tab...	Specify...	
Extended Criteria	The custom attributes that you want to use as criteria to select the processing rule. The following shows the extended criterion you should add:	
	Attribute	Operator
	EDI Batch	Equal
Action	<p>How you want Trading Networks to deliver the final batch EDI document.</p> <ul style="list-style-type: none">■ Select Execute a Service processing action to invoke a service that you create to deliver the final batch EDI document.■ Select Deliver Document By processing action to send the document to a VAN. For more information, see “Setting Up to Deliver Documents to VANs” on page 337 in Chapter 18, “Retrieving and Delivering EDI Documents from and to VANs”.	

Retrieving and Delivering EDI Documents from and to VANs

■ Overview	336
■ Setting Up to Retrieve Documents from a VAN	336
■ Setting Up to Deliver Documents to VANs	337
■ Handling Conventions Required by Specific VANs	340
■ Customizing the Built-in Services to Connect To Another VAN	341

Overview

When using the webMethods EDI Module (EDI Module) with webMethods Trading Networks (Trading Networks), you can connect to Value Added Networks (VANs) to retrieve inbound EDI documents from VANs and to deliver outbound EDI documents to VANs.

webMethods has tested and certified the EDI Module to connect with the GXS, ICC.NET, and MCI VANs. If you need to connect to another VAN, you might need to customize the services that the EDI Module provides to suit the specific VAN connectivity.



Note: The GXS VAN was formerly known as the GEIS VAN.

To learn more about VANS and an overview of how the EDI Module works with VANS, see Chapter 3, "Using the EDI Module with Trading Networks" in the *webMethods Trading Networks Concepts Guide*.

Setting Up to Retrieve Documents from a VAN

To retrieve inbound EDI documents from a VAN, invoke the `VAN.VANConnectivity:getFromVAN` service. You can schedule the `getFromVAN` service to run at the times you want to retrieve inbound EDI documents from the VAN. To do so, set up a user task for the service. For information about how to use the Server Administrator to set up a user task, see the *webMethods Integration Server Administrator's Guide*.

The input variables that you supply for the `getFromVAN` service identify the VAN to which you want to connect to retrieve EDI documents. The following table describes other input variables to the `getFromVAN` service that you can use to specify optional actions that you might want the `getFromVAN` service to perform in addition to retrieving the inbound EDI documents. To learn about *all* the input variables to the `getFromVAN` service, see the *webMethods EDI Module Built-In Services Reference*.

Use this input variable to the <code>getFromVAN</code> service	To have the <code>getFromVAN</code> service perform this optional action...
<code>saveInboundToTN</code>	To have the <code>getFromVAN</code> service submit the retrieved inbound EDI documents to Trading Networks for processing. If you set the <code>saveInboundToTN</code> variable to yes, the retrieved inbound EDI documents are processed through Trading Networks like any other inbound EDI documents.
<code>getReport</code>	To have the <code>getFromVAN</code> service get VAN-generated reports after receiving the inbound EDI documents.

Use this input variable to the getFromVAN service	To have the getFromVAN service perform this optional action...
---	--

PGPEnable

To have the getFromVAN service verify and decrypt the retrieved inbound EDI documents.

Note: PGP-encryption support is deprecated; it will not be available in a future release of the EDI Module.

PGP-encryption is supported *only* as part of the VAN connectivity to ICC.net. PGP-encryption is *not* generically supported across webMethods components.

 Note: In addition to invoking the getFromVAN service or in place of using the getFromVAN service, you can have the VAN.VANConnectivity.putToVAN service retrieve waiting inbound EDI documents when it delivers outbound EDI documents to the VAN. For more information about the putToVAN service, see “[Setting Up to Deliver Documents to VANs](#)” below. (Note that because the putToVAN service is registered in Trading Networks as a scheduled delivery service and assigned the name VANFTP, this chapter also refers to the putToVAN service as the VANFTP service.)

Setting Up to Deliver Documents to VANs

To send outbound EDI documents to a VAN, you define:

- Scheduled delivery queue in Trading Networks. Use the scheduled delivery queue to hold outbound EDI documents that are to be sent to a VAN.
- Processing rule to place outbound EDI documents into the scheduled delivery queue.

Defining the Scheduled Delivery Queue

To define a scheduled delivery queue for delivering EDI documents to a VAN, you define a Trading Networks public queue using the Trading Networks Console. You define one queue for each VAN to which you want to deliver documents. For example, if you want to connect to GXS, ICC.NET, and MCI, you must define a public queue for each VAN. If you

want, you can set up more than one queue for a VAN. For example, you might define multiple queues for a single VAN if you have more than one account for the VAN.

When you define the queue, you:

- **Assign the queue a name.** You can give the queue any name you want. When you define a processing rule to place outbound EDI documents in the queue, you will select the queue by the name you assign it.
- **Associate the queue with a scheduled delivery service.** You associate the queue with the VANFTP scheduled delivery service. The VANFTP scheduled delivery service is the VAN.VANConnectivity:putToVAN service. The EDI Module registers the putToVAN service as a Trading Networks scheduled delivery service and assigns it the name VANFTP in Trading Networks. The VANFTP service uses FTP to deliver documents to a specific VAN.
- **Assign the input values for the VANFTP service.** Set the values for the VANFTP service to indicate the VAN to which you want to connect and to set optional actions you want the VANFTP service to perform. For example, you can indicate that you want to retrieve VAN-generated reports.
- **Associate the queue with a schedule.** You specify the times that you want to deliver the outbound EDI documents to the VAN. At the times specified by the schedule, Trading Networks invokes the VANFTP scheduled delivery service to send the outbound EDI documents that are in the queue to the VAN.

To define a scheduled delivery service for delivering EDI documents to VANs

To create a scheduled delivery queue, define a public queue in Trading Networks. For steps to create a public queue, see the chapter about queues in the *webMethods Trading Networks User's Guide*.

The following table describe information that you need to supply when defining the queue.

Public queue setting	Description
Queue Name	The name you want to give the scheduled delivery queue. For example, if you are defining a queue to hold EDI documents to be sent to the ICC.NET VAN, you might name the queue "ICCNET Queue".
Delivery Service	The delivery service you want to associate with the queue. Select VANFTP.

Public queue setting	Description
Set Inputs	<p>The inputs for the VANFTP service. Click the Set Inputs button to open a dialog box that allows you to set the input variables for the VANFTP service. When Trading Networks invokes the VANFTP service, Trading Networks passes the inputs that you specify.</p>
	<p>The input variables that you specify for the VANFTP service identify the VAN to which you want to connect to deliver the EDI documents in the queue. The following describes other input variables to the VANFTP service that you can use to specify optional actions that you might want the VANFTP service to perform in addition to delivering the outbound EDI documents. To learn about <i>all</i> the input variables to the VANFTP service, see the description of the VAN.VANConnectivity:putToVAN service in the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
<u>Use this input variable to the VANFTP service</u>	<p>To have the VANFTP service perform this optional action...</p>
<i>PGPEnable</i>	<p>To have the VANFTP service sign and PGP encrypt the outbound EDI documents.</p>
	<p>Note: PGP-encryption support is deprecated; it will not be available in a future release of the EDI Module.</p>
<i>getReport</i>	<p>PGP-encryption is supported <i>only</i> as part of the VAN connectivity to ICC.net. PGP-encryption is <i>not</i> generically supported across webMethods components.</p>
	<p>To have the VANFTP service get VAN-generated reports after delivering the outbound EDI documents.</p>
<i>getInbound</i>	<p>To have the VANFTP service retrieve inbound EDI documents from the VAN after it delivers the outbound EDI documents from the queue. The VANFTP service always submits the inbound EDI documents that it retrieves to Trading Networks for processing.</p>
Schedule	<p>When you want Trading Networks to invoke the VANFTP service to deliver the EDI documents in the queue you are defining.</p>



Note: webMethods recommends that you do *not* set private queues for delivering EDI documents to VANs. You can only use a private queue for a specific trading partner while public queues can be used by multiple trading partners.

Defining a Processing Rule to Place Documents in the Queue

To deliver an outbound EDI document to a VAN, you define a processing rules that instructs Trading Networks to place the outbound EDI document into a scheduled delivery queue that is associated with the VANFTP service.

To define a processing rule for delivering EDI documents to VANs

To define a processing rule in Trading Networks, use the Trading Networks Console. For steps to create a processing rule, see the chapter about processing rules in the *webMethods Trading Networks User's Guide*.

When specifying actions on the Action tab:

- Select the Deliver Document By processing action.
- Select Schedule Delivery.
- Select the appropriate scheduled delivery queue that you associated with the VANFTP service from the list of registered queues.

Handling Conventions Required by Specific VANs

Each VAN has its own conventions and idiosyncrasies that you should be aware of when setting input variables to the VAN:VANConnectivity:getFromVAN service to send EDI documents to a VAN -AND- when setting the input variables to the VAN:VANConnectivity:putToVAN (VANFTP) service to retrieve EDI documents from a VAN.

For example:

- Some VANs require you to provide account names when connecting, while others do not.
- Some VANs have delineated inbound and outbound document boxes, while others do not.
- Some VANs support the retrieval of documents based on a file name pattern, while others do not.

- Some VANs support PGP encryption, while others do not.



Important! If you specify to PGP encrypt a document bound for a VAN that does not support PGP encryption, the document will be unrecognizable to the receiver (VAN).

- The availability and variety of reports, as well as the ways in which you access them, differs from VAN to VAN. For example, as of the 6.0 release, ICC.NET offers web-based reporting only.

Customizing the Built-in Services to Connect To Another VAN

If you need to connect to a VAN other than GXS, ICC.NET, or MCI, you might need to create customized services to interact with the VAN. First attempt to connect to the VAN as described previously in this chapter. If you encounter problems executing the VAN.VANConnectivity:getFromVAN service and/or the VAN.VANConnectivity:putToVAN (VANFTP) service, create customized services.

To create customized services to connect to other VANs

- 1 Create your customized service using the VAN.VANConnectivity:getFromVAN and the VAN:VANConnectivity:putToVAN (VANFTP) services as templates.

Be sure to give the customized services a different names. Your customized services should use the FTP connectivity services that webMethods provides in the VAN.VANConnectivity folder of the WmEDIforTN package.
- 2 If you created a customized service to replace the VAN.VANConnectivity:getFromVAN service, you can create a scheduled user task to have the service executed at the times you want to retrieve the EDI documents from the VAN. For information about how to use the Server Administrator to set up a user task, see the *webMethods Integration Server Administrator's Guide*.
- 3 If you created a customized service to replace the VAN.VANConnectivity:putToVAN (VANFTP) service:
 - a Register your customized service as a Trading Networks scheduled delivery service. To do so, invoke the `wm.tn.delivery:registerService` service. For information about this service, see the *webMethods Trading Networks Built-in Services Reference*.
 - b From the Trading Networks Console, update the definition for the scheduled delivery queue that you created for sending EDI documents to VANs by replacing the VANFTP delivery service with the registered service name of your customized service.

Handling Large Documents When Using Trading Networks

- Configuring Large Document Handling when Using webMethods Trading Networks 344
- Converting Documents to IData Objects Iteratively 344

Configuring Large Document Handling when Using webMethods Trading Networks

When using the webMethods EDI Module (EDI Module) with webMethods Trading Networks (Trading Networks), the EDI Module uses the Trading Networks large document settings. For information about configuring Trading Networks for large document handling, see the chapter about configuring Trading Networks in the *webMethods Trading Networks User's Guide*.

An exception is when you invoke services provided with the WmEDI package that perform different processing for large document handling, that is, the following services:

- For ANSI X12 and UN/EDIFACT documents:
 - `wm.b2b.edi:envelopeProcess`
 - `wm.b2b.edi:convertToValues`
 - `wm.b2b.edi.util:generateFA`
- For TRADACOMS documents:
 - `wm.b2b.edi.tradacoms:convertToValues`

You might invoke these services from a Trading Networks processing rule. These services always use the EDI Module large document settings as described in [“Configuring Large Document Handling” on page 94](#) in Chapter 7, “Handling Large Documents”.

To simplify large document handling for all of your documents, you might choose to assign identical values to some or all of the EDI Module and Trading Networks large document settings.

Converting Documents to IData Objects Iteratively

When you are converting a large document to an IData object using the `convertToValues` service, you should convert the document iteratively. For more information, see:

- [“Converting Documents to IData Objects Iteratively” on page 98](#) in Chapter 7, “Handling Large Documents”
- [“Processing the Document Iteratively Segment by Segment” on page 47](#) in Chapter 3, “Receiving and Processing Inbound EDI Documents”

Including Documents in a Business Process

■ Overview	346
■ Designing the Process Model	346
■ How to Assign Conversation IDs to EDI Documents	352
■ Sample Business Process	357
■ Monitoring Your Business Processes	357

Overview

Using webMethods Modeler, you design a process model that describes your *business process* (also called a *conversation*) that involves EDI documents. For an EDI document to be involved in a business process, it must have a conversation ID. A conversation ID is a unique identifier that is unique to an instance of a business process. All documents involved in the same instance of a business process contain the same conversation ID.

When webMethods EDI Module (EDI Module) receives an EDI document, it splits the original inbound EDI document based on the EDITPA *splitOption* variable. The EDI Module can split the original EDI document into Interchange, Group, and/or Transaction documents. The Interchange, Group, and/or Transaction documents are passed to a business process when they contain a conversation ID.

The EDI Module always assigns the Interchange and Group documents conversation IDs. As a result, because the Interchange and Group documents have a conversation ID, they are always passed to the process run time, which is the facility of the Integration Server that executes and manages business processes in the Integration Server.

The EDI Module only assigns a conversation ID to a Transaction document if you have created an *instance ID query* that identifies the value to set for the conversation ID. You define instance ID query for specific transaction sets (e.g., X12 4010 850) using the WmEDIforTN home page.



Note: You cannot process TRADACOMS documents in a business process.

To learn more about:

- For an overview of processing EDI documents in business processes, including an illustration of how EDI documents are passed to a business process, see Chapter 4, "EDI Documents in Business Processes" in the *webMethods EDI Module Concepts Guide*.
- Process models and using webMethods Modeler, see the *webMethods Modeler User's Guide*.

Designing the Process Model

You design a process model that involves EDI documents in the same way that you would form any process model. For more information about webMethods Modeler, see the *webMethods Modeler User's Guide*.

Sample process models that involve EDI documents are provided in the Knowledge Base of the webMethods Advantage Web site at <http://advantage.webmethods.com>. You can import the sample models into webMethods Modeler to view them.

Business processes that involve EDI documents typically only involve the Transaction document and the level of document for the functional acknowledgment (that is, a Group document for ANSI X12 and an Interchange document for UN/EDIFACT).

When you define the process model, the first step is always a step that waits for a document. You assign an input subscription to that step to specify the type of document for which to wait. You identify the document by its TN document type. You can set criteria for the start document so that the process begins only if the document contains specific information. For example, you can set criteria so that the process starts only if the sender of the start document is a specific partner. You set the criteria by assigning a subscribe filter to the start step.

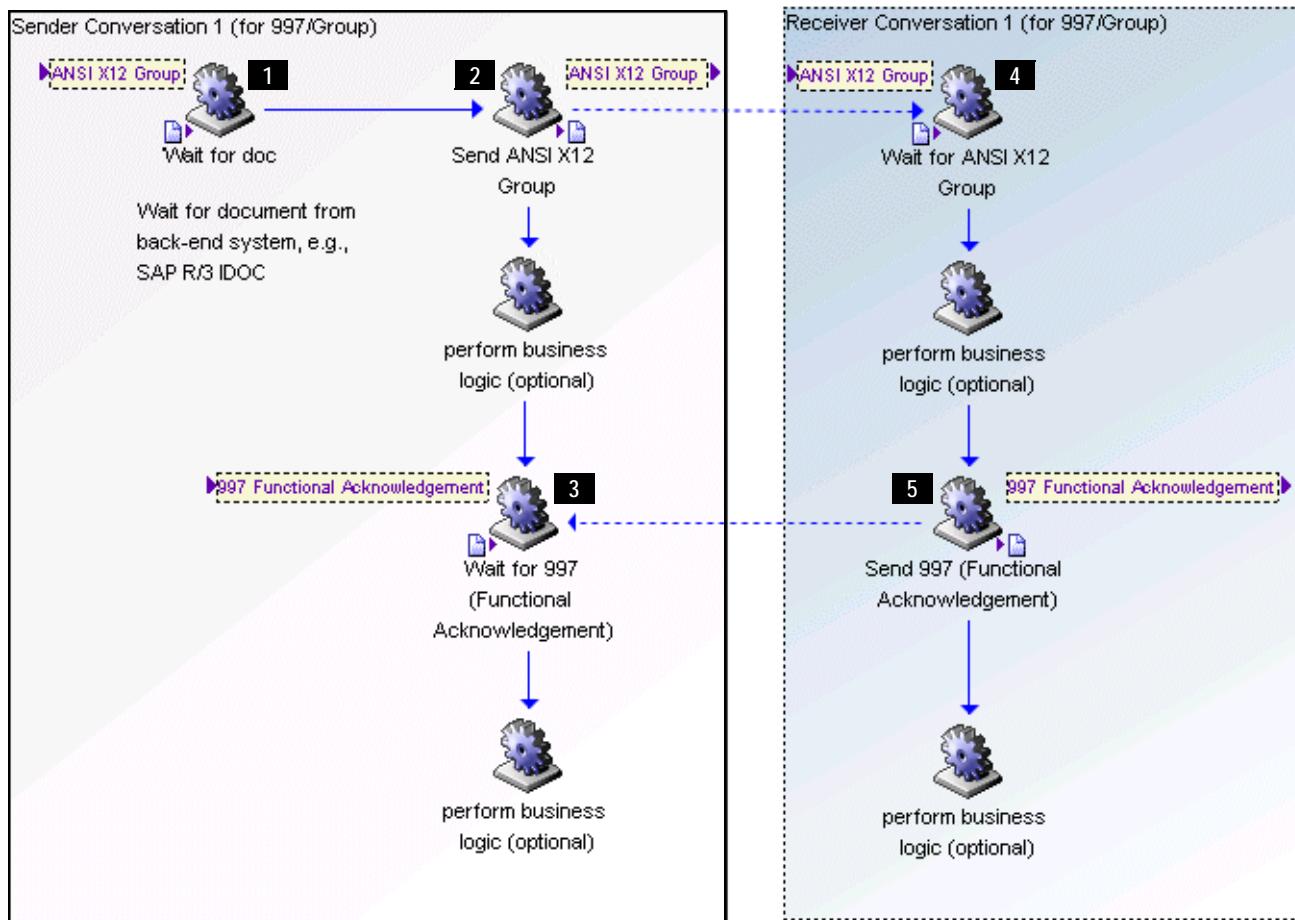
At run time, when the document arrives, the business process begins.

Sample Process Models for ANSI X12

The following screen shots illustrate how a sender and a receiver might set up criteria and actions for business processes involving ANSI X12 documents. One screen shot illustrates the processing of the Group document and the other the processing for the Transaction document. See the tables after the screen shots for more information.

 Note: The steps for sender and receiver are shown in single process models to illustrate how they interact. To establish these business processes, you create a separate process model for each sender and receiver.

Process Model Illustrating Processing of the ANSI X12 Group document

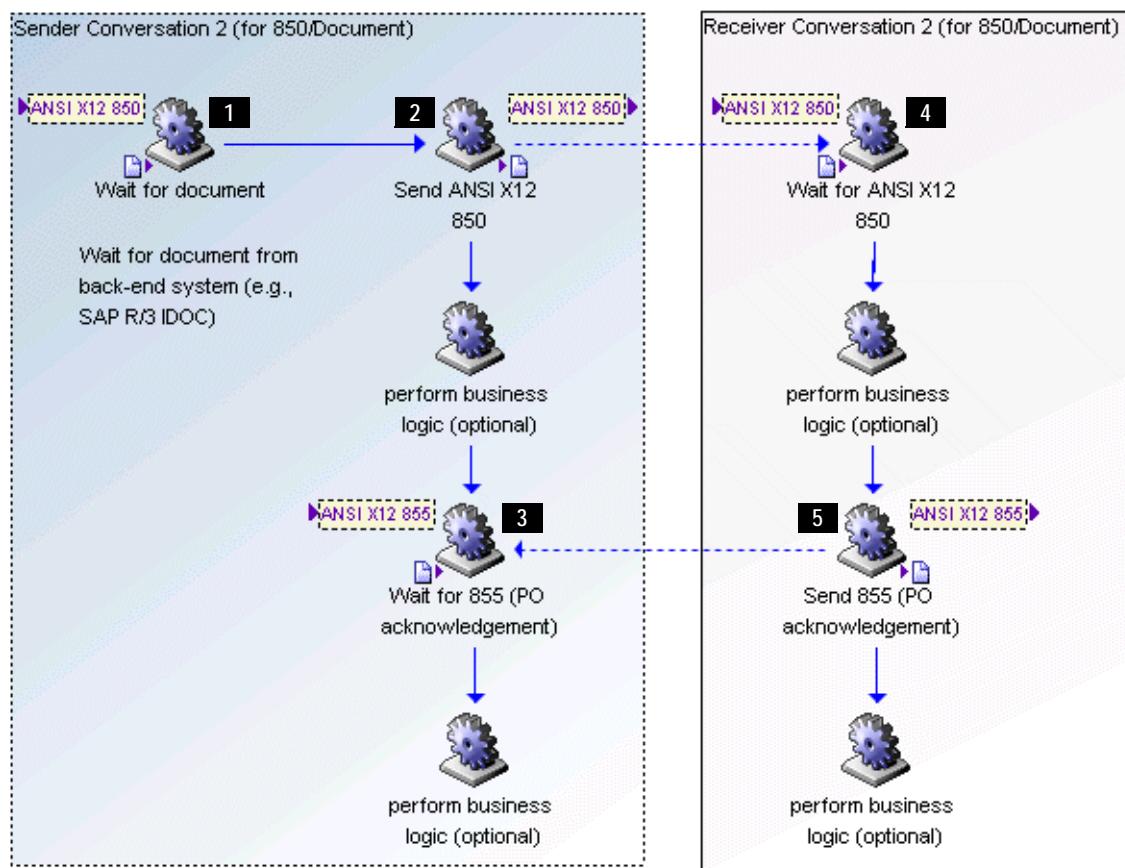


Step Description

- | Step | Description |
|------|--|
| 1 | The business process starts when a document (e.g., SAP R/3 IDOC) is received from the sender's back-end system. |
| 2 | The sender sends an ANSI X12 Group document. |
| 3 | The sender waits for the 997 (Functional Acknowledgment). |
| 4 | The receiver business process begins when the ANSI X12 Group document is received. |
| 5 | In response to the ANSI X12 Group document, the receiver sends the 997 (Functional Acknowledgment) back to the sender. |

Following any send or wait step, partners can choose to perform optional business logic.

Process Model Illustrating Processing of the ANSI X12 850 Transaction document



Step	Description
1	The business process starts when a document (e.g., SAP R/3 IDOC) is received from the sender's back-end system.
2	The sender sends an ANSI X12 850 Transaction document.
3	The sender waits for the response document (e.g., 855 PO Acknowledgment).
4	The receiver business process begins when the ANSI X12 850 Transaction document is received.
5	In response to the ANSI X12 850 Transaction document, the receiver sends the response document (e.g., 855 PO Acknowledgment) back to the sender.

Following any send or wait step, partners can choose to perform optional business logic.

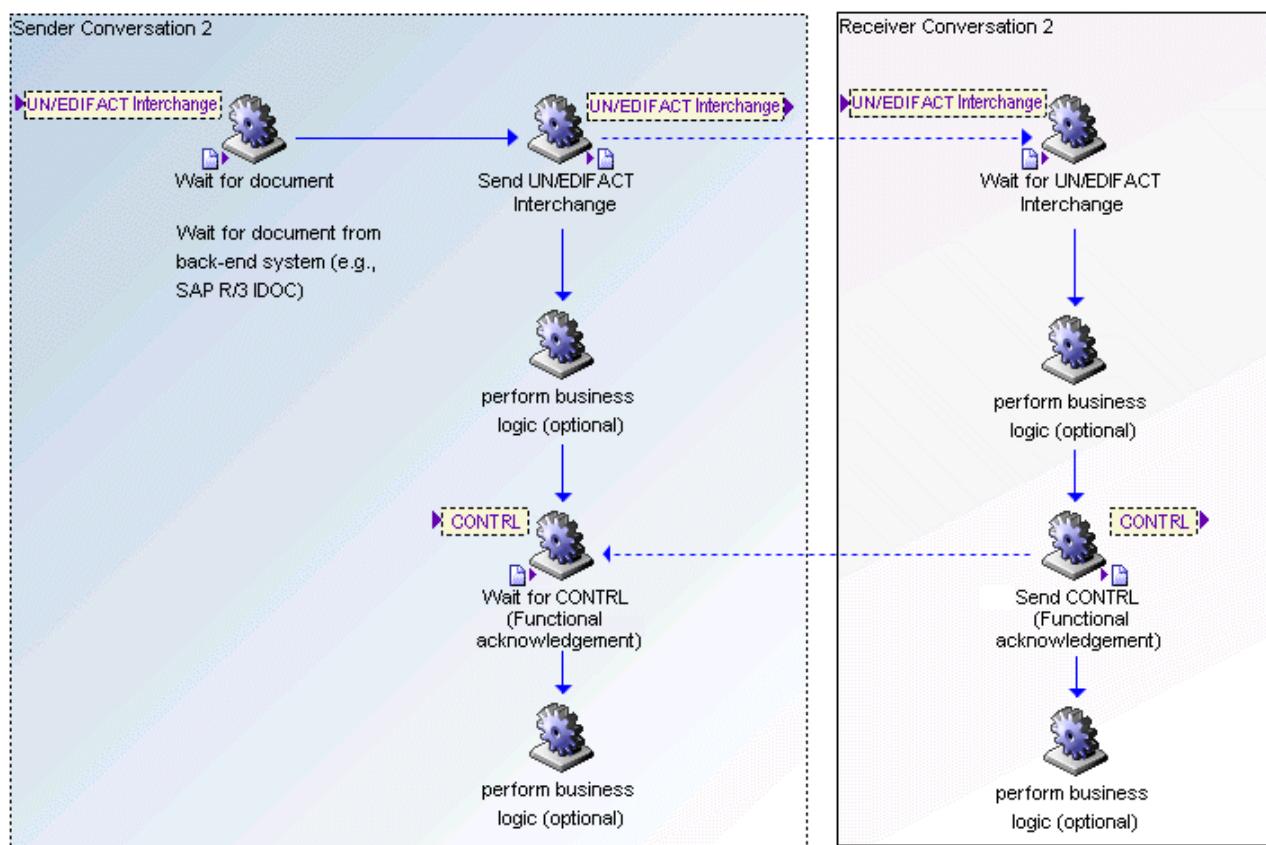
Sample Process Models for UN/EDIFACT

The following screen shots illustrate how a sender and a receiver might set up criteria and actions for business processes involving UN/EDIFACT documents. One screen shot illustrates the processing of the Interchange document and the other the processing for the Transaction document. See the tables after the screen shots for more information.



Note: The steps for sender and receiver are shown in single process models to illustrate how they interact. To establish these business processes, you create a separate process model for each sender and receiver.

Process Model Illustrating Processing of the UN/EDIFACT Interchange document

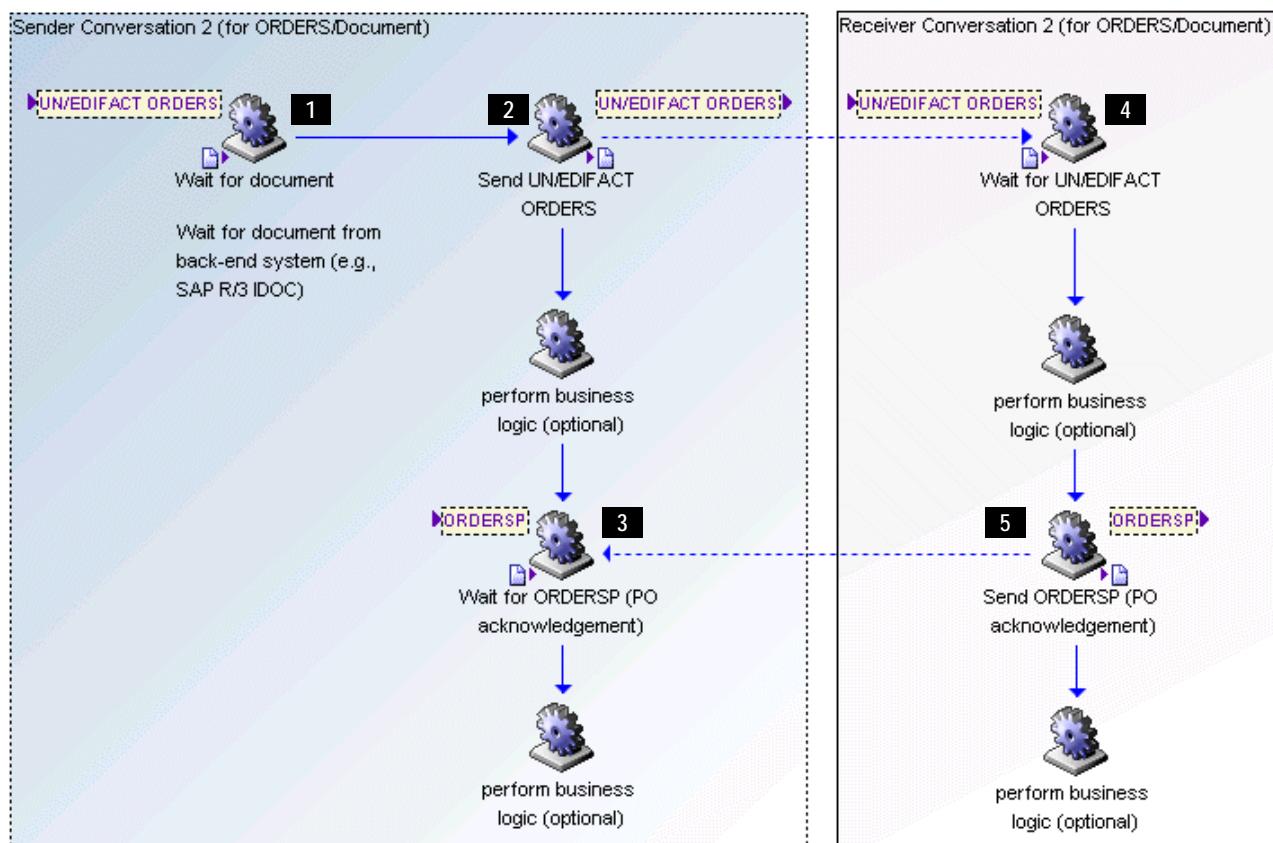


Step	Description
1	The business process starts when a document (e.g., SAP R/3 IDOC) is received from the sender's back-end system.
2	The sender sends an a UN/EDIFACT Interchange document.

Step	Description
3	The sender waits for the CONTRL message (Functional Acknowledgment).
4	The receiver business process begins when the UN/EDIFACT Interchange document is received.
5	In response to the UN/EDIFACT Interchange document, the receiver sends the CONTRL message (Functional Acknowledgment) back to the sender.

Following any send or wait step, partners can choose to perform optional business logic.

Process Model Illustrating Processing of the UN/EDIFACT Transaction document



Step	Description
1	The business process starts when a document (e.g., SAP R/3 IDOC) is received from the sender's back-end system.
2	The sender sends a UN/EDIFACT ORDERS message.

Step	Description
3	The sender waits for the response message (e.g., ORDERSP receipt).
4	The receiver business process begins when the UN/EDIFACT ORDERS message is received.
5	In response to the UN/EDIFACT ORDERS message, the receiver sends the response document (e.g., ORDERSP receipt) back to the sender.

Following any send or wait step, partners can choose to perform optional business logic.

How to Assign Conversation IDs to EDI Documents

For a document to be involved in a business process, it must have a conversation ID. This section describes how the EDI Module assigns a conversation ID to the Interchange, Group, and Transaction documents. The general format of a conversation ID is:

1. receiverID-senderID-instanceID
- OR-
2. senderID-receiverID-instanceID

When assigning the conversation ID, the EDI Module first attempts to assign the conversation ID as receiverID-senderID-instanceID. It checks to see if this conversation ID already exists and if it does, uses it, so that the document rejoins a running business process.

If the first conversation ID does not exist, the EDI Module flips the receiverID and senderID and uses the conversation ID senderID-receiverID-instanceID, which should be a new conversation ID and therefore start a new business process.

The value of instanceID varies based on whether the document is an Interchange document, Group document, or Transaction document.

Instance IDs for Interchange, Group, and Transaction Documents

The following table describes what the EDI Module uses for the `instanceID` portion of the conversation ID.

Type of Document	Instance ID	Notes
Interchange	control number from interchange envelope	<p>The EDI Module always assigns a conversation ID to the Interchange document. The conversation ID has the format:</p> <ol style="list-style-type: none"> 1. receiverID-senderID-controlNum -OR- 2. senderID-receiverID-controlNum
Group	control number form group envelope	<p>The EDI Module always assigns a conversation ID to the Group document. The conversation ID has the format:</p> <ol style="list-style-type: none"> 1. receiverID-senderID-controlNum -OR- 2. senderID-receiverID-controlNum
Transaction	value of instance ID query	<p>The EDI Module only assigns a conversation ID if you define an instance ID query for the transaction set that corresponds to the Transaction document. For example, if you want the EDI Module to assign a conversation ID for a Transaction document that contains an X12 4010 850 transaction set, you must define an instance ID query for an X12 4010 850 transaction. You define instance ID queries from the WmEDIforTN home page. For instructions, see “Defining an Instance ID Query for a Transaction” below.</p>

Defining an Instance ID Query for a Transaction

If you want to use a Transaction document in a business process, you must define an instance ID query for the specific type of transaction set. The instance ID queries are transaction set-specific, but not partner-specific. When you assign an instance ID query to a specific type of transaction set, such as purchase order (850 or ORDERS), the query applies to all transaction sets of that type that you receive. You cannot tailor a query to a specific partner or sender. You can, however, change a query at any time.

The instance ID query should identify a particular segment in the EDI document, for example the segment that contains the PO number. EDI Module uses the value of that segment as the instance ID for all transaction sets of the type. You should choose an instance ID query such that the value is the unique identifier common to both parties in the business process. All documents that you want to join a specific instance of a business process *must* have the same conversation ID.

Prerequisite

Before you can define an instance ID query for a transaction, you must have already installed the TN document type for the transaction. For instructions for installing TN document types, see “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104.

To define an instance ID query for a transaction

- 1 Open the Server Administrator if it is not already open.
- 2 In the Solutions menu of the navigation panel, click EDI. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the EDI Processes menu of the navigation panel, click Define Instance ID Queries.
- 4 From the Select Document Type list, select the TN document type that corresponds to the transaction for which you want to define an instance ID query.
- 5 In the Enter Instance Id Query field, type -or- copy and paste the instance ID query that you want to use. For the queries to use for functional acknowledgments (997 or CONTRL message), see “[Instance IDs for Functional Acknowledgments](#)” on page 356.

Typing the Instance ID Query

Form a query that consists of the document segments that represent the area of the transaction set that has the value on which you would like to base the conversation ID. Separate individual segments using a slash (/).

- Example 1:

To base a business process that is initiated by an 850 (purchase order) on the PO number, enter the following instance ID query:

ST/BEG/BEG03

The BEG03 segment corresponds to the 850's PO number.

- Example 2:

To base a responding document's (855 purchase order response) query on the same value (the PO number), enter the following query:

ST/BAK/BAK03

The BAK03 segment corresponds to the 855's PO number.

Remember when you enter queries that all conversation IDs belonging to the same instance of a business process must have identical values.

Copying and Pasting the Instance ID Query

To reduce the chance of error when typing the query, you can copy and paste the query. To do so, perform the following:

- a Open the IS document type that is associated with the transaction in the webMethods Developer. If you have not already created an IS document type for the transaction, you can do so using the procedure described in ["Creating an IS Document Type for a Transaction" on page 355](#).
 - b In the Editor Panel of the webMethods Developer, locate and select the segment on which you want to base the query.
 - c Right-click the segment and select copy.
 - d Paste the segment in the Enter Instance ID Query field.
- 6 Click Save.

Creating an IS Document Type for a Transaction

You can create an IS document type for a transaction from the flat file schema for the transaction. The flat file schema for a transaction is created when you install the TN document type for a transaction. For more information, see ["Installing TN Document Types and Creating Flat File Schemas" on page 104](#).

By default, the flat file schema is created in the WmEDIforTN package in *EDIFFSchema.Standard.Vversion.Transaction*. To create the corresponding IS document type, perform the following procedure:

 To create an IS document type for a transaction set from a flat file schema

- 1 Start the webMethods Developer if it is not already running.
- 2 In the Navigation Panel, navigate to and select the flat file schema for the transaction. For example, if you want to create an IS document type for the ANSI X12 4010 850 transaction set, navigate the WmEDIforTN package to locate and select:

EDIFFSchema.X12.V4010:T850

The webMethods Developer displays the flat file schema in the Flat File Schema Editor.

- 3 Click the Flat File Structure tab.
- 4 Click the Create Document Type icon in the tool bar.

The IS document type is created in the same folder as the flat file schema and is given the same name with a “DT” appended. For example, the IS document type that corresponds to the ANSI X12 4010 850 flat file schema is:

EDIFFSchema.X12.V4010:T850DT

Instance IDs for Functional Acknowledgments

webMethods that you use the following instance IDs for functional acknowledgments (FAs):

- For an ANSI X12 997, use ST/AK1/AK102, which corresponds to the group control number.
- For a UN/EDIFACT CONTRL message, use UCI/03, which corresponds to the interchange control number.

Sample Business Process

The EDI Module provides a sample to illustrate one way of using EDI documents in a business process. To read an overview of the sample, learn how to set it up and to run it, perform the following procedure.



Important! If you downloaded the WmEDIsamples package from the Knowledge Base on the Advantage Web site at <http://advantage.webmethods.com>, you should delete the WmEDIsamples package before going into production.

To learn about, set up, and run the sample

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **EDI Processes** menu of the navigation panel, click **Run a Demo**.

To continue, follow the instructions on the home page.

Monitoring Your Business Processes

You monitor business processes that involve EDI documents in the same manner that you would monitor any business process. That is, you use webMethods Monitor. For more information about webMethods Monitor, see the webMethods Monitor documentation.

Managing and Analyzing Information about EDI Documents

- Reconciling Functional Acknowledgments 361
- Viewing Log Information for EDI Documents 373
- Viewing Information About EDI Documents 375

Reconciling Functional Acknowledgments

■ Overview	362
■ Turning FA Reconciliation On	363
■ Information the EDI Module Records to Reconcile FAs	363
■ Creating FA Reconciliation Reports	368

1.

Overview

When you use the webMethods EDI Module (EDI Module) with webMethods Trading Networks (Trading Networks), you can have the EDI Module:

- Reconcile Functional Acknowledgments (FAs) with the EDI documents that they acknowledge.
- Generate an FA report to view the statuses of FAs.



Note: Functional acknowledgments (FAs) are not applicable to TRADACOMS.

To learn more about:

- What a functional acknowledgment (FA) is, see Chapter 2, "Using the EDI Module without Other webMethods Components", in the *webMethods EDI Module Concepts Guide*.
- The `wm.b2b.edi.util:generateFA` service that the EDI Module provides to generate FAs, see [Chapter 4, "Generating Functional Acknowledgments"](#).
- How to generate an FA when using Trading Networks, see ["Automatically Generating Functional Acknowledgements" on page 258](#) in [Chapter 15, "Optional Inbound Processing When Using Trading Networks"](#).

FA Reconciliation

To perform FA reconciliation, the EDI Module records each Group/Interchange EDI document that it sends and receives through Trading Networks. Whether it records a Group or an Interchange document depends on the EDI standard of the document.

- For ANSI X12, the EDI Module records each Group document that it sends or receives through Trading Networks.
- For UN/EDIFACT, the EDI Module records each Interchange-level document.

The EDI Module records information about these documents in the EDITRACKING table, which is an EDI Module-specific table in the Trading Networks database.

When the EDI Module receives the FA that corresponds to a Group/Interchange EDI document, the EDI Module updates the FA status for the Group/Interchange document in the EDITRACKING table.

- For ANSI X12, the FA is the 997 document, which acknowledges the group envelope and all of its contents.
- For UN/EDIFACT, the FA is the CONTRL message, which acknowledges an interchange envelope and all of its contents.



Important! For the EDI Module to be able to record information for Group/Interchange EDI documents and to update the FA status when the FA is sent or received, you must exchange the Group/Interchange EDI document and the FAs through Trading Networks.

In addition, for the document to appear in the EDITRACKING table, the document *must* be persisted to Trading Networks, and the document *must* have both a sender ID and a receiver ID.

FA Reports

You can generate an FA report:

- Using the Create FA Reports menu option from the home page of the WmEDIforTN package. For more information, see “[Creating FA Reconciliation Reports](#)” on [page 368](#).
- Invoking the `wm.b2b.edtn.FAReport:generateFA` service. For more information about this service, see *webMethods EDI Module Built-In Services Reference*.

Turning FA Reconciliation On

FA reconciliation is enabled on a partner pair (sender/receiver) basis. You define whether you want to reconcile FAs for a partner pair using the *FAReconciliation* EDITPA variable. Set the *FAReconciliation* EDITPA variable to `true` if you want the EDI Module to reconcile FAs; set it to `false` if you do not. If you want to turn FA reconciliation on or off for *all* partners, set the *FAReconciliation* EDITPA variable in the default EDITPA and leave *FAReconciliation* blank in all partner-specific EDITPAs. For more information, see “[FAReconciliation EDITPA Variable](#)” on [page 136](#). For more information about default and partner-specific EDITPAs, see “[Defining EDI Trading Partner Agreements](#)” on [page 116](#).

Information the EDI Module Records to Reconcile FAs

As stated above, when an ANSI Group or UN/EDIFACT Interchange document is sent or received through Trading Networks, the EDI Module records an entry for the document in the EDITRACKING table. Regardless of whether FA reconciliation is on or off, the EDI Module *always* records information in the EDITRACKING table for Group/Interchange EDI documents that are exchanged through Trading Networks. However, the value that the EDI Module records as the FA status for the Group/Interchange EDI documents in the EDITRACKING table is different based on whether FA reconciliation is on or off.

When an FA is sent or received through Trading Networks, the EDI Module attempts to locate an entry in the EDITRACKING table for the corresponding Group/Interchange EDI document. The EDI Module uses the document IDs to match the FA to the corresponding

Group/Interchange EDI document. The document ID of a Group document is the group control number. The document ID of an Interchange document is the IC control number. For more information, see “[DocumentID](#)” on page 197.



Note: For EDI Module to reconcile FAs with Group/Interchange EDI documents, the FAs and the Group/Interchange EDI documents must be exchanged on the same Integration Server.

The following table describes the information that the EDI Module records in the EDITRACKING table.

Type of document exchanged through Trading Networks	When <i>FAReconciliation</i> is true , the EDI Module....	When <i>FAReconciliation</i> is false , the EDI Module....
ANSI X12 Group -OR- UN/EDIFACT Interchange	Adds an entry to the EDITRACKING table for the Group/Interchange document and sets the FA status to None.	Adds an entry to the EDITRACKING table for the Group/Interchange document and sets the FA status to Disable.
FA	<p>Attempts to locate the Group/Interchange document to which the FA corresponds.</p> <ul style="list-style-type: none"> ■ If the EDI Module <i>does</i> locate the Group/Interchange document, it updates the FA status for the Group/Interchange document based on the status in the FA. For example, if the status in the FA is “A” (Accept), the EDI Module updates the FA status in the EDITRACKING table to Accept. For a list values for FA status in the EDITRACKING table, see “FA Statuses” on page 365. ■ If the EDI Module <i>does not</i> locate the Group/Interchange document, it adds an entry to the EDITRACKING table for the FA. 	<p>Attempts to locate the Group/Interchange document to which the FA corresponds.</p> <ul style="list-style-type: none"> ■ If the EDI Module <i>does</i> locate the Group/Interchange document: <ul style="list-style-type: none"> ■ If the current FA status for the Group/Interchange document is Disable, it does nothing. ■ If the current FA status for the Group/Interchange document is None, it updates the FA status to Disable. ■ If the EDI Module <i>does not</i> locate the Group/Interchange document, it adds an entry to the EDITRACKING table for the FA.

FA Statuses

The following table lists the values of the FA status recorded for an ANSI X12 Group or UN/EDIFACT Interchange document in the EDITRACKING table.

FA Status	Meaning
None	The EDI Module has not yet received or sent an FA to acknowledge this document.
Disable	The <i>FAReconciliation</i> EDITPA variable is set to <code>false</code> , which disables FA reconciliation and reporting. For more information, see “ FAReconciliation EDITPA Variable ” on page 136.
Duplicate	The EDI Module has one or more other documents recorded in the EDITRACKING table that match the FA for this document. For more information, see “ Controlling FA Status for Documents Submitted Multiple Times ” on page 366.
Error	The EDI Module received an FA for which it could not locate a matching document in the EDITRACKING table. The EDI Module adds an entry to the EDITRACKING table for the FA and assigns it this status.
Duplicate FA/Errors	The EDI Module received more than one FA that matches this document.
Accept	The EDI Module received a single FA that matches this document and the FA has either: <ul style="list-style-type: none"> ■ An “A”(Accept) status on the confirmed level (ANSI X12) ■ A “7” status on the confirmed level (UN/EDIFACT)
Accept with Errors	The EDI Module received a single FA that matches this document and the FA has an “E” (Errors) status on the confirmed level.
Partially Accept	The EDI Module received a single FA that matches this document and the FA has an “P” (Partially Accept) status on the confirmed level.
Reject	The EDI Module received a single FA that matches this document and the FA has either: <ul style="list-style-type: none"> ■ An “R” (Reject) status on the confirmed level (ANSI X12) ■ An “4” status on the confirmed level (UN/EDIFACT)
FA Errors	The EDI Module encountered other unknown errors.
Interchange Received	The EDI Module received an interchange.

Controlling FA Status for Documents Submitted Multiple Times

Perform the following procedure to control how the EDI Module assigns FA status when you send (or receive) a document multiple times before the receiver returns an FA.

To control what happens when a document is sent multiple times before receiving an FA

- 1 Open the Server Administrator if it is not already open.
- 2 In the Solutions menu of the navigation panel, click EDI. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the Configuration menu of the navigation panel, click Configure Properties.
- 4 Click Edit Properties Settings.
- 5 Add or update the *EDIResolveDuplicates* property as follows:

<u>Specify...</u>	<u>Meaning</u>
false	When you send a document multiple times before the receiver returns an FA, the status of each document is set to None ("100") in the EDITRACKING table. This indicates that the EDI Module has not yet received an FA to acknowledge the document. When the receiver returns an FA for one of these documents, the status of each document is changed to Duplicate ("120"). This indicates that multiple documents exist in the EDITRACKING table that match the FA for this document. This is the default.

Note: This behavior also applies to documents that you receive.

<u>Specify...</u>	<u>Meaning</u>
true	<p>When you send a document the first time, its status is set to None ("100") in the EDITRACKING table. This indicates that the EDI Module has not yet received an FA to acknowledge the document.</p> <p>If the same document is sent again before the receiver returns an FA, the status of the <i>first</i> document changes to Duplicate ("120"). This indicates that multiple documents exist in the EDITRACKING table that match the FA for this document. The status of the <i>second</i> document is set to None. This behavior occurs for each subsequent document sent before an FA is returned.</p> <p>When the receiver returns an FA, the FA will acknowledge only the document that was sent <i>last</i>. The status of the document that was sent last will be set according to the FA status reported in the FA document. The status of all prior documents remains Duplicate.</p>

Note: This behavior also applies to documents that you receive.

- 6 Click **Save Changes**. The EDI Module updates the configuration in memory (so the changes take effect immediately) and in the WmEDI/config/properties.cnf file.

Creating FA Reconciliation Reports

You can create an FA report that displays a list of ANSI X12 Group documents and/or UN/EDIFACT Interchange documents and their reconciliation statuses using either the **Create FA Reports** menu option from the home page of the WmEDIforTN package or using `wm.b2b.edtn:FAReport:generateFAReport`. This section describes how to generate the report from the WmEDIforTN home page. For information about using the `generateFAReport` service, see *webMethods EDI Module Built-In Services Reference*.

To create an FA report using the home page of the WmEDIforTN package

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **FA** menu of the navigation panel, click **Create FA Reports**.
- 4 Fill in the following fields to set the name of the report, search criteria for the entries that you want to include in the report, and how you want the entries in the report sorted.

For this field...	Specify
<code>reportFileName</code>	The name you want to give to the FA report file. The format of the file name is <i>timestamp (YYYY_MM_DD) reportFileName.template</i> . For example, for a <code>reportFileName</code> of <code>testReport</code> , and a Report template of <code>HTML</code> , the file might be saved as <code>2002_07_12testReport.html</code> .
Report template	Whether you want the report in <code>Text</code> or <code>HTML</code> format.
Sender ID	The corporate name (as specified in the Trading Networks profile) of the sender of the EDI documents you want included in the report. If you leave this field blank, the report will contain information for documents from all senders.
Receiver ID	The corporate name (as specified in the Trading Networks profile) of the receiver of the EDI documents you want included in the report. If you leave this field blank, the report will contain information for documents for all receivers.

For this field...	Specify
From date	<p>Indicates that you want to search for EDI Group/Interchange documents that were sent since the date you specify. For example, if you specify year: 2002, month: July, day: 12, hour: 12, minute: 00, the report would contain EDI documents sent after 12 AM, July 12, 2002. Specify hours in 24-hour format.</p> <p>Note: If you do not specify a value for <i>all</i> of the date and time fields (year, month, day, hour, and minute) the “From Date” will start at the date of the oldest document.</p>
To date	<p>Indicates that you want to search for EDI Group/Interchange documents sent before the date you specify. For example, if you specify year: 2002, month: July, day: 12, hour: 12, minute: 00, the report would contain EDI documents sent before 12 AM, July 12, 2002. Specify hours in 24-hour format.</p> <p>Note: If you do not specify a value for <i>all</i> of the date and time fields (year, month, day, hour, and minute) the “To Date” will start at the current date and time.</p>
FA from date	<p>Indicates that you want to search for entries for which FAs were sent since the date you specify. For example, if you specify year: 2002, month: July, day: 12, hour: 12, minute: 00, the report would contain documents that have corresponding FAs that were sent after 12 AM July 12, 2002. Specify hours in 24-hour format.</p> <p>Note: If you do not specify a value for <i>all</i> of the date and time fields (year, month, day, hour, and minute) the “From Date” will start at the date of the oldest document.</p>
FA to date	<p>Indicates that you want to search for entries for which FAs were sent before the date specified. For example, if you specify year: 2002, month: July, day: 12, hour: 12, minute: 00, the report would contain documents that have corresponding FAs sent before 12 AM July 12, 2002. Specify hours in 24-hour format.</p> <p>Note: If you do not specify a value for <i>all</i> of the date and time fields (year, month, day, hour, and minute) the “To Date” will start at the current date and time.</p>

For this field...	Specify
Group type	The group type of the EDI Group/Interchange documents that you want included in the report, e.g., PO or IN.
Version	The version of the EDI Group/Interchange documents that you want included in the report, e.g., 4010.
Status	The FA status of the documents that you want included in the report. For a list of FA statuses and their descriptions, see “ FA Statuses ” on page 365.
Sort by	How you want the EDI Module to sort the entries returned in the FA report. Default: Document Type.

- 5 Click Generate FA report. After the report is generated, a screen appears asking if you would like to view a list of all the FA reports on your Integration Server. To do so, click where indicated. A list of your reports appears.

FA Report List:	
<input type="checkbox"/> Check All	Report Name
<input type="checkbox"/>	2002_07_10_11_11_11_11.TXT
<input type="checkbox"/>	2002_07_10_02_39_02_39.TXT
<input checked="" type="checkbox"/>	2002_07_12_03_29_03_29testReport.TXT
<input checked="" type="checkbox"/>	2002_07_12_03_31_03_31testReport.html

Delete Files

- To delete one or more reports, select the check box next to the report name, and click Delete Files.
- To view one of the reports, click the report name. (These reports also are located in *webMethods6\IntegrationServer\WmEDIforTN\Pub\FAResorts*.)

Following is an HTML sample of an FA report:

FA report:

Document ID	Group ID	Group Type	Document Type	Group Version	FA Status	Doc Timestamp
5066i600ts06nk0100000001	1	PO	X12 Group	4010	Accept	Jul 12, 2002 2:05:00 PM
5066i600ts06nk0100000003	3	PO	X12 Group	4010	Reject	Jul 12, 2002 2:06:01 PM
5066i600ts06nk0100000005	8	PO	X12 Group	4010	None	Jul 12, 2002 4:04:09 PM
5066i600ts06nk0100000011	12	IN	X12 Group	4030	Duplicate	Jul 12, 2002 4:05:09 PM
5066i600ts06p5ki00000015	12	IN	X12 Group	4030	Duplicate	Jul 12, 2002 4:06:00 PM

Following is a text sample of an FA report:

Functional Acknowledgement Report Report Run Time: 2002_07_12 06:05:38

```

DocumentID: 5066i600ts06p5ki00000001
Group ID: 1
Sender ID: test1
Receiver ID: test2
Group Type: PO
Document Type: X12 Group
Group Version: 4010
FA Status: Accept
Doc Timestamp: Jul 12, 2002 2:05:00 PM
FA Timestamp: Jul 12, 2002 3:11:56 PM
Related DocID: 5066i600ts06p5ki00009876

```

The following describes the fields in the FA report.

Field	Definition
DocumentID	Document ID of the EDI document. For information about how the EDI Module forms the document ID, see "DocumentID" on page 197 .
Group ID	Group ID of the EDI document. For information about how the EDI Module forms the Group ID, see "GroupID" on page 197 .
Sender ID	The corporate name (as specified in the Trading Networks profile) of the sender of the EDI document.
Receiver ID	The corporate name (as specified in the Trading Networks profile) of the receiver of the EDI document.
Group type	The group type of the EDI document type, e.g., PO, IN.
Document type	The TN document type of the EDI document, e.g., X12 Group.
Group Version	The version of the EDI document, e.g., 4010.
FA Status	The FA status of the EDI document. For a list of FA statuses and their descriptions, see "FA Statuses" on page 365 .

Field	Definition
Doc Timestamp	Date and time the EDI Module added the entry for the EDI document to the EDITRACKING table.
FA Timestamp	Date and time the EDI Module added information for the FA that corresponds to the EDI document to the EDITRACKING table.
Related Doc ID	Document ID of the FA that corresponds to the EDI document.

Viewing Log Information for EDI Documents

- Overview 374
- Viewing Server Log 374
- Viewing the Trading Networks Activity Log 374

Overview

During processing, the webMethods EDI Module (EDI Module) logs information to the server log. Additionally, if you are using the EDI Module with webMethods Trading Networks (Trading Networks), the EDI Module also logs information to the Trading Networks activity log.

Viewing Server Log

The EDI Module logs informational, warning, and error messages to the server log of the Integration Server. The messages that the EDI Module adds to the server log have one of the following keys:

Key	For messages issued from the...
EDICOR	WmEDI package of the EDI Module
EDIFTN	WmEDIforTN package of the EDI Module

The key is part of the identification of a message. For example, the following is a message that is issued from the WmEDI package of the EDI Module and uses the key, EDICOR:

[EDICOR.000020.000404] No transaction set defined for SEF.

To view the server log

To view the server log, use the Server Administrator. For steps to view the server log, see the chapter about working with logs in the *webMethods Integration Server Administrator's Guide*.

Viewing the Trading Networks Activity Log

The EDI Module and Trading Networks log information about the processing of EDI documents through Trading Networks in the activity log.

To view the Trading Networks activity log

To view the activity log, use the Activity Log screen of the Trading Networks Console. For steps to view the activity log, see the *webMethods Trading Networks User's Guide*.

Viewing Information About EDI Documents

■ Overview	376
■ EDI Documents You Can View	376
■ Viewing EDI Documents Using the Trading Networks Console	377
■ Viewing Information about EDI Documents in Queues	379

Overview

When you use webMethods EDI Module (EDI Module) with webMethods Trading Networks (Trading Networks), you can view the EDI documents that have passed through your system if they have been saved to the Trading Networks database. You view documents using the *Transaction Analysis* screen of the Trading Networks Console.

To learn more about using the *Transaction Analysis* screen to view documents, see the chapter about managing documents in the *webMethods Trading Networks User's Guide*.

EDI Documents You Can View

By default, all Interchange, Group, and Transaction documents that the EDI Module creates based on the setting of the *splitOption* EDITPA variable are saved to the Trading Networks database.

For TRADACOMS documents all Transmission, Batch, and File documents that the EDI Module creates based on the setting of the *TRADACOMS/splitOption* EDITPA variable are saved to the Trading Networks database.

The default is to save the documents because of the TN document types that you install for EDI documents. In the TN document types, the **Trading Networks Save Document to Database** pre-processing action is set so the Interchange, Group, and Transaction documents (or Transmission, Batch, and File documents) are saved.

Additionally, you can save the original EDI document from which the EDI Module split the Interchange, Group, and Transaction documents (or Transmission, Batch, and File documents). You define whether to save the original EDI document using the *persistMultipleDocEnvelope* EDITPA variable.

For ANSI X12 and UN/EDIFACT users to learn more about:

- How the *EDI Module* uses the EDITPA variable *splitOption* to split documents into Interchange, Group, and Transaction documents, see Chapter 3, "Using the EDI Module with Trading Networks", in the *webMethods EDI Module Concepts Guide* and "["splitOption EDITTPA Variable"](#) on page 119.
- EDITPAs and how to define them, see "["Defining EDI Trading Partner Agreements"](#) on page 116;
- The *persistMultipleDocEnvelope* EDITPA variable, see "["persistMultipleDocEnvelope EDITPA Variable"](#) on page 123.

For TRADACOMS users to learn more about:

- How the *EDI Module* uses the EDITPA variable *TRADACOMS/splitOption* to split documents into Transmission, Batch, and File documents, see Chapter 3, "Using the EDI Module with Trading Networks", in the *webMethods EDI Module Concepts Guide* and "["TRADACOMS/splitOption" EDITPA Variable](#)" on page 153.
- EDITPAs and how to define them, see "[Defining EDI Trading Partner Agreements](#)" on page 116.
- The *persistMultipleDocEnvelope* EDITPA variable for TRADACOMS documents, see "["persistMultipleDocEnvelope" EDITPA Variable](#)" on page 123.

Viewing EDI Documents Using the Trading Networks Console

You can use the Transaction Analysis screen of the Trading Networks Console to view the following types of information about your EDI documents:

- Sender and receiver of the EDI document
- Date the document was received
- Processing status of the EDI document
- Content of the EDI document
- Documents related to the EDI document, e.g., the functional acknowledgment (FA) that is associated with an ANSI X12 Group document

By default the Trading Networks Console displays the related documents identified by their control numbers. If a document does not contain a control number (e.g., it is an invalid document), the Trading Networks Console cannot display document relationships based on control numbers. Instead, a warning message will be logged for the envelope or group document's activity.

Note: You can override this default to display related documents identified by their relationship labels instead of by their control numbers. For example, if you view an ANSI X12 envelope, the screen displays that envelope's related group with the relationship label Envelope - Group. To set the flag that controls this, see "[Viewing Related Documents With Relationship Labels](#)" on page 380.

To view EDI documents using the Trading Networks Console

For steps to view documents using the Transaction Analysis screen of the Trading Networks Console, see the chapter about managing documents in the *webMethods Trading Networks User's Guide*.

Trading Networks Processing Status and EDI Documents

The following table lists the processing statuses that Trading Networks sets while processing EDI documents along with their meanings. You can use these statuses as criteria for selecting the EDI documents that you want to view on the **Transaction Analysis** screen. Trading Networks displays these statuses for documents on the **Transaction Analysis** screen.

Processing Status	Meaning
NEW	Trading Networks has received the EDI document but it has not yet been completed recognizing and/or processing the document with its processing rules.
QUEUED	<p>The document is an EDI document that is either in a scheduled delivery queue for EDI batch processing or for delivery to VAN.</p> <p>For more information about EDI batch processing, Chapter 17, “Batching EDI Documents”. For more information about delivering documents to VANs, see Chapter 18, “Retrieving and Delivering EDI Documents from and to VANs”.</p>
DONE	<p>Trading Networks has completed its processing of the EDI document. Note that the following processing actions might not be complete:</p> <ul style="list-style-type: none"> ■ Execute a Service. Execution of the service might not be complete if it was invoked asynchronously or executed using a service execution task. ■ Deliver Document By. The document might not be delivered to the receiving partner yet.
DONE W/ ERRORS	<p>Trading Networks has completed its processing of the EDI document. However, errors occurred during document recognition or document processing. Note that the following processing actions might not be complete:</p> <ul style="list-style-type: none"> ■ Execute a Service. Execution of the service might not be complete if it was invoked asynchronously or executed using a service execution task. ■ Deliver Document By. The document might not be delivered to the receiving partner yet.

Processing Status	Meaning
ABORTED	<p>Trading Networks encountered a fatal error before completing the processing specified in the processing rule. This status is typically used in one of the following situations:</p> <ul style="list-style-type: none"> ■ Trading Networks detected an infinite loop in the processing rule. That is, a document triggers a processing rule that creates and submits a new document that triggers the same processing rule again. ■ Trading Networks encountered an internal error. ■ The built-in service <code>wm.tn.route:abort</code> was executed for the service. For more information about this service, see the <i>webMethods Trading Networks Built-in Services Reference</i>.

Viewing Information about EDI Documents in Queues

When you use the *Deliver Document By* processing action of a processing rule to place a document in a scheduled delivery queue, Trading Networks keeps track of the delivery using a *delivery task*. Trading Networks actually places the delivery task, which is associated with the document to deliver, in the scheduled delivery queue. As a result, to view information about documents in a scheduled delivery queue, you can use the *Tasks* screen of the Trading Networks Console.

You use scheduled delivery queues when you want to batch EDI documents or when you want to deliver EDI documents to VANs. To view information about the progress of the batching or delivery, use the *Tasks* screen of the Trading Networks Console.

To view information about queued EDI documents

For steps to view information about delivery tasks using the *Tasks* screen of the Trading Networks Console, see the chapter about managing tasks in the *webMethods Trading Networks User's Guide*.

To learn more about:

- For more information about how to set up for batching EDI documents, see [Chapter 17, "Batching EDI Documents"](#).
- For more information about how to set up to deliver EDI documents to VANs, see [Chapter 18, "Retrieving and Delivering EDI Documents from and to VANs"](#).

Viewing Related Documents With Relationship Labels

When you view related documents on the Transaction Analysis screen of the Trading Networks Console, by default the screen displays the related documents identified by their control numbers. You can override this default to display the related documents identified by their relationship labels instead. For example, if you view an ANSI X12 envelope, the screen displays that envelope's related group with the relationship label **Envelope - Group**. To do this, set the `EDIUseNewRelationshipLabel` flag to true, as described below.



Note: All documents that were submitted to Trading Networks *before* you set this flag to true will continue to be identified by their control numbers, not by relationship labels.

To set the relationship label flag

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **Configuration** menu of the navigation panel, click **Configure Properties**.
- 4 Click **Edit Properties Settings**.
- 5 Set the following property to true:

`EDIUseNewRelationshipLabel=true`

When you set this flag, the EDI Module updates the configuration in memory (so the changes take effect immediately) and in the `WmEDI/config/properties.cnf` file.

The following table lists the relationship labels of related documents.

For...	If you view related documents for...
ANSI X12 and all supported sub-standards	<p>Envelopes</p> <ul style="list-style-type: none"> ■ Related groups are shown labeled Envelope - Group
	<p>Groups</p> <ul style="list-style-type: none"> ■ Related envelopes are shown labeled Envelope - Group ■ Related transactions are shown labeled Group - Transaction ■ Related functional acknowledgments (FAs) are shown labeled Group - FA ■ Related duplicate FAs are shown labeled Group - DUP_FA or Group - Duplicate_Doc <p>Group - DUP_FA means that the FA (either sent or received) acknowledges a document that has already been acknowledged.</p> <p>Group - Duplicate_Doc means that the FA (either sent or received) can acknowledge more than one identical document that has not been acknowledged.</p>
	<p>Transaction</p> <ul style="list-style-type: none"> ■ Related groups are shown labeled Group - Transaction
	<p>Functional Acknowledgments (FAs)</p> <ul style="list-style-type: none"> ■ Related groups are shown labeled Group - FA ■ If the FA is a duplicate FA, related groups are shown labeled Group - DUP_FA or Group - Duplicate_Doc <p>Group - DUP_FA means that the FA (either sent or received) acknowledges a document that has already been acknowledged.</p> <p>Group - Duplicate_Doc means that the FA (either sent or received) can acknowledge more than one identical document that has not been acknowledged.</p>

For...	If you view related documents for...
UN/EDIFACT and all supported sub-standards	<p>Envelopes</p> <ul style="list-style-type: none"> ■ Related transactions are shown labeled Envelope - Transaction ■ Related groups are shown labeled Envelope - Group ■ Related functional acknowledgments (FAs) are shown labeled Envelope - FA ■ Related duplicate FAs are shown labeled Envelope - DUP_FA or Envelope - Duplicate_Doc <p>Envelope - DUP_FA means that the FA (either sent or received) acknowledges a document that has already been acknowledged.</p> <p>Envelope - Duplicate_Doc means that the FA (either sent or received) can acknowledge more than one identical document that has not been acknowledged.</p>
	<p>Transactions</p> <ul style="list-style-type: none"> ■ Related envelopes are shown labeled Envelope - Transaction ■ Related groups are shown labeled Group - Transaction
	<p>Groups</p> <ul style="list-style-type: none"> ■ Related groups are shown labeled Envelope - Group ■ Related transactions are shown labeled Group - Transaction
	<p>Functional Acknowledgments (FAs)</p> <ul style="list-style-type: none"> ■ Related envelopes are shown labeled Envelope - FA ■ If the FA is a duplicate FA, related envelopes are shown labeled Envelope - DUP_FA or Envelope - Duplicate_Doc
TRADACOMS	<p>Transmissions</p> <ul style="list-style-type: none"> ■ Related files are shown labeled Transmission - File ■ Related batches are shown labeled Transmission - Batch
	<p>Batches</p> <ul style="list-style-type: none"> ■ Related files are shown labeled Batch - File ■ Related batches are shown labeled Transmission - Batch

For...	If you view related documents for...
	Files <ul style="list-style-type: none">■ Related transmissions are shown labeled Transmission - File■ Related batches are shown labeled Batch - File
Multiple-envelope documents	Multiple-envelope documents and their child documents are shown labeled Multiple Envelope - Envelope

Non-Standard Processing

■ Overview	386
■ Defining Partner Information	386
■ Defining Control Number Information for Partners	408
■ Processing Inbound EDI Documents using Trading Networks	410
■ Forming EDI Documents to Send Outbound When using Trading Networks	412

Overview

The EDI Module can process the interchange, groups, and transaction sets of an ANSI X12 or UN/EDIFACT EDI document using either standard or non-standard processing.

- For standard processing, the EDI Module processes the interchange, all its groups, and all its transactions sets using settings that you define for the interchange sender/receiver pair.
- For non-standard processing, you can specify different processing settings for each group within the interchange. The EDI Module processes the groups and the transactions sets within each group using the settings you define for the group sender/receiver pairs.

For more information, see ["Advantages and Disadvantages of Standard vs Non-Standard Processing" on page 111](#) in Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT)".

This appendix contains *only* information for when non-standard processing differs from standard processing.

Defining Partner Information



Important! This section describes how to define partner information when using non-standard processing. It does *not* contain all information about defining partner information, but only areas that are different from standard processing. You should also read the information provided in [Chapter 9, "Defining Partner Information \(ANSI X12 and UN/EDIFACT\)"](#).

The following table lists the information you must define for interchange and group sender/receiver pairs that will be in the EDI documents you expect to process.

For...	Define...
Interchange sender/receiver pairs	<ul style="list-style-type: none"> ■ Interchange sender receiver pair information that you set using the EDI Module home page. For more information, see "Defining Interchange-Level Sender/Receiver Pair Information" on page 390.
Group sender/receiver pairs	<ul style="list-style-type: none"> ■ Trading Networks profiles for each group sender and receiver. For more information, see "Defining Trading Networks Profiles" on page 387.

For...	Define...
Relationship of group sender/receiver pairs to the interchange sender/receiver	<ul style="list-style-type: none"> ■ All the group sender/receiver pairs that are associated with the interchange sender/receiver pair. You define this information using the EDI Module home page. For more information, see "Defining Group-Level Sender/Receiver Pair Associations" on page 404.
Tailoring how the EDI Module processes documents	<ul style="list-style-type: none"> ■ Default EDITPA that defines the settings that you want to use for most partner pairs. ■ Partner-specific EDITPAs for group sender/receiver pairs if you want to override the settings in the default EDITPA. <p>For more information, see "Defining EDI Trading Partner Agreements" on page 116. For information related specifically to non-standard processing, see "Defining a Partner-Specific EDITPA" on page 388.</p>
How the EDI Module validates control numbers in documents from a sender/receiver pair	<ul style="list-style-type: none"> ■ Whether to validate control numbers. For more information, see "Turning Inbound Control Number Validation On or Off" on page 408. ■ Actions the EDI Module is to take when it encounters invalid control numbers. For more information, see "Defining Actions for Invalid Control Numbers" on page 408. ■ Control number validation settings (control number cap, minimum, increment, and window). For more information, see "Control Number Cap, Minimum, Increment, and Window" on page 172 in Chapter 11.

Defining Trading Networks Profiles

To identify the trading partners with whom you want to exchange documents, set up profiles. You must define profiles for:

- Your own corporation if you have not already done so. This is referred to as the My Enterprise profile in Trading Networks.
- Your trading partners *only* at the group level.

For more information about defining Trading Networks profiles, see "[Defining Trading Networks Profiles](#)" on page 113 in Chapter 9, "[Defining Partner Information \(ANSI X12 and UN/EDIFACT\)](#)".

Defining a Partner-Specific EDITPA

You only need to create partner-specific EDITPAs if you have one or more sender/receiver pairs that require settings that are different from those you specify in the default EDITPA. When creating a partner-specific EDITPA, you have to specify only the information that is different from the defaults. Define partner-specific EDITPAs for group-level sender/receiver pairs.

For more information about default and partner-specific EDITPAs, see “[Defining EDI Trading Partner Agreements](#)” on page 116 in Chapter 9, “[Defining Partner Information \(ANSI X12 and UN/EDIFACT\)](#)”. For more information about defining partner-specific EDITPAs, see “[Defining a Partner-Specific EDITPA](#)” on page 117.



Note: You can disable partner-specific EDITPAs. When you disable a partner-specific EDITPA, the EDI Module functions as if the partner-specific EDITPA does not exist. That is, the EDI Module uses the values in the default EDITPA.

wm.b2b.editn.TPA:EDITPA IS Document Type

When you create an EDITPA, you supply values for the variables in the `wm.b2b.editn.TPA:EDITPA IS` document type. This section describes *only* the EDITPA variables that are affected by non-standard processing. For a description of all the EDITPA variables, see “[wm.b2b.editn.TPA:EDITPA IS Document Type](#)” on page 388 in Chapter 9, “[Defining Partner Information \(ANSI X12 and UN/EDIFACT\)](#)”.

<i>splitOption</i> EDITTPA Variable	Default: Transaction
The <i>splitOption</i> variable indicates how you want the EDI Module to split an interchange segment with in an EDI document. You can specify Interchange, Group, or Transaction.	
When you are using non-standard processing and you specify Interchange, the EDI Module will split the document at the Group level.	
For more information, see “ splitOption EDITTPA Variable ” on page 119.	
<i>ControlNumberManagement/validateInboundEnvelopeControlNumbers</i> EDITPA Variable	Default: false
When you are using non-standard processing, the EDI Module does <i>not</i> use this EDITPA variable. Instead it uses the Validate inbound envelope control numbers setting that you set from the Interchange Information Detail screen of the EDI Module home page.	
For more information about accessing and using this screen, see “ Defining Interchange-Level Sender/Receiver Pair Information ” on page 390.	

ControlNumberManagement/duplicateControlNumberAction EDITPA Variable

Default: Error & Continue

The *ControlNumberManagement/duplicateControlNumberAction* variable indicates the action you want the EDI Module to take when it encounters a duplicate control number in an inbound document.

When you are using non-standard processing, this EDITPA is only used for duplicate group control numbers. To set the action for duplicate interchange control numbers, use the **Duplicate control number action** setting that you set from the **Interchange Information Detail** screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390.

You can specify one of the following for

ControlNumberManagement/duplicateControlNumberAction:

- Error & Continue
- ProcessNormally
- Reject

For more information, see

[“ControlNumberManagement/duplicateControlNumberAction EDITPA Variable” on page 125](#)

ControlNumberManagement/outOfSequenceControlNumberAction EDITPA Variable

Default: Error & Continue

The *ControlNumberManagement/outOfSequenceControlNumberAction* variable indicates the action you want the EDI Module to take when it encounters an out-of-sequence control number in an inbound document.

When you are using non-standard processing, this EDITPA is only used for out-of-sequence group control numbers. To set the action for out-of-sequence interchange control numbers, use the **Out of sequence control number action** setting that you set from the **Interchange Information Detail** screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” on page 390.

You can specify one of the following for

ControlNumberManagement/outOfSequenceControlNumberAction:

- Error & Continue
- ProcessNormally
- Reject

For more information, see [“ControlNumberManagement/useReverseRouting EDITPA Variable” on page 127](#).

FAGeneration EDITPA Variables

When you are using non-standard processing, the EDI Module does not use any of the *FAGeneration* EDITPA variables. Instead, the EDI Module uses the corresponding values that you specify on the **Interchange Information Detail** screen of the EDI Module home page. For more information about accessing and using this screen, see “[Defining Interchange-Level Sender/Receiver Pair Information](#)” below.

Defining Interchange-Level Sender/Receiver Pair Information

When you are using non-standard processing, the EDI Module uses the sender and receiver identified on the group envelope header to retrieve the Trading Networks profiles for the sender and receiver and to obtain a partner-specific EDITPA for the sender/receiver pair, if one exists.

GS*PO*186704136*138183702*20030829*1810*1*X*004010\
 sender receiver

The group envelope header in a EDI document contains only the value for the sender and receiver without the corresponding EDI ID qualifier code. For example, it might contain the D-U-N-S number for the sender (186704136), but not the EDI ID qualifier (01) that indicates that the value is a D-U-N-S number. Both the EDI ID qualifier and the value for the partner are required for the EDI Module and for Trading Networks to obtain the correct profiles and EDITPA to process the document.

You define interchange-level sender/receiver pair information to provide the EDI ID qualifiers for the group level senders and receivers. Using the WmEDIforTN home page you define the EDI ID qualifier to use for the group sender and group receiver based on the interchange sender/receiver pair. In other words, when an EDI document has a specific interchange sender/receiver pair, you specify the group EDI ID qualifiers that you want EDI Module and Trading Networks to use for the group header sender and receiver.

Additionally, when you define the interchange-level sender/receiver pair information, you can specify information that services you create to form outbound EDI documents can use. Specifically, you can define the information to form the interchange headers for outbound EDI documents.

The EDI Module stores the interchange-level sender/receiver pair information that you define in the EDIEnvelope table, which is an EDI Module-specific table in the Trading Networks database.

To define interchange-level sender/receiver pair information, perform the following procedure for each interchange-level sender/receiver pair that you expect in the EDI documents that you want the EDI Module to process.



Tip! If you want to define information for an interchange sender/receiver pair that is similar to a pair you have already defined, you can copy the information for the existing pair and modify it. For instructions, see “[Editing and Deleting Interchange-Level Sender/Receiver Pair Information](#)” on page 403.

To define information for an interchange-level sender/receiver pair

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **Partner Set Up** menu of the navigation panel, click **Manage Interchange Info**.
- 4 Click **Add Interchange Information**.
- 5 In the **ID Qualifiers** section of the screen, specify the following to identify a specific interchange sender/receiver pair:

For this ID Qualifiers field...	Specify...
Sender ID	The sender ID for the interchange-level sender. For example, if the sender is identified with a D-U-N-S number in the interchange header, specify the sender's D-U-N-S number.
Sender Qualifier	The EDI ID qualifier that corresponds to the sender ID. For example, 01 for a D-U-N-S number.
Receiver ID	The receiver ID for the interchange-level receiver.
Receiver Qualifier	The EDI ID qualifier that corresponds to the receiver ID.

- 6 In the **Envelope Information** section of the screen, specify the production mode of the documents being exchanged between the trading partners. Select **Production**, **Testing**, or **Custom**.

- 7 In the **Inbound Information** section of the screen, specify the following information that the EDI Module uses when processing an inbound EDI document:

For this Inbound Information field...	Specify...
Create Doc	Whether you want the EDI Module to save the interchange document to the Trading Networks database when processing a document from the interchange sender/receiver pair defined in the ID Qualifiers section of the screen. Specify either Yes or No. Note: If you select to save the document, Trading Networks sets both the sender and receiver that it associates with the saved documents as Unknown. Trading Networks is unable to associate the actual sender and receiver because that would require the sender and receiver to have a profile, and when you use non-standard processing, you do not create profiles for interchange-level senders and receivers.
GS Sender Qualifier	The EDI ID qualifier that corresponds to the sender value on the group header in the document. Specify * (asterisk) if you want to use the EDI ID qualifier from the interchange header.
GS Receiver Qualifier	The EDI ID qualifier that corresponds to the receiver value on the group header in the document. Specify * (asterisk) if you want to use the EDI ID qualifier from the interchange header.

- 8 In the **Inbound Information - Control Number Validation** section of the screen, specify the following settings that the EDI Module uses when validating interchange control numbers in an inbound EDI document. For more information about how these settings are used, see “[Trading Networks Attributes and EDI Documents](#)” on page 196 in Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”.

Note: To set up control number validation for group control numbers, use the *ControlNumberManagement* EDITPA variables, which are described starting with “[wm.b2b.editn.TPA:EDITPA IS Document Type](#)” on page 388.

For this Inbound Control Number Validation field...	Specify...
Validate inbound envelope control numbers	Whether you want the EDI Module to verify the control numbers in the interchange header when processing a document from the interchange sender/receiver pair defined in the ID Qualifiers section of the screen. Specify either Yes or No.
Duplicate control number action	The action you want the EDI Module to take when it encounters a duplicate control number in an interchange header. <u>Select...</u> <u>To have the EDI Module</u>
Error & Continue	The EDI Module logs the error; then continues to process the EDI document that contains the invalid control number normally.
Process Normally	The EDI Module logs a warning; then continues to process the EDI document that contains the invalid control number normally.

For this Inbound Control Number Validation field...	Specify...
Reject	<p>The EDI Module logs the error and does <i>not</i> process the document normally. The EDI Module does <i>not</i> split the EDI document. Typically, the EDI Module splits an inbound EDI based on the EDITPA <i>splitOption</i> variable and sends the documents it splits out to Trading Networks for processing. However, if you select Reject, the EDI Module sends the document without splitting it to Trading Networks processing rules.</p> <p>Additionally, the EDI Module set the Trading Networks custom attribute EDI Status as follows:</p> <ul style="list-style-type: none">■ For a duplicate control number, sets the custom attribute EDI Status to Duplicate Control Number.■ For an out-of-sequence control number, sets the custom attribute EDI Status to Out of Sequence Control Number. Note that this is for the Out of sequence control number action field that is described below. <p>You can use the custom attribute EDI Status in processing rule criteria. You should create a processing rule to handle this rejected document. For information, see “Defining Processing Rules to Handle Documents with Invalid Control Numbers” on page 247.</p>
Out of sequence control number action	<p>The action you want the EDI Module to take when it encounters an out-of-sequence control number in an interchange header. Select Error & Continue, Process Normally, or Reject. For a description of these settings, see the descriptions of the settings above for Duplicate control number action.</p> <p>You can later force processing of the out-of-sequence document if you want. For more information, see “Reprocessing EDI Documents with Invalid Control Numbers” on page 250 in Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”.</p>

- 9 In the **Inbound Information - FA Generation** section of the screen, specify the following information that the EDI Module uses to determine whether it should automatically generate functional acknowledgements (FAs) for inbound EDI document and settings for generating the FAs. For more information about automatic FA generation, see “[Automatically Generating Functional Acknowledgements](#)” on page 258 in Chapter 15, “[Optional Inbound Processing When Using Trading Networks](#)”.

For this Inbound Information - FA Generation field...	Specify...	
Auto Generate FA	Whether you want to turn automatic FA generation on. Select one of the following:	
	Set Auto Generate FA to:	To have the EDI Module:
	<input type="radio"/> On	Always automatically generate FAs.
	<input type="radio"/> Per Document	Automatically generate FAs based on the indicator flag in the interchange header (ISA14 or UNB09).
	<input type="radio"/> Off	Never automatically generate FAs.
	For more information, see “ Turning Automatic FA Generation On or Off ” on page 259.	
FA Level	The level of detail that you want the EDI Module to acknowledge in the FAs that it generates. Select one of the following:	
	Set FA Level to:	To have the EDI Module acknowledge at the:
	<input type="radio"/> Default	Envelope level (group for ANSI X12 and interchange for UN/EDIFACT)
	<input type="radio"/> TransactionSet	Transaction set level
	<input type="radio"/> Segment	Segment level
	<input type="radio"/> Element	Element level

Note: If you are generating FAs at the element level, be sure to configure the maximum number of errors to report per FA transaction. For more information, see “[Configuring the Maximum Number of Transaction Errors](#)” on page 58.

For more information, see “[Variables that Affect How the EDI Module Generates the FA](#)” on page 260.

<u>For this Inbound Information - FA Generation field...</u>	<u>Specify...</u>
Process Document	<p>How you want the EDI Module to process a transaction, group, or UN/EDIFACT interchange based on its FA status. Use this field to define the FA statuses that are acceptable and unacceptable.</p> <ul style="list-style-type: none"> ■ For acceptable FA statuses, the EDI Module processes a transaction, group, or UN/EDIFACT interchange using its normal processing. ■ For unacceptable FA statuses, the EDI Module performs different processing. <p>Select one of the following:</p>
<u>Set Process Document to:</u>	<u>If you want:</u>
All	<ul style="list-style-type: none"> ■ Acceptable FA statuses: all statuses ■ Unacceptable FA statuses: no FA statuses are unacceptable
Only Accepted	<ul style="list-style-type: none"> ■ Acceptable FA statuses: Accepted ■ Unacceptable FA statuses: <ul style="list-style-type: none"> ■ Not Allowed ■ Rejected ■ Partially Accepted ■ Accepted, But Errors Were Noted
Not Rejected	<ul style="list-style-type: none"> ■ Acceptable FA statuses: <ul style="list-style-type: none"> ■ Not Allowed ■ Partially Accepted ■ Accepted, But Errors Were Noted ■ Accepted ■ Unacceptable FA statuses: Rejected

For more information, see “[Actions the EDI Module Takes Based on FA Status](#)” on page 272.

<u>For this Inbound Information - FA Generation field...</u>	<u>Specify...</u>
Generate Control Number	How the EDI Module is to generate the control numbers that is uses in the interchange and group headers of the FAs that is automatically generates. Select one of the following:
Set Generate Control Number to:	To have the EDI Module:
From Inbound Document	Use the control numbers from the corresponding headers of the inbound EDI document that the FA acknowledges.
Random	Randomly generate control numbers for the interchange and group headers of the FA.
From Control Number Table	Obtain the control numbers from the EDIControlNumber table.
For more information, see “Variables that Affect How the EDI Module Generates the FA” on page 260 .	
Syntax Error Status	How you want the EDI Module to report the syntax error status for a transaction, group, or UN/EDIFACT interchange. The EDI Module uses the syntax error status along with the logical error status and child transaction rejected status (if applicable) to determine the FA status for a transaction, group, or UN/EDIFACT interchange.
Set Syntax Error Status to:	To have the EDI Module:
Rejected	Report the syntax error status as “Rejected” if syntax errors are encountered. Specify Rejected if you want to reject elements that have syntax errors.
Accepted, But Errors Were Noted	Report the syntax error status as “Accepted, But Errors Were Noted” if syntax errors are encountered. Specify Accepted, But Errors Were Noted if you want to know whether there are syntax errors, but do not want to reject an element because of them.

For this Inbound Information - FA Generation field...	Specify...	
	Accepted	Report the syntax error status as “Accepted” regardless of any syntax errors that might be encountered. Specify Accepted if you do not want to check for syntax errors.
Logical Error Status	Set Logical Error Status to:	To have the EDI Module:
	Rejected	Report the logical error status as “Rejected” if logical errors are encountered. Specify Rejected if you want to reject elements that have logical errors.
Accepted, But Errors Were Noted	Accepted	Report the logical error status as “Accepted, But Errors Were Noted” if logical errors are encountered. Specify Accepted, But Errors Were Noted if you want to know whether there are logical errors, but do not want to reject an element because of them.
	Accepted	Report the logical error status as “Accepted” regardless of any logical errors that might be encountered. Specify Accepted if you do not want to check for logical errors.
For more information, see “ Syntax Error Status ” on page 264.		
For more information, see “ Logical Error Status ” on page 265.		

<u>For this Inbound Information - FA Generation field...</u>	<u>Specify...</u>
<u>Child Transaction Rejected Status</u>	<p>How you want the EDI Module to report the child transaction rejected status for a group or UN/EDIFACT interchange. The child transaction rejected status indicates whether child elements of a group or UN/EDIFACT interchange have an FA status of "Rejected".</p> <p><u>Set Child Transaction Rejected Status to:</u></p>
<u>Rejected</u>	<p><u>To have the EDI Module:</u></p> <p>Report the child transaction rejected status as:</p> <ul style="list-style-type: none"> ■ "Rejected" if the FA status of any of the child transactions is either "Rejected" or "Accepted, But Errors Were Noted". ■ "Accepted" if the FA statuses of all the child transactions are "Accepted".
<u>Partially Accepted</u>	<p>Report the child transaction rejected status as:</p> <ul style="list-style-type: none"> ■ "Rejected" if the FA statuses of all of the child transactions are "Rejected". ■ "Partially Accepted" if the FA status of at least one child transaction is "Accepted", but the FA status of other child transactions are "Rejected" and/or "Accepted, But Errors Were Noted". ■ "Accepted, But Errors Were Noted" if the FA statuses of the child transactions are either "Rejected" and/or "Accepted, But Errors Were Noted" -and- no child transactions are "Accepted". ■ "Accepted" if the FA statuses of all the child transactions are "Accepted".

For this Inbound Information - FA Generation field...	Specify...
Accepted, But Errors Were Noted	<p>Report the child transaction rejected status as:</p> <ul style="list-style-type: none"> ■ “Rejected” if all the child transactions are “Rejected”. ■ “Accepted, But Errors Were Noted” if the FA statuses of the child transactions are “Rejected”, “Accepted, But Errors Were Noted”, and “Accepted”. ■ “Accepted” if the FA statuses of all the child transactions are “Accepted”. <p>For more information, see “Child Transaction Rejected Status on page 267.</p>

- 10 In the Outbound Information - FA Generation section of the screen, specify the following information that the EDI Module uses to for the sender and receiver in the BizDocEnvelope that it creates for the FA. When defining a Trading Networks processing rule to deliver the FA, you use the BizDocEnvelope information when defining the processing rule criteria.

For this Outbound Information - FA Generation field...	Specify...				
Sender	The trading partner that you want to identify as the sender of the FA for Trading Networks processing.				
Receiver	The trading partner that you want to identify as the receiver of the FA for Trading Networks processing.				
Add Group	<p>Whether to add a group to the FA.</p> <p>Select... <u>To have the EDI Module</u></p> <table> <tr> <td>Yes</td><td>Add a group to the FA.</td></tr> <tr> <td>No</td><td>Not add a group to the FA. This is the default.</td></tr> </table>	Yes	Add a group to the FA.	No	Not add a group to the FA. This is the default.
Yes	Add a group to the FA.				
No	Not add a group to the FA. This is the default.				

<u>For this Outbound Information - FA Generation field...</u>	<u>Specify...</u>
Control Number With Leading Zero	Whether to pad control numbers in the FA with leading zeros.
Yes	Pad control numbers in the FA with leading zeros.
No	Do not pad control numbers with leading zeros.

- 11 In the **Outbound Information - Delimiters** section of the screen, specify the following information that a service you create to form outbound EDI documents can use:

<u>For this Outbound Information - Delimiters field...</u>	<u>Specify...</u>
Segment	The segment terminator for the outbound EDI document (e.g., "+"). The default is the new line character.
Field	The field separator for each EDI segment (e.g., !). The default is the "*" character.
SubField	The separator for composite elements (e.g, *). The default is the ":" character.
Release	The release character for the outbound EDI document (e.g., "?").
Decimal	The decimal separator to use in the outbound EDI document. If you want the outbound document to use the European format, specify the "," character. For example, using the European format a number would be formatted as 100,10 (European format) instead of 100.10, as is common in the US.

- 12 In the Outbound Information - Envelope Information section of the screen, specify the following information that a service you create to form outbound EDI documents can use to form the interchange headers:

<u>For this Outbound Information - Envelope Information field...</u>	<u>Specify...</u>
ISA01	The value to use for the ISA01 element of an ANSI X12 interchange header in the outbound EDI document.
ISA02	The value to use for the ISA02 element of an ANSI X12 interchange header in the outbound EDI document.
ISA03	The value to use for the ISA03 element of an ANSI X12 interchange header in the outbound EDI document.
ISA04	The value to use for the ISA04 element of an ANSI X12 interchange header in the outbound EDI document.
ISA011	The value to use for the ISA11 element of an ANSI X12 interchange header in the outbound EDI document.
ISA012	The value to use for the ISA12 element of an ANSI X12 interchange header in the outbound EDI document.
ISA014	The value to use for the ISA14 element of an ANSI X12 interchange header in the outbound EDI document.
UNB01	The value to use for the UNB01 element of a UN/EDIFACT UNB header in the outbound EDI document.
UNB06	The value to use for the UNB06 element of a UN/EDIFACT UNB header in the outbound EDI document.
UNB07	The value to use for the UNB07 element of a UN/EDIFACT UNB header in the outbound EDI document.
UNB08	The value to use for the UNB08 element of a UN/EDIFACT UNB header in the outbound EDI document.
UNB09	The value to use for the UNB09 element of a UN/EDIFACT UNB header in the outbound EDI document.
UNB10	The value to use for the UNB10 element of a UN/EDIFACT UNB header in the outbound EDI document.
UNB11	The value to use for the UNB11 element of a UN/EDIFACT UNB header in the outbound EDI document.

- 13 Click **Save Changes**. The EDI Module adds the information you defined for the interchange sender/receiver pair identified in the ID Qualifiers section of the screen to the EDIEnvelope table in the Trading Networks database.

Editing and Deleting Interchange-Level Sender/Receiver Pair Information

At times you might want to edit, delete, or copy information of a defined interchange-level sender/receiver pair. To display the information with which you want to work, you first search for the interchange-level sender/receiver pair to have the EDI Module display its information. Then you can edit, delete, or copy the information.

To edit existing Interchange-level sender/receiver pair information

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **Partner Set Up** menu of the navigation panel, click **Manage Interchange Info**.
- 4 To search for the information with which you want to work, specify one or more of the following criteria:

For this field...	Specify...
Sender ID	The sender ID of the interchange-level sender/receiver pairs that you want to display.
Sender Qualifier	The EDI ID qualifier of the sender of the interchange-level sender/receiver pairs that you want to display.
Receiver ID	The receiver ID of the interchange-level sender/receiver pairs that you want to display.
Receiver Qualifier	The EDI ID qualifier of the receiver of the interchange-level sender/receiver pairs that you want to display.
Production Mode	The production mode specified in the information for the interchange-level sender/receiver pairs that you want to display.
Sort By	How you want the EDI Module to sort the returned list of interchange-level sender/receiver pairs.
Maximum Results	The maximum number of matching interchange-level sender/receiver pairs you want the EDI Module to return.

- 5 Click **Search**. The EDI Module displays the list of interchange-level sender/receiver pairs that match your criteria at the bottom of the screen.

- 6 You can edit, delete, or copy information for any of the listed interchange sender/receiver pairs.
 - To edit information for an interchange sender/receiver pair, click  in the Edit column. The EDI Module displays the information for the interchange sender/receiver pair. You can edit fields as necessary.
 - To delete information for an interchange sender/receiver pair, click  in the Delete column.
 - To copy information for an interchange sender/receiver pair, click  in the Copy column. This enables you to define information for a new interchange sender/receiver pair that uses most of the same information as the original interchange sender/receiver pair. The EDI Module displays the information for the new interchange sender/receiver pair. You can edit fields as necessary.
- 7 If you edited or made a copy of information for an interchange sender/receiver pair, click Save Changes to complete the procedure.

Defining Group-Level Sender/Receiver Pair Associations

You define group-level sender/receiver pair associations to associate sender/receiver pairs at the group level with a specific interchange-level sender/receiver pair. For example, group-level sender A and receiver B are associated with interchange-level sender X and receiver Y.

Group sender	Group receiver	Interchange sender	Interchange receiver
A	B	X	Y

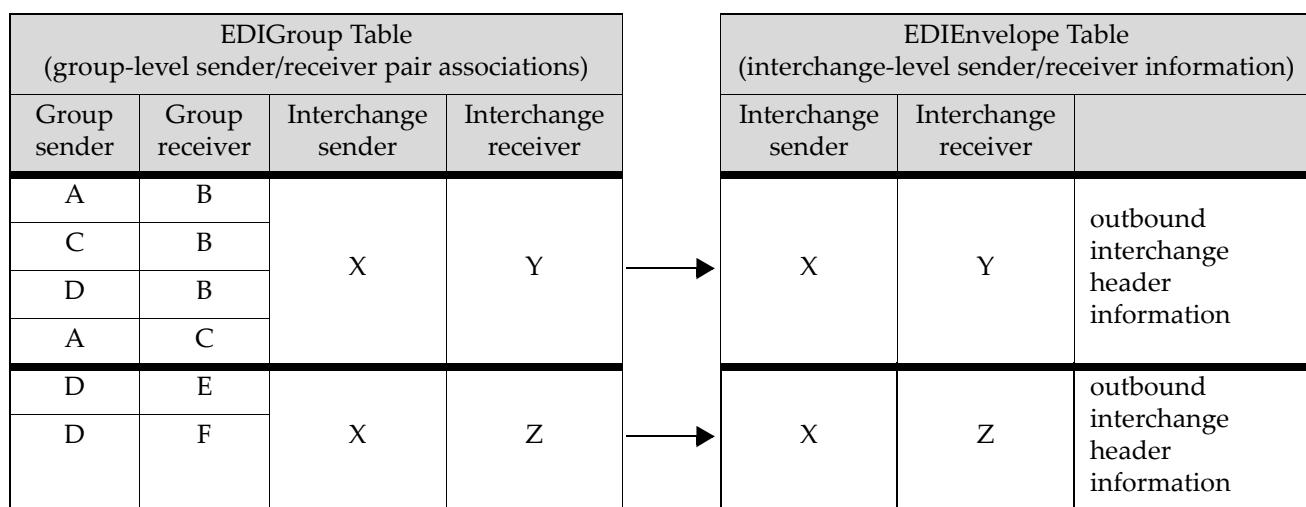
When you define group-level sender/receiver pair associations, you must define all the group-level sender/receiver pairs that are associated with a single interchange-level sender/receiver pair. A group-level sender/receiver pair can be associated with only one Interchange-level sender/receiver pair.

Group sender	Group receiver	Interchange sender	Interchange receiver
A	B	X	Y
C	B		
D	B		
A	C	X	Z
D	E		
D	F		

These associations are for use by services that you create to form outbound EDI documents. Because you are using non-standard processing, your service will have the group-level senders and receivers. However, to create an interchange header for the outbound EDI document, your service needs the interchange-level sender/receiver pair. For your service to make a correlation between group-level sender/receiver to interchange-level sender/receiver, it uses the group-level sender/receiver pair associations that you define. For example, if your service has group-level sender A and receiver B, using the group-level sender/receiver pair associations, it can determine that the associated interchange-level sender/receiver pair is interchange-level sender X and receiver Y.

After your service retrieves the interchange-level sender/receiver pair information, your service can create the interchange headers by using the information you defined for the interchange sender/receiver pair.

The EDI Module saves the group-level sender/pair associations in the EDIGroup table, which is an EDI Module-specific table in the Trading Networks database. The EDIGroup table works together with the EDIEvelope table, which contains the interchange-level sender/receiver pair information that is described in [“Defining Interchange-Level Sender/Receiver Pair Information” on page 390](#). The following graphic illustrates how the EDIEvelope and EDIGroup tables work together.



Use the following procedure to associate group-level sender/receiver pairs with a specific interchange-level sender/receiver pair. You must define the information for the interchange-level sender/receiver pair *before* you can associate group-level sender/receiver pairs with it. For instructions for defining interchange-level sender/receiver pairs, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 390](#).

To define Group-level sender/receiver pair associations

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **Partner Set Up** menu of the navigation panel, click **Manage Interchange Info**.
- 4 To search for the interchange-level sender/receiver pair with which you want to associate group-level sender/receiver pairs, specify one or more of these criteria:

For this field...	Specify...
Sender ID	The sender ID of the interchange-level sender/receiver pairs that you want to display.
Sender Qualifier	The EDI ID qualifier of the sender of the interchange-level sender/receiver pairs that you want to display.
Receiver ID	The receiver ID of the interchange-level sender/receiver pairs that you want to display.
Receiver Qualifier	The EDI ID qualifier of the receiver of the interchange-level sender/receiver pairs that you want to display.
Production Mode	The production mode specified in the information for the interchange-level sender/receiver pairs that you want to display.
Sort By	How you want the EDI Module to sort the returned list of interchange-level sender/receiver pairs.
Maximum Results	The maximum number of matching interchange-level sender/receiver pairs you want the EDI Module to return.

- 5 Click **Search**. The EDI Module displays the list of interchange-level sender/receiver pairs that match your criteria at the bottom of the screen.
- 6 In the row for the interchange-level sender/receiver pair to which you want to associate group-level sender/receiver pairs, click  in the **GS** column. The EDI Module displays the **Cross Reference > GS Pairs** screen that lists the interchange-level sender/receiver pair and any existing group-level sender/receiver pairs.
- 7 For each group-level sender/receiver pair that you want to associate with the interchange-level sender/receiver pair, perform the following:
 - a Click **Add GS Pair**. The EDI Module displays the **Add GS Pairs** screen.

- b Specify the following about the group-level sender/receiver pair:

<u>For this field...</u>	<u>Specify</u>
Sender ID	The sender ID for the group-level sender.
Sender Qualifier	The EDI ID qualifier for the group-level sender.
Receiver ID	The receiver ID for the group-level receiver.
Receiver Qualifier	The EDI ID qualifier for the group-level receiver.

- c Click the **Save Changes** button.

Querying for Group-Level Sender/Receiver Pair Associations

At times you might want to view the group-level sender/receiver pair associations, information, for example to determine whether you already have created an association for a group-level sender/receiver pair. To query, perform the following procedure.

To query for existing group-level sender/receiver pair associations

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDI**. The Server Administrator opens a new browser window to display the EDI Module home page.
- 3 From the EDI Module home page, in the **Partner Set Up** menu of the navigation panel, click **Query Envelope IDs**.
- 4 To search for the group-level sender/receiver pair that you want to view, specify one or more of the following criteria:

<u>For this field...</u>	<u>Specify</u>
ID	The sender ID or receiver ID for a group-level sender or receiver.
Qualifier	The EDI ID qualifier associated with either the group-level sender or receiver.
Sort By	How you want the EDI Module to sort the returned list of group sender/receiver pairs.
Maximum Results	The maximum number of matching group-level sender/receiver pairs you want the EDI Module to return

- 5 Click **Search**. The EDI Module displays the list of group-level sender/receiver pairs that match your criteria at the bottom of the screen.

Defining Control Number Information for Partners



Important! This section describes the areas of defining control number information that is different when you are using non-standard processing. It does *not* contain all information about defining control number information for partners. You should also read the information provided in [Chapter 11, “Defining Control Number Information for Partners”](#).

Turning Inbound Control Number Validation On or Off

When using non-standard processing, the way you turn validation on or off is different for interchange control numbers and group control numbers:

Type of Control Number	How to turn validation on or off
Interchange	Use the Validate inbound envelope control numbers setting on the EDI Module home page. For more information about how to define this setting, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 390 .
Group	<p>Use the EDITPA variable <code>ControlNumberManagement/validateInboundGroupControlNumbers</code>. To turn group control number validation:</p> <ul style="list-style-type: none"> ■ On, set this EDITPA variable to <code>true</code>. ■ Off, set this EDITPA to <code>false</code>. <p>For more information, see “ControlNumberManagement/validateInboundGroupControlNumbers EDITPA Variable” on page 125.</p>

Defining Actions for Invalid Control Numbers

When using non-standard processing, you specify the actions to take for invalid interchange and group control number in different ways as described in the table below. For a list of the actions you can specify and a description of each, see [“Defining Actions for Invalid Control Numbers” on page 170](#) in [Chapter 11, “Defining Control Number Information for Partners”](#).

Type of control number	Duplicate or Out-of-sequence	How to define the action
Interchange	Duplicate	Use the Duplicate control number action setting on the EDI Module home page. For instructions for defining this setting, see “ Defining Interchange-Level Sender/Receiver Pair Information ” on page 390 .
	Out-of-sequence	Use the Out of sequence control number action setting on the EDI Module home page. For instructions for defining this setting, see “ Defining Interchange-Level Sender/Receiver Pair Information ” on page 390 .
Group	Duplicate	Use the EDITPA variable ControlNumberManagement/duplicateControlNumberAction . For a list of the settings for this EDITPA variable, see “ ControlNumberManagement/duplicateControlNumberAction EDITPA Variable ” on page 125 .
	Out-of-sequence	Use the EDITPA variable ControlNumberManagement/outOfSequenceControlNumberAction . For a list of the settings for this EDITPA variable and a description of the processing the EDI Module performs for each, see “ ControlNumberManagement/useReverseRouting EDITPA Variable ” on page 127 .

Defining the Control Number Settings



Important! This section does *not* provide the complete procedure for defining control number settings. It describes the different information you must provide when using non-standard processing. For the complete procedure, see “[Defining Control Number Settings](#)” on [page 177](#) in Chapter 11, “[Defining Control Number Information for Partners](#)”.

The control number settings are described in “[Control Number Cap, Minimum, Increment, and Window](#)” on [page 172](#) in Chapter 11, “[Defining Control Number Information for Partners](#)”.

You define these settings for a unique combination of:

- sender/receiver pair
- production mode (e.g., Testing for Production)

- EDI standard and version
- type, that is, “Envelope” or group type (e.g., PO)

When defining the settings and you are using non-standard processing, you must supply different values for the sender/receiver pair that you define in the **ID Pair Information** section of the **Add Control Number** screen. For non-standard processing, supply the following for the **ID Pair Information** fields listed below:

For this ID Pair Information field...	Specify
Sender ID	The sender ID for the interchange or group.
Sender Qualifier	The EDI ID qualifier associated with the sender ID, e.g., 01 to identify a D-U-N-S number.
Receiver ID	The receiver ID for the interchange or group.
Receiver Qualifier	The EDI ID qualifier associated with the receiver ID, e.g., 01 to identify a D-U-N-S number.

Processing Inbound EDI Documents using Trading Networks



Important! This section describes how you need to set up items for inbound processing differently when using non-standard processing. It does *not* contain all information about inbound processing or setting up inbound processing. You should also read the information provided in [Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”](#).

Before You Can Process Inbound EDI Documents

In addition to the items listed in [“Before You Can Process Inbound EDI Documents”](#) on page 195 in [Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”](#), when using non-standard processing you must also define the interchange sender/receiver pair information. For instructions on defining this information, see [“Defining Interchange-Level Sender/Receiver Pair Information”](#) on page 390.

Variables that Affect Inbound Processing

When using non-standard processing, you must define the information for interchange/sender receiver pairs that is described in the table below. You define this information using the EDI Module home page. For more information about these settings including how to define them, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 390](#).

In addition to the information described in this section, you must also define EDITPA variables for group sender/receiver pairs. For information about the EDITPA variables that you must define, see [“Specifying EDITPA Variables that Affect Inbound Processing” on page 196](#) in Chapter 13, “Processing Inbound ANSI X12 and UN/EDIFACT Documents using Trading Networks”.

Interchange Sender/Receiver Pair Inbound Information Setting	Description
Create Doc	Whether you want the EDI Module to save the Interchange document to the Trading Networks database.
GS Sender Qualifier	The EDI ID qualifier to use with the sender IDs on the group headers.
GS Receiver Qualifier	The EDI ID qualifier to use with the receiver IDs on the group headers.
Validate inbound envelope control numbers	Whether you want the EDI Module to verify the control numbers in the interchange header of the interchange segment that it is processing.
Duplicate control number action	The action that you want the EDI Module to take when it encounters a duplicate control number in the interchange header of the interchange segment that it is processing.
Out of sequence control number action	The action that you want the EDI Module to take when it encounters an out-of-sequence control number in the interchange header of the interchange segment that it is processing.
Auto Generate FA	Whether you want the EDI Module to automatically generate FAs for an inbound EDI document. If you specify On or Per Document for Auto Generate FA, specify the other variables in the Inbound Information - FA Generation section.

Forming EDI Documents to Send Outbound When using Trading Networks



Important! This section describes how you need to set up items for outbound processing differently when using non-standard processing. It does *not* contain all information about outbound processing or setting up outbound processing. You should also read the information provided in [Chapter 16, “Forming EDI Documents to Send Outbound When Using Trading Networks”](#).

Before Creating the Service to Form an EDI Document

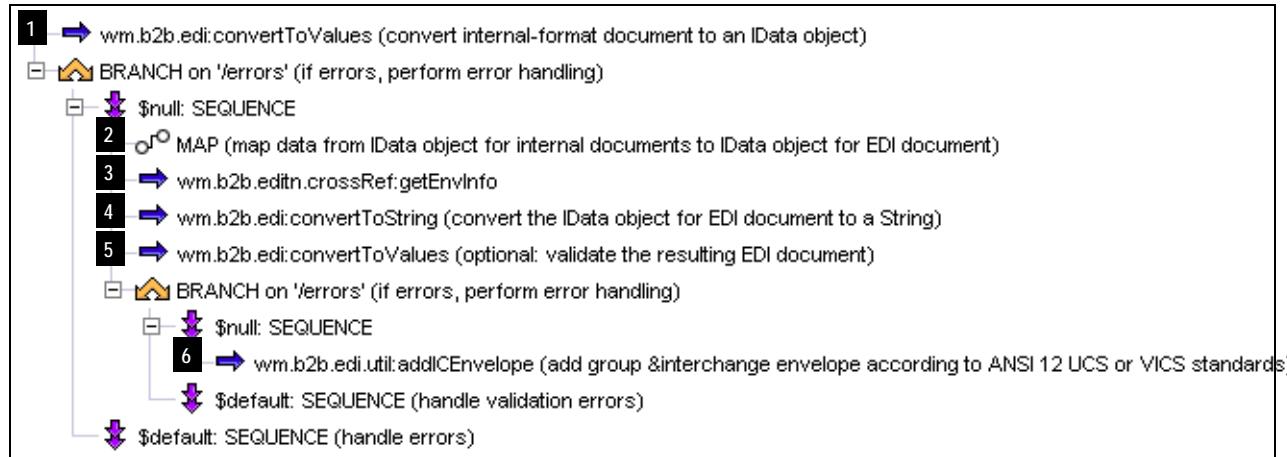
In addition to the items listed in [“Before Creating the Service to Form an EDI Document” on page 285](#) in [Chapter 16, “Forming EDI Documents to Send Outbound When Using Trading Networks”](#), when using non-standard processing you must also define:

- Interchange sender/receiver pair information. For instructions on defining this information, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 390](#).
- Group-level sender/receiver pair associations. For instructions on defining this information, see [“Defining Group-Level Sender/Receiver Pair Associations” on page 404](#).

Creating the Service When Using Non-Standard Processing

The following shows sample code that illustrates how to form an EDI document from an internal-format document when using Trading Networks with the EDI Module and when using non-standard processing.

Sample code for forming an EDI document when using non-standard processing



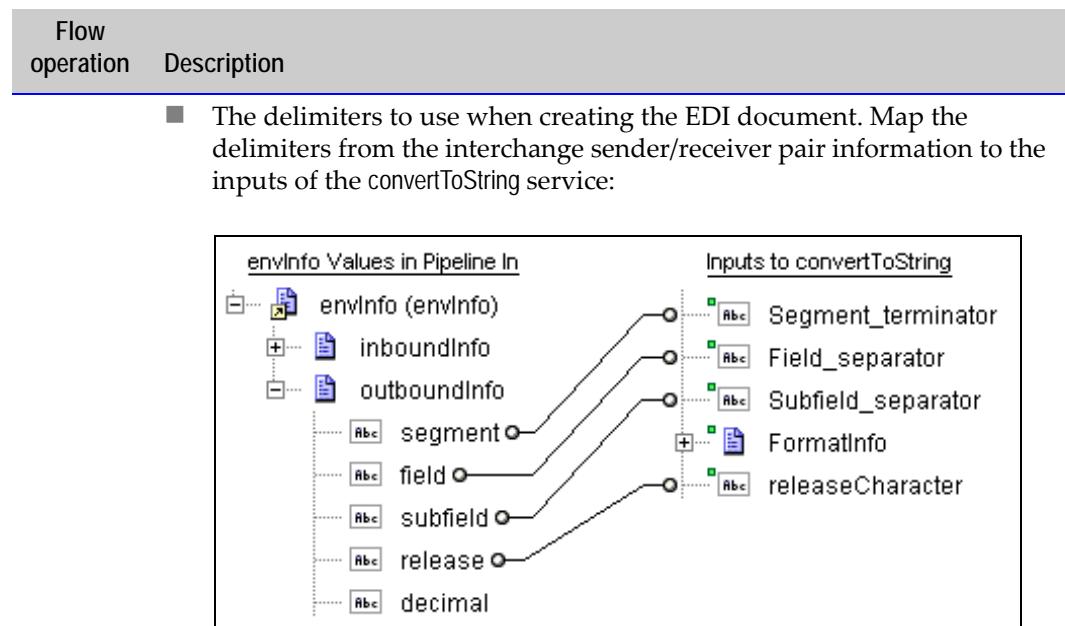
Flow operation	Description
1	<p>Invoke the <code>wm.b2b.edi:convertToValues</code> service to convert the incoming internal-format document that is either a String or InputStream into an <code>IData</code> object. If you want, you can set the input variables of the <code>convertToValues</code> service to have it validate the structure of the internal-format document.</p> <p>The inputs to the <code>convertToValues</code> service include the internal-format document and the flat file schema that defines the structure for the internal-format document. For backward compatibility, you can use an IS document type to define the structure of internal-format document rather than a flat file schema. However, it is recommended that you use flat file schemas. For more information about the <code>convertToValues</code> service, see the <i>webMethods EDI Module Built-In Services Reference</i>.</p>
	<p>Note: If the internal-format document is passed to your service as an <code>IData</code> object, you can still validate its structure before forming the EDI document. See “Validating the Input Internal-Format Document When it is an <code>IData</code> Object” on page 91.</p>

Flow operation	Description
2	<p>Map data from the internal-format document IData object into the EDI document IData object. Depending on the complexity of your mapping requirements, you might need to add more logic than a MAP flow operation, or create a separate service to perform the mapping. For more information about how to map, see Chapter 5, “Mapping Data to Form New Documents”.</p>
3	<p>Invoke the <code>wm.b2b.editn.crossRef:getEnvInfo</code> service to obtain the delimiters and header information you can use for the outbound EDI document. You pass this service the group sender and receiver and it locates the information for the associated interchange sender/receiver.</p> <ul style="list-style-type: none"> ■ To find the interchange sender/receiver pair, the service uses the information you defined in “Defining Group-Level Sender/Receiver Pair Associations” on page 404. ■ The service returns the information for the interchange sender/receiver that you defined in “Defining Interchange-Level Sender/Receiver Pair Information” on page 390.
4	<p>Invoke the <code>wm.b2b.edi:convertToString</code> service to convert the EDI document from an IData object to String format.</p>

The inputs to the convertToString service include:

- The IData object that contains the data for your EDI document. Map this IData object to the input variable *Values* of the convertToString service.
- The flat file schema for the EDI document. The convertToString service uses the flat file schema to determine how to form the EDI document.

For backward compatibility, you can use an IS document type as input to the convertToString service rather than a flat file schema for files with delimited fields and records.



- 5** Optionally, invoke the `wm.b2b.edi:convertToValues` service against the EDI document to validate the structure of your final EDI document.

The inputs to the `convertToValues` service include:

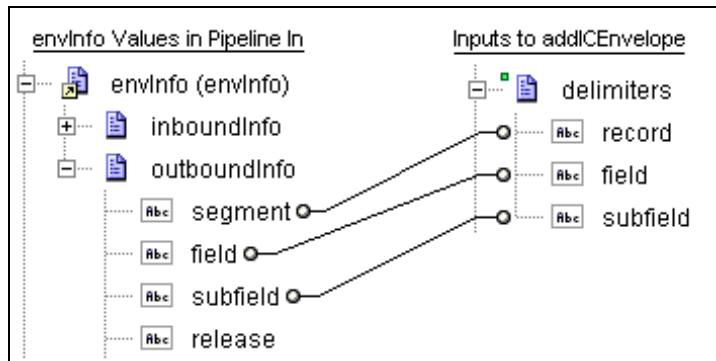
- The EDI document. The output variable `string` from the `convertToString` service contains the EDI document. Map this to the input variable `edidata` of the `convertToValues` service.
- The flat file schema that defines the structure for the EDI document.

- 6** Invoke the `wm.b2b.edi.util:addICEnvelope` service to add the group and interchange envelope to the EDI document. If you are creating a UN/EDIFACT EDI document, use the `wm.b2b.edi.util:addICEnvelopeEDIFACT` service. For more information, see [“Adding UN/EDIFACT Envelopes” on page 91](#).

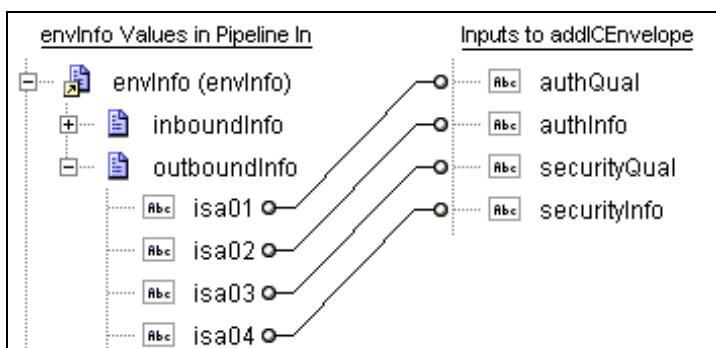
Flow operation	Description
----------------	-------------

When setting the inputs to the addICEnvelope:

- Map the delimiters from the interchange sender/receiver pair information to the inputs of the addICEnvelope service:



- Map the information from the interchange sender/receiver pair to the inputs of the addICEnvelope service:



- Set the *ctlFromTable* variable to true to have the service obtain the control number. See “[Obtaining Control Numbers for Outbound Processing \(ANSI X12 and UN/EDIFACT\)](#)” on page 290 for more information.

For information about delivering the document, see “[Delivering the EDI Document](#)” on page 291 in Chapter 16, “[Forming EDI Documents to Send Outbound When Using Trading Networks](#)”.

Using the 6.0.1 Version of the Batching Feature

■ Overview	418
■ The EDI Module Version 6.0.1 Batching Feature	418
■ Before You Can Batch EDI Documents	426
■ Controlling How the batchProcess Service Forms the Batch Document	427
■ Setting Up to Batch EDI Documents	430
■ Delivering the Batch EDI Document	437

Overview

The information provided in this appendix is for backward compatibility. It describes how to use the batching feature that was available in webMethods EDI Module (EDI Module) in version 6.0.1. webMethods recommends that you perform batching as described in [Chapter 17, “Batching EDI Documents”](#); however, the method described in this appendix is still supported.

The EDI batching feature provided with 6.0.1 requires that you set up more scheduled delivery queues than you need to set up for the recommended method that is described in [Chapter 17, “Batching EDI Documents”](#) and also results in more batch EDI documents. It can also require two passes through webMethods Trading Networks (Trading Networks) to create the batch EDI document.

The trigger that tells the EDI Module which way you want to perform batching is the *oneBatchQueue* input variable to the `wm.b2b.editn.batch:batchProcess` service. You define the *oneBatchQueue* input variable when you define the Trading Networks scheduled delivery queue for batching.

- If you set *oneBatchQueue* to `NONE`, the EDI Module uses backward compatibility to perform batch processing using the method supported in the 6.0.1 version of the EDI Module. This method is described in this appendix.
- If you set *oneBatchQueue* to `SINGLEOUTPUT` or `MULTIPLEOUTPUTS`, the EDI Module performs batching as described in [Chapter 17, “Batching EDI Documents”](#).

By default, the *oneBatchQueue* input variable is set to `NONE`.

To learn more about the recommended batching feature, see the section about batching in Chapter 3, "Using the EDI Module with Trading Networks" of the *webMethods EDI Module Concepts Guide*.

The EDI Module Version 6.0.1 Batching Feature

Non-standard

Important! You *cannot* use EDI batching feature if you are using non-standard processing. For more information about processing levels, see [“Using Standard or Non-Standard Processing” on page 110](#).

To batch the documents, the EDI Module provides the `wm.b2b.editn.batch:batchProcess` service. When you install the EDI Module, this service is registered as a Trading Networks delivery service and assigned the name EDI Batch in Trading Networks. The `batchProcess` service combines EDI documents into a single document and adds group-level and/or interchange-level headers and trailers to the document. To use the `batchProcess` service, you use Trading Networks scheduled delivery.

You can have the EDI Module create a batch EDI document in a single pass through Trading Networks or in two passes through Trading Networks.

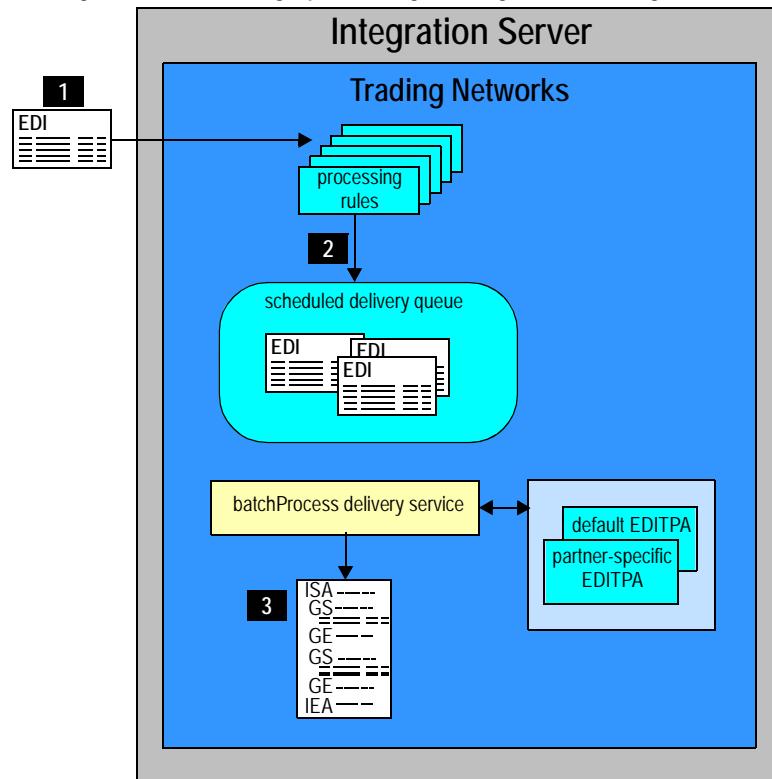
You will likely need to set up multiple queues to achieve the batch EDI documents you want to create. Each queue creates a batch document where all combined documents use a single version of the same EDI standard and all headers and trailers identify the same sender and receiver.

Batching the Documents in a Single Pass Through Trading Networks

For a single pass through Trading Networks, the batchProcess service combines all the documents in the queue into a single EDI document and adds both group and interchange headers and trailers (or just interchange headers and trailers for a UN/EDIFACT document). After a single pass through Trading Networks, the outbound EDI document is ready for delivery. The batchProcess service routes the batch EDI document back to Trading Networks. You define a Trading Networks processing rule that indicates how to deliver it.

The following diagram illustrates how to form a batch EDI document in a single pass through Trading Networks. For more information, see the table after the diagram.

Batching documents in a single pass through Trading Networks using version 6.0.1 batching



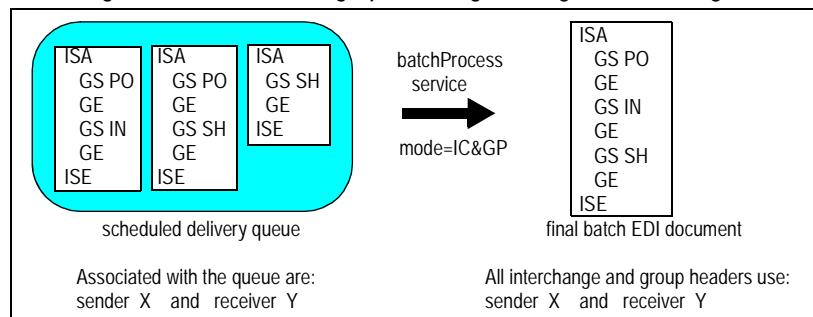
Step	Description
1	EDI document is sent to Trading Networks.
2	<p>Trading Networks uses its processing rules to determine how to process the EDI document. Trading Networks selects a processing rule that you create that uses the Deliver Document By processing action to deliver a document to a scheduled delivery queue associated with the batchProcess service.</p> <p>For more information about scheduled delivery and defining queues for scheduled delivery, see the <i>webMethods Trading Networks User's Guide</i>.</p>
3	<p>When the schedule that is associated with the queue indicates, Trading Networks invokes the batchProcess service to combine the EDI documents in the queue into a single EDI document. Because of the input variables that you set for the service, the service adds group and interchange headers and trailers to the final document. The batchProcess delivery service uses information from the EDITPA to form the headers and trailers. For more information about how the EDI documents are combined, see "How the Documents are Combined When Using a Single Pass" below.</p> <p>The batch EDI document is ready for delivery. For more information, see "Delivering the Batch Document" on page 426.</p>

How the Documents are Combined When Using a Single Pass

All EDI documents in the scheduled delivery queue will be combined with group envelopes and interchange envelopes added in a single pass. This is referred to as *IC&GP mode*. The batchProcess service combines the EDI documents in the queue, creating:

- One group envelope for each group type (e.g., PO) represented by the EDI documents in the queue. The transaction sets associated with the group type are included in the group envelope.
- A single interchange envelope.

Combining EDI documents in a single pass through Trading Networks using version 6.0.1 batching



After combining the EDI documents and adding group envelopes and an interchange envelope, the batchProcess service routes the final batch EDI document to Trading Networks processing rules. You need to define a processing rule that defines how to deliver the final batch document.

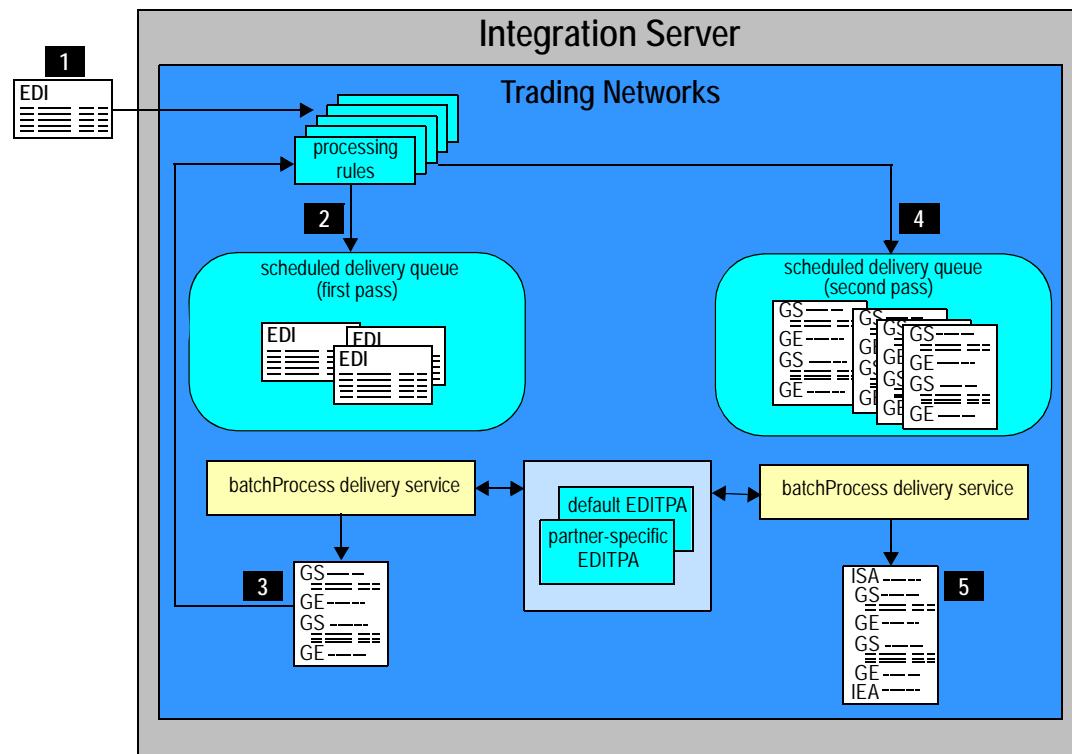
 Note: For more information about how you specify the sender and receiver that the batchProcess service uses in the group and interchange headers, see “[EDITPA Variables that the batchProcess Service Uses](#)” on page 429.

Batching the Documents in Two Passes Through Trading Networks

For two passes through Trading Networks, the batchProcess service combines all the documents in the queue into a single EDI document and adds only group headers and trailers. The batchProcess service then sends the document back to Trading Networks for a second pass. For the second pass, the document with group headers and trailers is sent to Trading Networks processing rules. Trading Networks selects another processing rule that you create to deliver the document into a different queue that also uses the batchProcess service. This second queue is to hold documents that group headers and trailers. When Trading Networks invokes the batchProcess service for this second queue, the batchProcess service combines the documents with group headers and trailers and adds interchange headers and trailers to the document to form the final outbound EDI document. The batchProcess service routes this final EDI document back to Trading Networks processing rules. You define a Trading Networks processing rule that indicates how to deliver the final EDI document.

The following diagram illustrates how to form a batch EDI document in two passes through Trading Networks. For more information, see the table after the diagram.

Batching documents in two passes through Trading Networks using version 6.0.1 batching



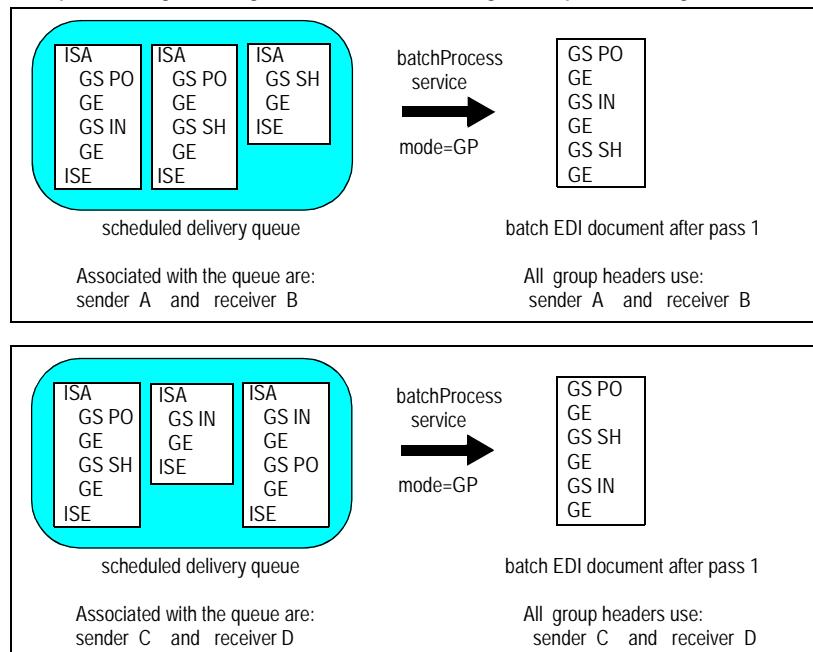
Step	Description
1	The EDI document is sent to Trading Networks.
2	Trading Networks uses its processing rules to determine how to process the EDI document. For the first pass through Trading Networks, Trading Networks selects a processing rule that you create that uses the Deliver Document By processing action to deliver a document to a scheduled delivery queue associated with the batchProcess service. For more information about scheduled delivery and defining queues for scheduled delivery, see the <i>webMethods Trading Networks User's Guide</i> .

Step	Description
3	<p>When the schedule that is associated with the queue indicates, Trading Networks invokes the batchProcess service to combine the EDI documents in the queue into a single EDI document. Because of the input variables that you set for the service, the service adds group headers and trailers to the final document. The batchProcess delivery service uses information from the EDITPA to form the headers and trailers. For more information about how the EDI documents are combined, see "How the Documents are Combined When Using Two Passes" on page 424.</p> <p>The batchProcess service then sends the EDI document it formed with group header and trailers back to Trading Networks.</p>
4	<p>Trading Networks uses its processing rules to determine how to process the EDI document. For the second pass through Trading Networks, Trading Networks selects a processing rule that you create that uses the Deliver Document By processing action to deliver a document to a different scheduled delivery queue that is also associated with the batchProcess service.</p>
5	<p>When the schedule that is associated with the queue indicates, Trading Networks invokes the batchProcess service to combine the EDI document in the queue into a single EDI document. Because of the input variables that you set for the service, the service adds the interchange headers and trailers to the document. The batchProcess service uses information from the EDITPA to form the headers and trailers. For more information about how the EDI documents are combined, see "How the Documents are Combined When Using Two Passes" on page 424.</p> <p>The final EDI document is ready for delivery. For more information, see "Delivering the Batch Document" below.</p>

How the Documents are Combined When Using Two Passes

You use two scheduled delivery queues, one for each pass through Trading Networks. In the first pass, you place EDI documents in a queue to add group envelopes. This is referred to as *GP mode*. The batchProcess service combines the EDI documents in the queue, creating one group envelope for each group type (e.g., PO) represented by the EDI documents in the queue. The transaction sets associated with the group type are included in the group envelope. Note that the following diagram shows two queues that each perform the first pass through Trading Networks.

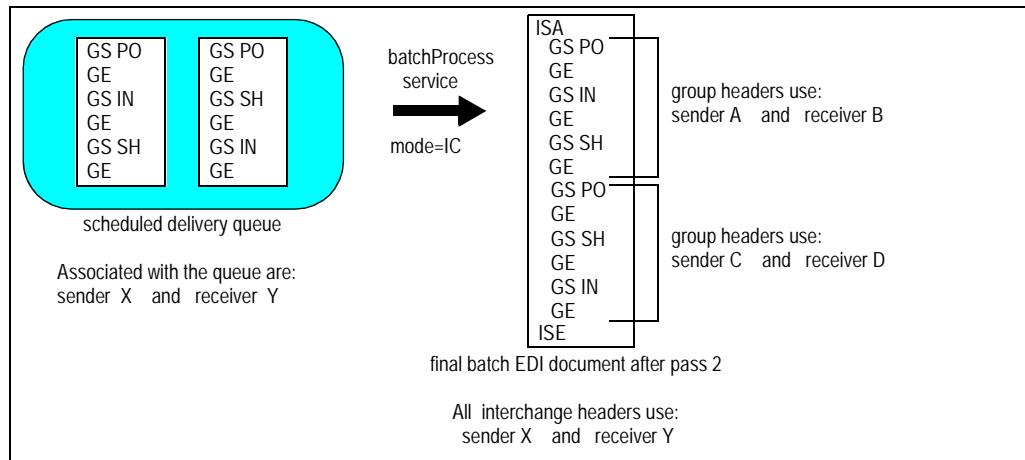
First pass through Trading Networks when batching in two passes using version 6.0.1 batching



After combining the EDI documents and adding group envelopes, the batchProcess service routes the document back to Trading Networks to select another processing rule that drops the newly formed documents into another (second) queue.

This second queue adds an interchange envelope. This is referred to as *IC mode*. The batchProcess service combines the EDI documents in the queue, creating one interchange envelope. Note that the following diagram uses the group documents created in the previous diagrams.

Second pass through Trading Networks when batching in two passes using version 6.0.1 batching



After combining the EDI documents and adding an interchange envelope, the batchProcess service routes the final batch EDI document to Trading Networks processing rules. You need to define a processing rule that defines the action to deliver the final batch document.

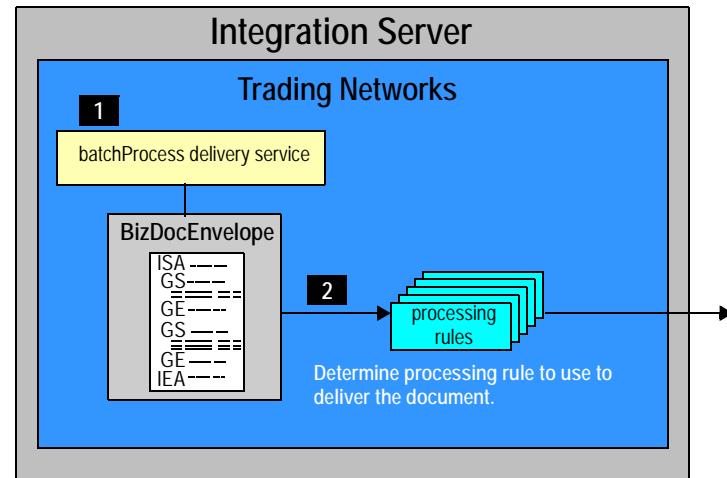


Note: For more information about how you specify the sender and receiver that the batchProcess service uses in the group and interchange headers, see “[EDITPA Variables that the batchProcess Service Uses](#)” on page 429.

Delivering the Batch Document

After the batchProcess service forms the final batch EDI document, it routes the document back to Trading Networks for delivery. The following diagram illustrates the process for delivering a batch EDI document. For more information, see the table after the diagram.

Delivering a batch EDI document using version 6.0.1 batching



Step	Description
1	The batchProcess service forms the final outbound EDI document and routes the document (as a BizDocEnvelope) to Trading Networks processing rules.
2	Trading Networks determines the processing rule to use to deliver the outbound batch EDI document. You create the processing rule to define how you want to deliver the document.

Before You Can Batch EDI Documents

Before the EDI Module can batch EDI documents using the method described in this appendix, you must do the following:

- Install the TN document types for the EDI documents that you will place in the scheduled delivery queues. For instructions on how to install TN document type for EDI documents, see “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104.
- Install the TN document types for the final batch EDI document. For instructions on how to install TN document type for EDI documents, see “[Installing TN Document Types and Creating Flat File Schemas](#)” on page 104.
- Define the default and optionally partner-specific EDITPAs for the sender/receiver pairs to use for group and interchange headers of the batch EDI documents. For more information, see “[Defining EDI Trading Partner Agreements](#)” on page 116. For more information

about the variables in the EDITPAs that affect batching EDI documents, see “[EDITPA Variables that the batchProcess Service Uses](#)” on page 429.

Controlling How the batchProcess Service Forms the Batch Document

You control how the batchProcess service combines the EDI documents in a schedule delivery queue using the input variables to the batchProcess service and by defining EDITPA variables that are used during batch processing.

For more information about:

- The input variables to the batchProcess service, see “[Input Variables to the batchProcess Service](#)” on page 427.
- EDITPA variables, see “[EDITPA Variables that the batchProcess Service Uses](#)” on page 429. For more information about EDITPAs in general, see “[Defining EDI Trading Partner Agreements](#)” on page 116.

Input Variables to the batchProcess Service

When you define the scheduled delivery in Trading Networks, you define the input variables to the service. For more information, see “[Defining the Scheduled Delivery Queues](#)” on page 432. Trading Networks invokes the service to act on documents that are placed in the scheduled delivery queue using the input variables that you specify.

The following table list some of the input variables to the batchProcess service and a description for each. For a description of all the input variables, see the *webMethods EDI Module Built-In Services Reference*.

Input variables to batchProcess service	Description
<i>oneBatchQueue</i>	Indicates the method of batching that you want to use. For backward compatibility, set <i>oneBatchQueue</i> to NONE to use version 6.0.1 batching.

Input variables to batchProcess service	Description
<i>senderIDQualifier</i> <i>senderID</i>	The batchProcess service uses these input variables to locate the partner-specific EDITPA.
<i>receiverIDQualifier</i> <i>receiverID</i>	The batchProcess service uses variables that you define in the partner-specific and default EDITPAs to control how it combines the EDI documents in the queue. For more information, see “EDITPA Variables that the batchProcess Service Uses” on page 429 .
	Additionally, if the <i>envelopeIdentifier</i> variables in the EDITPA are null, the batchProcess service uses these input variables for the sender and receiver of the group and/or interchange headers that it creates in the batch EDI document.
<i>mode</i>	The batchProcess service uses the <i>mode</i> input variable to determine the types of envelopes to add to the batch document that the batchProcess service creates. Specify one of the following: <ul style="list-style-type: none"> ■ IC&GP to add both group and interchange envelopes. This is the default. ■ GP to add group envelopes. ■ IC to add interchange envelopes.
<i>delimiters</i>	The batchProcess service uses the <i>delimiters</i> input variable for the delimiters to use when forming the batch EDI document. The EDITPA also contains delimiters. The <i>delimiters</i> input variable overrides values from the EDITPAs.
	<p>Note: When processing documents in the queue, the batchProcess service must determine the delimiters that the documents in the queue use. To determine the delimiters that documents in the queue use, the batchProcess service uses the interchange header of each document. However, if a document in the queue does <i>not</i> contain an interchange header, the batchProcess service cannot determine the delimiters from the document itself. In this situation, the batchProcess service uses the delimiters you specify for the output batch EDI document. That is, either the input variable <i>delimiters</i> or the delimiters specified in the EDITPA.</p>

EDITPA Variables that the batchProcess Service Uses

To locate the partner-specific EDITPA, the batchProcess service uses the sender and receiver you define in the input variables for the batchProcess service. The following table lists the EDITPA variables that the batchProcess service uses and a description of each.

EDITPA variables that the batchProcess service uses	Description
<i>UNAmode</i>	The batchProcess service uses the <i>UNAmode</i> variable to determine whether to create a UNA segment prior to the interchange. The <i>UNAmode</i> variable is specific to UN/EDIFACT. For more information, see “UNAmode EDITPA Variable” on page 138 .
<i>envelopeIdentifier</i>	<p>The batchProcess service uses the <i>envelopeIdentifier</i> variable for the sender and receiver to use on the group and interchange headers that it adds to the batch EDI document.</p> <p>If the <i>envelopeIdentifier</i> EDITPA variables are null, the batchProcess service uses the input variables to the batchProcess service: <i>senderIDQualifier</i>, <i>senderID</i>, <i>receiverIDQualifier</i>, and <i>receiverID</i>.</p>
	<p>Note: Descriptions of the <i>envelopeIdentifier</i> variables start with “envelopeIdentifier/sender/qualifier EDITPA Variable” on page 141.</p>
<i>ICheaderInfo</i>	The batchProcess service uses the <i>ICheaderInfo</i> variable for generating the interchange headers in the batch EDI document. Descriptions of the <i>ICheaderInfo</i> variables start with “ICheaderInfo/ISA/ISA01 EDITPA Variable” on page 142 .

EDITPA variables that the batchProcess service uses	Description
<i>delimiters</i>	<p>The batchProcess service uses the <i>delimiters</i> variable for the delimiters to use when forming the batch EDI document. You can override the EDITPA <i>delimiters</i> variable by specifying the batchProcess service input variable, <i>delimiters</i>.</p> <p>Note: When processing documents in the queue, the batchProcess service must determine the delimiters that the documents in the queue use. To determine the delimiters that documents in the queue use, the batchProcess service uses the interchange header of each document. However, if a document in the queue does <i>not</i> contain an interchange header, the batchProcess service cannot determine the delimiters from the document itself. In this situation, the batchProcess service uses the delimiters you specify for the output batch EDI document. That is, either the input variable <i>delimiters</i> or the delimiters specified in the EDITPA.</p>

Setting Up to Batch EDI Documents

To batch EDI documents, you need to:

- Determine the number of scheduled delivery queues you need to define. See “[Determining the Number of Queues You Need to Define](#)” below.
- Define the scheduled delivery queues for batching EDI documents. See “[Defining the Scheduled Delivery Queues](#)” on page 432.
- Define the processing rules to deliver documents to the scheduled delivery queues. See “[Defining Processing Rules to Batch EDI Documents](#)” on page 433.

Determining the Number of Queues You Need to Define

The number of scheduled delivery queues you need depends on your site's needs. You indicate your sites needs by setting up the variables that control how the batchProcess service combines the EDI documents. For more information, see ["Controlling How the batchProcess Service Forms the Batch Document" on page 427](#). Specifically, the variables that determine the number of scheduled delivery queues you need are one queue for each unique combination of the following:

Variable	Description
sender	<p>Who you want identified as the sender in the group and/or interchange headers of the batch EDI document.</p> <p>You specify the sender using the <i>envelopeIdentifier</i> variable in the EDITPA. If the <i>envelopeIdentifier</i> EDITPA variables are null, the batchProcess service uses the input variables to the batchProcess service: <i>senderIDQualifier</i> and <i>senderID</i>.</p>
receiver	<p>Who you want identified as the receiver in the group and/or interchange headers of the batch EDI document.</p> <p>You specify the receiver using the <i>envelopeIdentifier</i> variable in the EDITPA. If the <i>envelopeIdentifier</i> EDITPA variables are null, the batchProcess service uses the input variables to the batchProcess service: <i>receiverIDQualifier</i> and <i>receiverID</i>.</p>
mode	<p>The type of envelopes you want the batchProcess service to add to the batch EDI document.</p> <p>You specify the mode using the <i>mode</i> input variable to the batchProcess service. You can specify one of the following:</p> <ul style="list-style-type: none"> ■ IC&GP to add both group and interchange envelopes. ■ GP to add group envelopes. ■ IC to add interchange envelopes.
version	<p>The version of the EDI standard to use for the batch EDI document.</p> <p>You specify the version using the <i>version</i> input variable to the batchProcess service.</p>

For example, if you set up your queues so that the group envelopes in the batch EDI document have a different sender and receiver than the interchange envelope, you would set up one queue in which the mode for batchProcess is GP and another queue in which the mode for batchProcess is IC. You might use this scenario when you want to send batch EDI documents to individual departments within a company.

If you set up a queue in which the mode for the batchProcess service is **GP&IC**, the group envelope and the interchange envelope share the same sender, receiver, and version because the same batchProcess service is creating both the group envelope and the interchange envelope. You use this scenario when you want to send batch EDI documents to a company and let the company disperse the EDI documents to the various departments.

Defining the Scheduled Delivery Queues

You define the scheduled delivery queues using the Trading Networks Console.

To define a scheduled delivery queue for batching EDI documents

To create a scheduled delivery queue, define a public queue in Trading Networks. For steps to create a public queue, see the chapter about queues in the *webMethods Trading Networks User's Guide*.

The following table describe information that you need to supply when defining the queue:

Public queue setting	Description
Queue Name	The name you want to give the scheduled delivery queue. For example, if you are defining a queue for EDI documents for which the receiver is CYG Company, you might set the Queue Name to "CYG Queue."
Delivery Service	The delivery service you want to associate with the queue. Select EDI Batch. Note: When you install the EDI Module, the EDI Module automatically registers the <code>wm.b2b.edtn:batchProcess</code> service with Trading Networks as a scheduled delivery service and assigns it the name EDI Batch. You assign the batchProcess service to a scheduled delivery queue by selecting EDI Batch for Delivery Service.

Public queue setting	Description
Set Inputs	The inputs for the batchProcess service. Click the Set Inputs button to open a dialog box that allows you to set the input variables for the batchProcess service. When Trading Networks invokes the batchProcess service, Trading Networks passes the inputs that you specify. For more information, see "Input Variables to the batchProcess Service" on page 427 and the description of the <code>wm.b2b.editn.batch:batchProcess</code> service in the <i>webMethods EDI Module Built-In Services Reference</i> .
Schedule	When you want Trading Networks to invoke the batchProcess service to act on the EDI documents in the queue you are defining. When defining a schedule, consider how often your trading partners, VAN, etc. should receive the batch EDI documents.

 Note: webMethods recommends that you do not set up private queues for EDI batching because of the limitations that private queues pose. EDI batching with private queues results in EDI documents always going to a specific receiver's queue because you define private queues in a trading partner profile.

Defining Processing Rules to Batch EDI Documents

You define processing rules that instruct Trading Networks to place an EDI document into a scheduled delivery queue that is being used for batching EDI documents. The processing rules you need to define depends on whether you are batching EDI documents in one pass through Trading Networks or whether you are using two passes:

- If you are batching the EDI documents in one pass, you need to define processing rules to drop EDI documents into the scheduled delivery queues that use `mode IC&GP`, which indicates that the batchProcess service is to add both interchange and group envelopes to the batch EDI document in a single pass through Trading Networks.
- If you are batching the EDI documents in two passes, you need to define:
 - Processing rules to drop EDI documents into the scheduled delivery queues that use `mode GP`, which indicates that the batchProcess service is to add group envelopes to the batch EDI document in a first pass through Trading Networks.
 - Processing rules to drop EDI documents into the scheduled delivery queues that use `mode IC`, which indicates that the batchProcess service is to add an interchange envelope to the batch EDI document in a second pass through Trading Networks.

To define a processing rule to batch EDI documents

To define a processing rule in Trading Networks, use the Trading Networks Console. For steps to create a processing rule, see the chapter about processing rules in the *webMethods Trading Networks User's Guide*

The following table lists details about how to define the processing rules.

On this Processing Rules tab...	Specify...								
Criteria	<p>The standard criteria you want to use to select the processing rule. Specify criteria that describes the EDI documents you want to place in a scheduled delivery queue. How you set the criteria depends on mode (GP, IC, or IC&GP) that the scheduled delivery queue associated with this processing rule uses:</p> <table> <thead> <tr> <th>Mode</th><th>Description</th></tr> </thead> <tbody> <tr> <td>GP</td><td> <p>If the mode is GP, you are batching in two passes and this is the first pass. In this situation, the criteria should identify the EDI documents that you want to combine and to which you want to add group envelopes.</p> <p>For example, if you want to use the Sender criterion, you might identify a trading partner that sends EDI documents that you want to batch.</p> </td></tr> <tr> <td>IC</td><td> <p>If the mode is IC, you are batching in two passes and this is the second pass. In this situation, the criteria should identify the EDI documents that you want to combine and to which you want to add an interchange envelope.</p> <p>For example, if you want to use the Sender criterion, you might identify the sender that the batchProcess service assigns to the batch EDI document it created in the first pass.</p> </td></tr> <tr> <td>IC&GP</td><td> <p>If the mode is IC&GP, you are batching in one pass. In this situation, the criteria should identify the EDI documents that you want to combine and to which you want to add group and injecting envelopes.</p> <p>For example, if you want to use the Sender criterion, you might identify a trading partner that sends EDI documents that you want to batch.</p> </td></tr> </tbody> </table>	Mode	Description	GP	<p>If the mode is GP, you are batching in two passes and this is the first pass. In this situation, the criteria should identify the EDI documents that you want to combine and to which you want to add group envelopes.</p> <p>For example, if you want to use the Sender criterion, you might identify a trading partner that sends EDI documents that you want to batch.</p>	IC	<p>If the mode is IC, you are batching in two passes and this is the second pass. In this situation, the criteria should identify the EDI documents that you want to combine and to which you want to add an interchange envelope.</p> <p>For example, if you want to use the Sender criterion, you might identify the sender that the batchProcess service assigns to the batch EDI document it created in the first pass.</p>	IC&GP	<p>If the mode is IC&GP, you are batching in one pass. In this situation, the criteria should identify the EDI documents that you want to combine and to which you want to add group and injecting envelopes.</p> <p>For example, if you want to use the Sender criterion, you might identify a trading partner that sends EDI documents that you want to batch.</p>
Mode	Description								
GP	<p>If the mode is GP, you are batching in two passes and this is the first pass. In this situation, the criteria should identify the EDI documents that you want to combine and to which you want to add group envelopes.</p> <p>For example, if you want to use the Sender criterion, you might identify a trading partner that sends EDI documents that you want to batch.</p>								
IC	<p>If the mode is IC, you are batching in two passes and this is the second pass. In this situation, the criteria should identify the EDI documents that you want to combine and to which you want to add an interchange envelope.</p> <p>For example, if you want to use the Sender criterion, you might identify the sender that the batchProcess service assigns to the batch EDI document it created in the first pass.</p>								
IC&GP	<p>If the mode is IC&GP, you are batching in one pass. In this situation, the criteria should identify the EDI documents that you want to combine and to which you want to add group and injecting envelopes.</p> <p>For example, if you want to use the Sender criterion, you might identify a trading partner that sends EDI documents that you want to batch.</p>								

On this Processing Rules tab...	Specify...						
Extended Criteria	<p>The custom attributes that you want to use as criteria to select the processing rule.</p> <p>You will typically only use extended criteria when the scheduled delivery queue associated with this processing rule uses mode IC. In this situation, the following shows the extended criterion you should add:</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; width: 30%;">Attribute</th><th style="text-align: left; width: 30%;">Operator</th><th style="text-align: left; width: 40%;">Value</th></tr> </thead> <tbody> <tr> <td>EDI Batch</td><td>Equal</td><td>Group</td></tr> </tbody> </table> <p>For more information about the EDI Batch custom attribute, see "The EDI Batch Attribute and Processing Rules" on page 436.</p>	Attribute	Operator	Value	EDI Batch	Equal	Group
Attribute	Operator	Value					
EDI Batch	Equal	Group					
Action	<p>What you want Trading Networks to do with the EDI document it is processing.</p> <p>To place the document into a scheduled delivery queue that is used for batching EDI documents, do the following:</p> <ul style="list-style-type: none"> ■ Select Deliver Document By processing action. ■ For the Deliver Document By processing action, select Scheduled Delivery. ■ Select the appropriate scheduled delivery queue that you associated with the batchProcess service from the list of registered queues. 						

The EDI Batch Attribute and Processing Rules

When defining a processing rule, you can use the extended criteria of the processing rule to instruct Trading Networks to select a processing rule based on custom attributes that are associated with the document. When you use the EDI batching, the batchProcess service associates the EDI Batch custom attribute with the batch EDI document that it creates.

The following table describes how the batchProcess service sets the value of the EDI Batch custom attribute based on whether the mode is GP, IC, or IC&GP.

Mode	EDI Batch attribute set to	How to use it
GP	Group	When batching in two passes through Trading Networks, when the EDI Batch attribute is Group , the processing rule should have the Deliver Document By action place the EDI document into a scheduled delivery queue that uses mode IC for its second pass through Trading Networks.
IC	Interchange	When batching in two passes through Trading Networks, when the EDI Batch attribute is Interchange , this signals that the batch EDI document is final and is ready to be delivered. Use this criterion in a processing rule that you create to deliver the batch EDI document. For more information, see "Delivering the Batch EDI Document" below.
IC&GP	Interchange	When batching in one pass through Trading Networks, when the EDI Batch attribute is Interchange , this signals that the batch EDI document is final and is ready to be delivered. Use this criterion in a processing rule that you create to deliver the batch EDI document. For more information, see "Delivering the Batch EDI Document" below.

Delivering the Batch EDI Document

After the batchProcess service creates the final batch EDI document, you can use a Trading Networks processing rule to deliver it. When defining the processing rule to deliver a batch EDI documents, use the EDI Batch attribute to select batch EDI documents that are final. To deliver a batch EDI document, the processing rule can use one of the following actions:

- Execute a Service processing action to invoke a service that you create to deliver the final batch EDI document.
- Deliver Document By processing action to send the document to a VAN.

To define a processing rule to deliver a final batch EDI document

To define a processing rule in Trading Networks, use the Trading Networks Console. For steps to create a processing rule, see the chapter about processing rules in the *webMethods Trading Networks User's Guide*.

The following table lists details about how to define the processing rules.

On this Processing Rules tab...	Specify...						
Criteria	<p>The standard criteria you want to use to select the processing rule. Specify criteria that describes the final batch EDI document that the batchProcess service creates. For example, if you want to use the Sender criterion, identify the sender that the batchProcess service associates with the final batch EDI document.</p>						
Extended Criteria	<p>The custom attributes that you want to use as criteria to select the processing rule. The following shows the extended criterion you should add:</p> <table border="1"> <thead> <tr> <th>Attribute</th><th>Operator</th><th>Value</th></tr> </thead> <tbody> <tr> <td>EDI Batch</td><td>Equal</td><td>Interchange</td></tr> </tbody> </table> <p>For more information about the EDI Batch custom attribute, see “The EDI Batch Attribute and Processing Rules” on page 436.</p>	Attribute	Operator	Value	EDI Batch	Equal	Interchange
Attribute	Operator	Value					
EDI Batch	Equal	Interchange					

On this Processing Rules tab...	Specify...
Action	<p>How you want Trading Networks to deliver the final batch EDI document.</p> <ul style="list-style-type: none">■ Select Execute a Service processing action to invoke a service that you create to deliver the final batch EDI document.■ Select Deliver Document By processing action to send the document to a VAN. For more information, see “Setting Up to Deliver Documents to VANs” on page 337 in “Retrieving and Delivering EDI Documents from and to VANs” on page 335.

Index

Symbols

RID element 97

Numerics

997 document, functional acknowledgment 56, 362

A

A = Accepted validation code 69

ABORTED, processing status on Transaction Analysis screen 379

Accept FA status, FA reconciliation 365

Accept with Errors FA status, FA reconciliation 365

Accepted setting

- FAGeneration/rejectionRules/logicalErrorStatus EDITPA variable 266

- FAGeneration/rejectionRules/syntaxErrorStatus EDITPA variable 265

- Logical Error Status field 398

- logicalErrorStatus input variable of generateFA 61

- Syntax Error Status field 398

- syntaxErrorStatus input variable of generateFA 60

Accepted, But Errors Were Noted setting

- Child Transaction Rejected Status field 400

- FAGeneration/rejectionRules/logicalErrorStatus EDITPA variable 266

- FAGeneration/rejectionRules/syntaxErrorStatus EDITPA variable 265

- Logical Error Status field 398

- logicalErrorStatus input variable of generateFA 61

- Syntax Error Status field 397

- syntaxErrorStatus input variable of generateFA 60

Accepted, But Errors Were Noted, validation code for FAs 69

Accepted, validation code for FAs 69

acknowledgments, functional (FAs), see *FAs* 258

Action, processing rule criteria 331, 333

actions, for invalid control numbers

- defining 170

- Error & Continue 170, 241, 393

- Process Normally 170, 241, 393

- Reject 170, 242, 394

activity log, viewing 374

Add Group field, Outbound Information - FA Generation 400

addICEnvelope, *See* `wm.b2b.edi.util:addICEnvelope` service adding

- control number information 177

- external ID types for EDI ID qualifiers 114

- external ID types to Trading Networks for EDI ID qualifiers 114

- group and interchange headers to outbound EDI document 290

- UN/EDIFACT envelopes 91

addNewEDIVersion service 107

addToTradacomsTransmission service 290

advantages of

- non-standard processing 111

- processing document segment by segment 42

- processing entire document at one time 42

- segment mapping 73

- standard processing 111

Agreement Details screen 116, 118, 150, 152

Agreement ID, EDITPA 117, 118, 151, 152

Agreement Status of EDITPA 116, 150

Alert e-mail processing rule action

- setting when FA status unacceptable 281

- setting when invalid control number 249

All setting

- FAGeneration/processDocument EDITPA variable 274, 275

- Process Document field 396

ANSI X12 810 document

- example of iterative processing 48

- example of mapping 50

ANSI X12 850 document

- example of mapping 80

- N1 segment mapping example 73

ANSI X12 997 document 57

- ST/AK1/AK102 instance ID 356

ANSI X12 document 92

- business process 347

- EDI Processing Mode, custom attribute 123

- FA reconciliation 362

- sample process models 347

- ANSI X12 Group document
FA status 365
FAReconciliation 364
- ANSI X12 group envelope 56
- ANSI X12 interchange header 402
- ANSI X12 standard, FA 997 document 56
- application/EDI
content type 32, 187
edidata data type 41
- application/EDIstream
content type 32, 187
edidata data type 41
- application/UNEDIFACT
content type 32, 187
edidata data type 41
- application/x
-wmedisemiwrap132 content type 187
-wmedisemiwrap80 content type 187
-wmediwrap132 content type 187
-wmediwrap80 content type 187
- application/X12
and data type for edidata variable 41
content type 32, 187
- applyFormat, format service 28
- assigning conversation IDs to EDI documents 352
- associating
custom attribute with EDI document 204, 225
EDI format services with EDI data types 29
group-level sender/receiver pair 404
queue with scheduled delivery service 338
- attributes
custom, *See* custom attributes
document, *See* document attributes
system, *See* system attributes 197, 223
Trading Networks, *See* Trading Networks attributes
- Attributes variable in pipeline 211, 232
- Auto Generate FA field 260, 395
- automatic generation of FAs
See FAs automatic generation
- B**
- back-end systems 284
- batch and transmission headers
control numbers for outbound EDI documents 284
- batch control number, validating 154
- Batch documents, TRADACOMS
DocumentID system attribute 223
Execute a Service processing action 231
GroupID system attribute 223
system attribute 223
viewing 376
- batch documents, version 6.0.1 method 426
controlling how the batchProcess service forms the document 427
- delivering batch EDI document 437
- determining number of queues to define 431
- GP mode 424
- IC mode 424
- IC&GP mode 420
- prerequisites 426
- procedure to define queues 432
- processing rules 433
- setting up 430
- single pass through Trading Networks 419, 433
- two passes through Trading Networks 421, 422, 433
- batch headers
control numbers, validating 154
validating batch control numbers 154
- batching
and FA reconciliation 321
and UNAmode EDITPA variable 138
BizDocEnvelope for final batch EDI document 321
collection areas for interchanges in batch EDI data 312
combining EDI documents in queue 311
- creating
multiple documents each with one interchange 310
single document with multiple interchanges 310
- defining processing rule for batch document 331
- delimiters
for collection areas 313
replacing to match output EDI batch data 313
to use for batch EDI document 318
used when no interchange header 317
- Deliver Document By processing action 331, 333
- delivering batch EDI document 332
- error, IS document published 320
- example of sorting documents to be batched 314

- Execute a Service processing action 332, 333
- extracting and sorting transactions from documents
 - so they can be batched 311
 - steps for 311, 315
- failure notification, setting publishBatchFailEvent variable 319
- headers
 - group in batch EDI document 320
 - interchange in batch EDI document 320
 - when no interchange header in documents to be batched 315
- prerequisites to batching when using Trading Networks 326
- private queues, why not to use 330
- processing rule to deliver batch EDI document 331
- recombining transactions to form batch EDI document 320
- setting up how to batch 329
- subcollection areas for groups in batch EDI data 312
- task status, updating for document included in batch 311, 315
- viewing progress of 379
- `wm.b2b.editn.batch:batchProcess` service 310, 326
- batchProcess service
 - See* `wm.b2b.editn.batch:batchProcess` service
- bizdoc variable
 - in pipeline 211, 232
 - output for `createEDIBizDoc` service 304
- bizdoc/Attributes/EDI Processing Mode variable 214, 218
- BizDocEnvelope
 - for final batch EDI document 321
 - in pipeline 211, 232
 - service to create for outbound EDI document 304
- boundary characters of mainframe EDI documents 187
- BRANCH flow operation, example of in N1Segment service 76
- business process
 - ANSI X12 documents, sample process models 347
 - defined 346
 - EDI documents, Transaction document 347
 - monitor 357
 - UN/EDIFACT documents, sample process models 350
- C
- cap, control number
 - default 172
 - defining 177
 - definition 172
 - setting 177
- Check for Duplicate Document pre-processing action 208, 229
 - setting when invalid control number 249
- child transaction rejected status
 - and FA status for group 64, 270
 - and FA status for UN/EDIFACT interchange 65, 271
 - description 62, 267
- Child Transaction Rejected Status field 268, 399
- client
 - application/EDI stream 32
 - content type 32
 - content type to send EDI documents to TN 186
 - FTP, *See* FTP client
 - HTTP, *See* HTTP client
 - Integration Server that submits the documents to Trading Networks 186
 - sending documents as `InputStreams` 97
- codes, validation 69
- coding services to process Transaction and Group documents 210, 231
- collection areas for batching
 - definition 312
 - delimiters
 - associated with 318
 - associated with default collection area 317
 - when no interchange header 317
- compliance check 47
 - for EDI documents, description 47
 - for processing document iteratively 52
 - variable of envelopeProcess service 47
 - `wm.b2b.edi:envelopeProcess` service 47
- complianceCheck variable for `wm.b2b.edi:envelopeProcess` service 47
- concatenating
 - `ISA05, ISA06, and GS02` fields 121
 - `ISA07, ISA08, and GS03` fields 121
 - multiple strings using variable substitution 78
 - Strings, diagram of sample pipeline for 78
 - transaction contents with its header and trailer into a String 45
- configuration file
 - `format.xml` configuration file 25
 - `properties.cnf`, EDI Module 58, 94, 149, 366, 380
- Configuration menu of EDI Module home page 26
- Configure Field Formats, option of EDI Module home page 26, 29

Configure Format Settings screen, associating EDI format services with EDI data types 29

Configure Properties, option of EDI Module home page 58, 94, 149, 366, 380

configuring

- EDI Module for large document handling 94
- File Polling listener port 37
- how to convert field values in documents 26
- maximum number of transaction errors 58

content of EDI document, viewing on Transaction Analysis screen 377

content type

- and edidata variable 41
- application/EDI content type 32, 187
- application/EDIstream 32, 187
- application/UNIEDIFACT 32, 187
- application/X12 32, 187
- application/x-wmedisemiwrap132 187
- application/x-wmedisemiwrap80 187
- application/x-wmediwrap132 187
- application/x-wmediwrap80 content type 187

boundary characters of mainframe EDI documents 187

File Polling 191

- non-mainframe EDI document 187
- of client to send EDI documents to TN 186

Content type field, File Polling listener port 37

CONTRL message, functional acknowledgment 56, 362

control number cap

- Control Number Cap field 180
- default 172
- defining 177
- definition 172
- setting 177

Control Number field 180

control number increment

- Control Number Increment field 180
- default 172
- defining 177
- definition 172
- setting 177

control number minimum

- Control Number Minimum field 180
- default 172
- defining 177
- definition 172
- setting 177

control number window

- Control Number Window field 180

default 173

defining 177

definition 173

- example of using to determine duplicate 176
- example of using to determine out-of-sequence 176
- setting 177

control numbers

- actions when invalid
 - Error & Continue 170, 241, 393
 - Process Normally 170, 241, 393
 - Reject 170, 242, 394
- adding information 177

cap

- default 172
- defining 177
- definition 172

defining

- action to take when invalid 170
- settings 177

determining whether duplicate or out-of-sequence 175

determining whether valid 174

duplicate

- defining action for 171, 172
- definition 175
- reprocessing documents with 250
- user status of document 242

group headers 238

increment

- default 172
- defining 177
- definition 172

interchange headers 238

invalid and EDITPA splitOption variable use 242

minimum

- default 172
- defining 177
- definition 172

next expected 172

- determining 175

obtaining for outbound processing 290

obtaining for outbound processing, TRADACOMS 291

- outbound EDI documents 284
- out-of-sequence
 - defining action for 171, 172
 - definition 176
 - reprocessing documents with 250
 - service to reprocess documents with 252
 - user status of document 242
- procedure to validate control number 239
- processing rule to handle invalid control number 247
- searching for 181
- setting initial value 177
- used in FA 261
- validation 238
 - batch control numbers 154
 - file control numbers 155
 - group control numbers 125
 - interchange control numbers 124
 - transmission control numbers 154
 - turning on or off 168
- window
 - default 173
 - defining 177
 - definition 173
 - example of using to determine duplicate 176
 - example of using to determine out-of-sequence 176
- Control Numbers menu of EDI Module home page 178, 181
- controlNumber input variable 304
- ControlNumberManagement
 - duplicateControlNumberAction EDITPA variable 125, 127, 389
 - use during inbound processing 245, 247
 - outOfSequenceControlNumberAction EDITPA variable 389
 - useReverseRouting EDITPA variable 127
 - validateInboundBatchControlNumbers EDITPA variable
 - use during inbound processing 246
 - validateInboundEnvelopeControlNumbers EDITPA variable 124
 - use during inbound processing 244
 - use during outbound processing 294
 - validateInboundFileControlNumbers EDITPA variable
 - use during inbound processing 246
 - validateInboundGroupControlNumbers EDITPA variable 125
 - use during inbound processing 244
 - use during outbound processing 294
 - validateInboundTransmissionControlNumbers EDITPA variable
 - use during inbound processing 246
- conventions used in this document 15
- conversation ID
 - assigning to EDI documents 352
 - business process, instance of 346
 - instanceID 352, 353
- ConversationID system attribute 197, 198
- conversations
 - defined 346
 - See also* business process
- converting
 - EDI documents to IData objects iteratively 98
 - field values 25, 26
 - field values from one form to another 77
 - large documents to IData objects iteratively 344
 - segment or group of segments to IData object 51
 - unparsed section to IData object 52
 - XML document into, example of 92
- convertToValues service, *See* `wm.b2b.edi:convertToValues` service 46
- Create Doc field 392, 411
- Create FA Reports menu option 363
- creating
 - client to Integration Server 32
 - client to send documents to EDI Module for processing 97
 - custom document attributes for EDI documents 204, 225
 - customized services to connect to other VANs 341
 - FA reconciliation reports 368
- flat file schema
 - for internal-format documents 24
 - from SAP IDOC 23
 - from SEF file 22
- FTP client 35
- IS document type for transaction from flat file schema 356
- mapping service
 - best practices for EDI to XML 79
 - best practices for XML to EDI 85
 - EDI to XML 80
 - XML to EDI 85
- profiles 113, 148
- segment-mapping service, example of 74
- service
 - form EDI document 88
 - process EDI documents 40
 - process EDI documents, prerequisites 40

- criteria, processing rules 207, 228
batch EDI documents 331
deliver batch EDI document 332
document with invalid control number 248
outbound FA 278
Cross Reference > GS Pairs screen 406
ctlFromTable variable 289, 290
custom attributes
 defined 196, 222
 defining processing rule 205, 226
EDI FA Status 202
EDI Group Type 200
EDI Outbound FA 200
 processing rules outbound FAs 279
EDI Processing Mode 122, 200
EDI Status 200, 248
 processing rules for documents with invalid control numbers 248
 processing rules for documents with unacceptable FA statuses 280
EDI Version 201
Envelope CntrlNum Status 201, 224
Group CntrlNum Status 201, 224
setting value at run time 204, 225
that EDI Module sets 200, 224, 247
Transaction CntrlNum Status 224
Custom Mode
 and ISA015 field of ANSI X12 document 123
 envelope information 391
 processingMode variable 122
customizing
 EDI flat file schema for iterative processing 48
 services to connect to other VANs 341
- D**
- Data Status of EDITPA 116, 150
date
 document received, viewing on Transaction Analysis screen 377
 example of converting field values 77
 searching for FA reconciliation reports
 FA from date field 369
 FA to date field 369
 From date field, FA reconciliation report 369
 To date field, FA reconciliation report 369
Decimal field 401
decryption and VAN.VANConnectivity:getFromVAN service 337
default
 EDITPA 116, 118, 150
 defining 116, 150
 modifying 116, 150
 FTP port 9021 35
 level of document you want FA to acknowledge 57
 maximum number of transaction errors 58
 pre-processing settings 208, 229
Default setting
 FA Level field 395
 FAGeneration/FA Level EDITPA variable 260
defining
 control number settings 177
 custom document attributes for EDI documents 204, 225
 default EDITPA 116, 150
 group-level sender/receiver pairs 404
 instance ID query for transaction 354
 interchange-level sender/receiver pairs 390
 for non-standard processing 390
 partner information for
 non-standard processing 386
 standard processing 112, 146
 partner-specific EDITPA 117, 151, 388
process model 347
processing rule to
 batch EDI documents 331
 batch EDI documents when using version 6.0.1 batching 433
 batch EDI documents, criteria for 331
 deliver EDI documents to VANS 340
 deliver final batch EDI document 332
 process document with invalid control number 247, 279
 process inbound EDI documents 206, 227
 process outbound FAs 278
 use custom attribute as criteria 205, 226
profiles 113, 147, 387
scheduled delivery queues 329, 432
 determining number of queues 329
 number to define when using version 6.0.1 batching 431
 to deliver EDI documents to VAN 337
deleting
 FA reconciliation report 370
interchange-level sender/receiver pair information 403

- delimiters 401
 - batching EDI documents 318
 - defaults used in batching when cannot be retrieved from other sources 318
 - EDITPA variable 327, 430
 - outbound EDI documents 284
 - used in batch EDI documents 313
 - `wm.b2b.editn.batch:batchProcess` service
 - and EDITPA variable 327, 430
 - input variable 428
 - delimiters/field EDITPA variable 138
 - delimiters/record EDITPA variable 138
 - delimiters/release EDITPA variable 139
 - delimiters/subfield EDITPA variable 139
- Deliver Document By processing action 209, 230
 - delivering batch EDI document 332
 - delivering EDI documents to VANs 340
 - outbound FA 279
 - processing rule criteria 331, 333
 - scheduled delivery queue 310
 - viewing information about EDI documents in queues 379
- delivering
 - EDI document
 - directly from service that forms it 292
 - methods available 291
 - routing via Trading Networks 301
 - submitting to Trading Networks 293
 - outbound batch EDI document when using version 6.0.1 batching 426
- Delivery Service
 - public queue setting 338
 - scheduled delivery queue 330
- delivery tasks
 - scheduled delivery queue 379
 - statuses for batch EDI documents 319
- diagram of
 - batching documents
 - example of collection and subcollection areas 313
 - multiple batch documents each with single interchange 324
 - MULTIPLEOUTPUTS batching 324
 - one batch document with multiple interchanges 322
 - scheduled delivery queue to hold documents 314
 - SINGLEOUTPUT batching 322
 - transactions sorted into collection areas 314
- batching documents using version 6.0.1 batching
 - combining EDI documents in a single pass through Trading Networks 420
 - delivering batch EDI documents 426
 - first pass through Trading Networks when batching in two passes 424
 - single pass through Trading Networks 419
 - two passes through Trading Networks 422
- concatenating Strings
 - sample pipeline for 78
 - using variable substitution 79
- control numbers in ANSI X12 ISA, GS, and ST headers 197
- delivering outbound EDI document 303
 - delivering from the service that forms it 292
 - drop variables after invoking `wm.tn.doc.xml:routeXml` service 301
 - setting scope to prepare to execute `routeXml` service 300
 - submitting back to Trading Networks 298
- forming outbound EDI document
 - adding UN/EDIFACT envelopes to EDI document 91
 - drop variables after invoking `wm.tn.doc.xml:routeXml` service 301
 - from an internal-format document 89
 - limiting pipeline elements to execute the `routeXml` service 300
 - pipeline after setting scope for the INVOKE `routeXml` flow operation 300
 - saving output from the `addICEnvelope` service 299
 - service to create a BizDocEnvelope 304
 - setting scope to prepare to execute `routeXml` service 300
 - validating internal-format document used to create EDI document 91
 - when using non-standard processing 413
 - when using standard processing 286
- generating FAs
 - invoking `generateFA` before invoking `envelopeProcess` 66
 - invoking `generateFA` immediately after invoking `envelopeProcess` 67
 - invoking `generateFA` when processing an inbound document iteratively, segment by segment 68
- inbound EDI document
 - processing entire document at one time 43
 - processing iteratively, segment by segment 50
 - processing Transaction document 213, 234
 - processing transactions in a Group document 215

- Process Model Illustrating Processing of
ANSI X12 850 Transaction document 349
ANSI X12 Group document 348
UN/EDIFACT Interchange document 350
UN/EDIFACT Transaction document 351
reprocessing documents with out-of-sequence control
numbers 252
sampleServices:X12toValue 83
Tutorial.EDItoXML.segmentMapping:N1Segment service 77
 BRANCH flow operation 76
 inputs and outputs 75
 LOOP flow operation 76
Tutorial.EDItoXML:dateParse service, inputs and outputs 78
Tutorial.EDItoXML:EDItoPOMap service, inputs and outputs 81
Tutorial.EDItoXML:EDItoXMLPOMap service 82
Tutorial.EDItoXML:EDItoXMLPOMap service, final map of 82
Tutorial.EDItoXML:processEDI850_4010 service 84
digital signature, Verify Digital Signature pre-processing action 209, 229
Disable FA status, FA reconciliation 137, 365
disabling FA reconciliation 137
disadvantages of
 non-standard processing 111
 processing document segment by segment 42
 processing entire document at one time 42
 standard processing 111
Doc Exchange menu of EDI Module home page 104, 108
Doc Timestamp, FA reconciliation report field 372
docID variable 212, 233
docProcessingSvcName variable 83
DocResxxxx.dat, large file naming 96
docTypeID variable 212, 233
docTypeName variable 212, 233
document attributes, custom, *See* custom attributes
document IDs
 FA reconciliation 363
 of FA 372
document recognition 104
Document Type
 See also TN document types
 FA reconciliation report field 371
 processing rule critiera 207, 228
documentation
 additional 16
 conventions used 15
 feedback 16
DocumentID
 FA reconciliation report field 371
 system attribute 197, 223, 304
 variable in pipeline 211, 232
documents
 EDI *See* EDI documents
 internal-format, *See* internal-format documents
 XML, *See* XML documents
documentTypeName input variable 84
DONE processing status on Transaction Analysis screen 378
DONE W/ ERRORS processing status on Transaction Analysis
screen 378
DUNS external ID type, Trading Networks 114
Duplicate control number action field 393, 409, 411
duplicate control numbers
 defining action for 171, 172
 reprocessing documents with 250
 user status 242
Duplicate FA status, FA reconciliation 365
Duplicate FA/Errors status, FA reconciliation 365
- ## E
- E = Accepted, But Errors Were Noted validation code 69
EDI Batch custom attribute
 associating with batch EDI document 332
 Interchange value 333
 processing rule criteria 333
EDI Batch delivery service 330
EDI content type 37
EDI data types, associating with format services 29
EDI documents 138
 and splitOption variable 119
 and Trading Networks attributes 196, 222
 associating custom attributes with 204, 225
 batching, prerequisites 326
 batching, prerequisites when using version 6.0.1 batching 426
 combining those to be batched 311
 configuring how to convert field values 26
 content, viewing on Transaction Analysis screen 377

- conversation IDs 352
- converting to *IData* objects iteratively 98
- creating
 - flat file schema for 20
 - from internal-format document 88
 - service to form outbound 88
 - service to form outbound, prerequisites for 285
 - service to process inbound 40
- flat file schema that defines structure 20
- forming outbound using standard processing 286
- inbound, *See* inbound EDI documents
- information you can access for outbound 284
- large, *See* large documents
- level you want FA to acknowledge 57
- logic to include in service to
 - form EDI document 89
 - process EDI documents 41
 - process EDI documents iteratively 50
- mapping to another format 73
- outbound, *See* outbound EDI documents
- processing inbound
 - entire document at one time 42, 43, 66
 - segment by segment 42, 47, 68
 - using Trading Networks 194, 220
- processing rule
 - process document with invalid control number 247, 279
 - process inbound 206, 227
- receiver, viewing on Transaction Analysis screen 377
- reconciling with FAs 362
- sender, viewing on Transaction Analysis screen 377
- sending to Integration Server 32
 - using File Polling 37
- setting up Trading Networks to deliver 302
- submitting to Trading Networks 186
- summary information 211, 232
- viewing
 - related documents 380
 - viewing on Transaction Analysis screen 376, 377
- ways to process 41
- EDI FA Status custom attribute 202
- EDI Group Type custom attribute 200
- EDI ID qualifier
 - adding external ID types for 114
 - adding external ID types to Trading Networks 114
 - and partners' external ID 113, 147
 - and Trading Networks 114
 - envelopedIdentifier/receiver/qualifier variable 142
 - envelopedIdentifier/sender/ID variable 141
 - for D-U-N-S number as external ID type 114
 - GS Receiver Qualifier 392, 411
 - GS Sender Qualifier 392, 411
 - Receiver Qualifier 183, 391, 403, 406, 407, 410
 - ReceiverQualifier 179
 - Sender Qualifier 178, 182, 391, 403, 406, 407, 410
- EDI Module
 - content handler 32, 37
 - ConversationID system attribute 197
 - custom attributes 200, 224, 247
 - DocumentID system attribute 197, 223
 - EDI FA Status custom attribute 202
 - EDI Group Type custom attribute 200
 - EDI Outbound FA custom attribute 200
 - EDI Processing Mode custom attribute 200
 - EDI Status custom attribute 200, 248
 - EDI Version custom attribute 201
 - Envelope CntrlNum Status custom attribute 201, 224
 - FA status, determining 263
 - Group CntrlNum Status custom attribute 201, 224
 - GroupID system attribute 197, 223
 - home page
 - Add External ID Types 114
 - adding control numbers 178
 - adding external ID types 114
 - adding interchange sender/receiver pair information 391
 - associating EDI format services with EDI data types 29
 - Configuration menu 26
 - Configure Field Formats 29
 - Configure Properties 58, 94, 149, 366, 380
 - configuring large document handling 94
 - configuring maximum number of errors in FA reports 58
 - configuring TRADACOMS external ID types 149, 380
 - Control Numbers menu 178, 181
 - controlling FA status for documents sent multiple times 366
 - Create FA Reports 368

- Define Instance ID queries 354
- defining group-level sender/receiver pair associations 406
- deleting interchange sender/receiver pair information 403
- Doc Exchange menu 104, 108
- EDI Processes menu 354, 357
- editing interchange sender/receiver pair information 403
- FA menu 368
- Format Settings screen 29
- Install TN Document Types 104, 108
- installing flat file schemas 104, 108
- Manage Control Numbers 178, 181
- Manage Interchange Info 391, 403, 406
- Partner Set Up menu 114, 391, 403, 406, 407
- Properties Settings screen 58, 94, 149, 366, 380
- Query Envelope IDs 407
- querying for group-level sender/receiver pair associations 407
- Run a Demo 357
- run demo of EDI documents in a business process 357
- searching for control numbers 181
- server log 374
- system attributes 197, 223
- Transaction CntrlNum Status custom attribute 224
- EDI Outbound FA custom attribute 200
- EDI Processes menu of EDI Module home page 354, 357
- EDI Processing Mode custom attribute 122, 200, 214, 218
Testing 215, 218
- EDI recognizer
and FAReconciliation EDITPA variable 293
and Trading Networks 194, 220, 291, 293group level split 293

parsing EDI document 211, 232
- EDI standards
adding support for new versions 107
- EDI Status custom attribute 200, 248
- Duplicate Control Number 248
- Generate FA - Accepted, But Errors Were Noted 273
- Generate FA - Not Allowed 273
- Generate FA - Partially Accepted 273
- Generate FA - Rejected 273
- how set 200
- Out of Sequence Control Number 248
- Processed 201
- processing rules for documents with invalid control numbers 248, 279, 280
- values for invalid control numbers 248
- values when FA status unacceptable 273
- EDI Trading Partner Agreement, *See* EDITPA
- EDI Version custom attribute 201
- EDIBigDocThreshold configuration property 95
- EDIControlNumber table 290, 291
key for entries in table 173
settings contained in 173use 167
use during inbound processing 239
- EDICOR server log key 374
- edidata pipeline variable 32
and application/X12 content type 41
and content type 41application/EDI 41
application/EDIstream 41application/UNEDIFACT 41
- file polling 37
generating FAs 66, 68
- inputs for service 41
- non-mainframe EDI documents 187
- EDIEnvelope table 390, 405
- EDIFFSchema folder 104, 105
- EDIFFSchema.Standard.Vversion:Transaction 355
- EDIFTN server log key 374
- EDIGroup table 405
- EDIMaxFATransactionErrors configuration property 58
- EDITPA
ControlNumberManagement
duplicateControlNumberAction variable 125, 127, 389
outOfSequenceControlNumberAction variable 389useReverseRouting variable 127
validateInboundEnvelopeControlNumbers variable 124
validateInboundGroupControlNumbers variable 125
- default, *See* default EDITPA
- definition 116, 150
- delimiters
field variable 138
record variable 138
release variable 139
subfield variable 139

- envelopeIdentifier
 - receiver/ID variable 141
 - receiver/qualifier variable 142
 - sender/ID variable 140
 - sender/qualifier variable 141
- FAGeneration
 - addGroup variable 129
 - autoGenerateFA variable 129
 - FALevel variable 129
 - generateControlNumber variable 130
 - processDocument variable 135
 - rejectionRules/childTransactionRejectedStatus variable 133
 - rejectionRules/logicalErrorStatus variable 132
 - rejectionRules/syntaxErrorStatus variable 131
- FAReconciliation variable 136
- GSRouting
 - receiverQualifier variable 122
 - routingMode variables 120
 - senderQualifier variable 121
- ICheaderInfo
 - GS/GS07 variable 143
 - ISA/ISA01 variable 142
 - ISA/ISA02 variable 142
 - ISA/ISA03 variable 143
 - ISA/ISA04 variable 143
 - ISA/ISA11 variable 143
 - UNB/UNB01 variable 144
 - UNB/UNB06 variable 144
 - UNB/UNB08 variable 144
- IS Document Type 117, 118, 151, 152
- partner-specific, *See* partner-specific EDITPA
- persistMultipleDocEnvelope variable 123, 158
- processingMode variable 122, 214, 218
- publishBatchFailEvent variable 140, 159
- routingMode variable 120
- splitOption variable 119, 388
- TRADACOMS/ControlNumberManagement
 - duplicateControlNumberAction variable 155
 - outOfSequenceControlNumberAction variable 156
 - validateInboundBatchControlNumbers variable 154
 - validateInboundFileControlNumbers variable 155
 - validateInboundTransmissionControlNumbers variable 154
- TRADACOMS/enforceApplicationReference variable 157
- TRADACOMS/Outbound/applicationReference variable 296
- TRADACOMS/Outbound/applicationReferenceFromFile variable 296
- TRADACOMS/Outbound/createReconciliationMessage variable 295
- TRADACOMS/Outbound/recipientTransmissionReference variable 295
- TRADACOMS/splitOption variable 153, 295
- TRADACOMS/storageOption variable 154
- Trading Networks Console 116, 117, 150, 151
- UNAmode variable 138
- wm.b2b.editn.TPA:EDITPA IS document type 118, 152
- EDITPA variables
 - delimiters 327, 430
 - delimiters/field 138
 - delimiters/record 138
 - delimiters/release 139
 - delimiters/subfield 139
 - envelopeIdentifier 429
 - envelopeIdentifier/receiver/ID 141
 - envelopeIdentifier/receiver/qualifier 142
 - envelopeIdentifier/sender/ID 140
 - envelopeIdentifier/sender/qualifier 141
 - FAGeneration/addGroup 129
 - FAGeneration/autoGenerateFA 129
 - FAGeneration/FALevel 129
 - FAGeneration/generateControlNumber 130
 - FAGeneration/processDocument 135
 - FAGeneration/rejectionRules/childTransactionRejectedStatus 133
 - FAGeneration/rejectionRules/logicalErrorStatus 132
 - FAGeneration/rejectionRules/syntaxErrorStatus 131
 - FAReconciliation 136, 321, 363
 - GSRouting/receiverQualifier 122
 - GSRouting/routingMode 120
 - GSRouting/senderQualifier 121
 - ICheaderInfo 328, 429
 - ICheaderInfo/GS/GS07 143
 - ICheaderInfo/ISA/ISA01 142
 - ICheaderInfo/ISA/ISA02 142
 - ICheaderInfo/ISA/ISA03 143
 - ICheaderInfo/ISA/ISA04 143
 - ICheaderInfo/ISA/ISA11 143

- ICheaderInfo/UNB/UNB01 144
ICheaderInfo/UNB/UNB06 144
ICheaderInfo/UNB/UNB08 144
processing with Trading Networks 196, 222
processingMode 122
publishBatchFailEvent 140, 159, 319, 327
splitOption 119, 388
TRADACOMS/enforceApplicationReference 157
TRADACOMS/Outbound/applicationReference 296
TRADACOMS/Outbound/applicationReferenceFromFile 296
TRADACOMS/Outbound/createReconciliationMessage 295
TRADACOMS/Outbound/recipientTransmissionReference 295
TRADACOMS/splitOption 153
 use during outbound processing 295
TRADACOMS/storageOption 154
UNAmode 138, 328, 429
wm.b2b.edtn.batch:batchProcess service 327, 328
EDITRACKING table
 FAReconciliation 364
 updates during batching EDI documents 321
EDIUseNewRelationshipLabel flag, setting 380
Element setting
 FA Level field 395
 FAGeneration/FA Level EDITPA variable 260
elements
 definition in SEF file 29
 mapping 80
enabling FA reconciliation 363
encoding input variable 304
endTradacomsTransmission service 290
Envelope CntrlNum Status custom attribute 201, 224
envelope validation, wm.b2b.edi:envelopeProcess service 47
envelopeDocuments variable 211, 212, 232, 233
envelopelIdentifier EDITPA variable, and the
 wm.b2b.edtn.batch:batchProcess service 429
envelopelIdentifier/receiver/ID EDITPA variable 141
envelopelIdentifier/receiver/qualifier EDITPA variable 142
envelopelIdentifier/sender/ID EDITPA variable 140
envelopelIdentifier/sender/qualifier EDITPA variable 141
envelopeProcess service, *See* **wm.b2b.edi:envelopeProcess** service
 service 450
Error & Continue setting
 action when invalid control number 170, 241
ControlNumberManagement
 duplicateControlNumberAction EDITPA variable 389
 outOfSequenceControlNumberAction EDITPA variable 389
Duplicate control number action field 393
Out of sequence control number action field 393
Error FA status, FA reconciliation 365
errors
 array and generateFA service 67
 convertToValues service 90
 FA transaction 58
 maximum number to report per FA transaction 58
 recognition 208, 228
variable in the BizDocEnvelope, 209, 229
wm.b2b.edi:convertToValues 217
wm.b2b.edi:envelopeProcess service 216
 wm.pub.flow:debugLog service 216
Errors variable in pipeline 211, 232
example of
 ANSI X12 810 document, looping 48
 ANSI X12 document, mapping 50
 conventions for VANs 340
 converting XML document into outbound EDI (ANSI X12) document 92
 document that processes an internal-format document (XML format) 85
 EDI Module setting system attributes 197
 flow services in sampleServices folder 53
 input variables for
 pub.client:ftp service 36
 pub.client:http service 34
 pub.client:http service, with wm.tn:receive service 188
mapping
 ANSI X12 850 document 80
 N1segment 74
 XML data to segments of EDI documents 85
sampleServices:
 Iterator810 service 53
 UNEDIFACTtoValues service 53
 X12toValues service 53
segment mapping 73
service that sets attribute values at run time 205, 226
Tutorial.EDItoXML:processEDI850_4101 service 53
Tutorial.EDItoXMLfolder, segment mapping 73

- Execute a Service processing action 204, 209, 225, 230
 batch EDI document 333
 delivering batch EDI document 332
 FA status unacceptable 281
 invalid control number 249
 outbound FA 279
 processing inbound EDI document 194, 220
 Extended Criteria for processing rules 208, 228, 331, 333
 external ID types
 and GSRouting variables 120
 defined 114
 DUNS (Trading Networks) 114
 for EDI ID qualifiers 114
 Mutually-Defined (Trading Networks) 114
 external IDs
 and GSRouting variables 120
 when defining profiles 113
 external IDs, TRADACOMS
 when defining profiles 147
 externalFormatString, format service 28
- F**
- FA Errors, FA reconciliation 365
 FA from date field, FA reconciliation report 369
 FA Level field 260, 395
 FA menu of EDI Module home page 368
 FA reconciliation
 See FAReconciliation EDITPA variable
 and batching 321
 disabled 137
 enabled 363
 reports, *See* FA reconciliation report
 statuses in report 365
 FA reconciliation report
 deleting 370
 fields 371
 HTML sample of 370
 Status field 370
 viewing 370
 WmEDIforTN\Pub\FAReports directory 370
 FA statuses
 acceptable 274
 FA reconciliation report
 Accept 365
 Accept with Errors 365
 Disable 137, 365
 Duplicate 365
 Duplicate FA/Errors 365
 Error 365
 FA Errors 365
 Interchange Received 365
 None 365
 Partially Allowed 365
 Reject 365
 how determined for automatic FA generation 263
 how generateFA service determines 59
 processing rule to handle documents with unacceptable 279
 reported in FAs
 Accepted 56, 263
 Accepted, But Errors Were Noted 56, 263
 FA 56, 263
 Not Allowed 56, 263
 Not Required 56
 Partially Accepted 56, 263
 Rejected 56, 263
 unacceptable 274
 values of, in EDITRACKING Table 365
 FA Timestamp FA reconciliation report field 372
 FA to date field, FA reconciliation report 369
 FA validation code 69
 FA, validation code for FAs 69
 FAGeneration
 addGroup EDITPA variable 129
 autoGenerateFA EDITPA variable 129
 use during outbound processing 294
 FALevel EDITPA variable 129
 generateControlNumber EDITPA variable 130
 processDocument EDITPA variable 135
 rejectionRules/childTransactionRejectedStatus EDITPA
 variable 133
 rejectionRules/logicalErrorStatus EDITPA variable 132
 rejectionRules/syntaxErrorStatus EDITPA variable 131
 fail delivery task status 319
 false, setting
 applyFormat format service 28
 FAReconciliation EDITPA variable 363, 364
 isEnabled format service 27

- FAReconciliation EDITPA variable 136, 363
and batching 321
and EDI recognizer 293
and splitOption value 119
- FAs
997 documents 56, 362
automatic generation of FAs
child transaction rejected status 267
control number in FA 261
FA status, how determined 263
generated FA description 262
generating based on indicator flag in interchange header 260
group FA status, determining 270
how child transaction rejected status set 267
how EDI Module processes documents based on FA status 272
level to acknowledge 260
logical error status 265
Not Allowed status 264
setting up 258
steps to generate 276
syntax error status 264
transaction FA status, determining 268
turning on or off 259
UN/EDIFACT interchange FA status, determining 271
automatically generating 258
child transaction
 whether allowed in group 59, 264
 whether any rejected 62, 267
child transaction rejected status, determining 62, 267
configuring maximum number of transaction errors 58
CONTRL message 56, 362
creating FA reconciliation reports 368
document level to acknowledge 57
errors, configuring maximum number to report 58
FA statuses, acceptable 274
FA statuses, unacceptable 274
FAReconciliation 364
generated FA description 262
generating
 FA reconciliation report 363
FAs automatically 258
FAs manually 56
FAs when processing documents all at one time 66
FAs when processing documents iteratively segment by segment 68
instance IDs for 356
logical error status, determining 61, 265
Not Allowed, determining status 59, 264
output from `wm.b2b.edi.util:generateFA` service 69
prerequisites for generating 57
processing rule to process outbound 278
reconciliation reports, *See* FA reconciliation report
reconciliation with EDI documents 362
service for creating 56
status of FA, determining status 60, 264
status, *See* FA statuses
syntax error status, determining 60, 264
through Trading Networks 363
validation codes 69
`wm.b2b.edi.util:generateFA` service 56
`wm.b2b.edtn.FAReport:generateFA` service 363
- Field field 401
field separator 401
 in outbound EDI document 138
- field values
 converting 77
 how to convert 26
- file control number
 validating 155
- File documents, TRADACOMS
 DocumentID system attribute 223
 Execute a Service processing action 231
 GroupID system attribute 223
 sample code for 234, 236
 system attribute 223
 viewing 376
- file headers
 control numbers, validating 155
 validating file control numbers 155
- File Polling
 listener port 37
 non-mainframe EDI documents 187
 to submit documents to Trading Networks 191
 `wm.tn.receive` service 188
- File Polling listener port 37, 191

- flat file schemas
 creating for
 each type of EDI document 20, 105
 internal-format document 24
 IS document for transaction set 356
 creating from
 IDOC 23
 SAP IDOC 23
 customizing
 iterative processing 47
 with Trading Networks 106
 defining
 EDI document structure 20
 internal-format document structure 40
 EDIFFSchema folder 104, 105
 EDIFFSchema.Standard.Vversion:Ttransaction 355
 format services 25, 29
 installing 104
 migrating earlier templates 23
 naming convention 22
 SEF file and `wm.b2b.edi:SEFParse` service 29
 TN document type 105
 Trading Networks 103
 with multiple consecutive HL segments 105
 format services
 `applyFormat` setting 28
 associating with
 EDI data types 29
 fields in flat file schema 25
 converting field values 25
 defined 25
 `externalFormatString` setting 28
 flat file schemas 29
 `internalFormatString` setting 28
 `isEnabled` setting 27
 name setting 26
`format.xml` configuration file
 associating EDI format services with EDI data types 29
 configuring how to convert field values in documents 25
 defined 25
 entries 26
 From Control Number Table setting of Generate Control Number field 397
 From date, FA reconciliation report field 369
 From Inbound Document setting of Generate Control Number field 397
 FromControlNumberTable setting of FAGeneration/generateControlNumber EDITPA variable 261
 FromInboundDocument setting
 FAGeneration/generateControlNumber EDITPA variable 261
 FTP client
 creating 35
 logic to include 35
 non-mainframe EDI documents 187
 wm.tn:receive service 188
 FTP listening port 189
 FTP method
 port 8021 35
 submitting EDI documents 35
 to IS FTP port 35
 to Trading Networks via FTP port 189
 functional acknowledgments (FAs), see *FAs* 56

G

- GEIS VAN 336
 Generate Control Number field 261, 397
 generateFA service, *See* `wm.b2b.edi.util:generateFA` service
 generating
 FA reports 363
 functional acknowledgments (FAs)
 automatically 258
 by invoking generateFA service 66
 prerequisites to invoking generateFA service 57
 turning automatic generation on or off 259
 interchange headers in batch EDI document 328
 getEDIstring service, *See* `wm.b2b.edi.util:getEDIstring` service 45
 getFromVAN service, *See* `VAN.VANConnectivity:getFromVAN` service
 getReport input variable to getFromVAN service 336
 getTspace service, *See* `wm.b2b.editn:getTspace` service
 GP mode, defined 424
 group
 and interchange headers, *See* group and interchange headers
 control numbers
 compliance check 47
 validating 125

- counts, compliance check 47
envelopes, *See* group envelopes
FA status of FA 60, 264
FA status, determining 64, 270
of segments, converting to IData object 51
processing 110
sender and receiver pairs, *See* group-level sender/receiver pair
group and interchange headers
adding to outbound EDI document 290
control numbers 238
control numbers for outbound EDI documents 284
envelopeIdentifier EDITPA variable 429
final batch EDI document 320
validating group control numbers 125
validating interchange control numbers 124
Group CntrlNum Status custom attribute 201, 224
Group documents
ANSI X12 business process 347
ANSI X12 FA reconciliation 362
conversation ID 198, 346
DocumentID system attribute 197
Execute a Service processing action 210
GroupID system attribute 198
GS Only variable 120
GS&ISA variable 121
instance IDs 353
OFF variable 121
sample code for 215
system attribute 197
viewing 376
group envelopes
interchange-level sender/receiver pair 390
wm.b2b.edi.util:addGroupEnvelope service 90
Group ID, FA reconciliation report field 371
Group Type field
control number setting 180, 184
FA reconciliation report field 370, 371
group type, determining based on transaction type 316
Group Version, FA reconciliation report field 371
Group, splitOption variable
outbound processing of EDI documents 293
processing Group document 215
setting up Trading Networks to deliver EDI document 297
groupDocuments variable 212, 233
GroupID system attribute 197, 223, 304
group-level sender/receiver pairs 404
and interchange-level sender/receiver pair 404
partner information to define 112
viewing 407
grpCtlNumber variable 290
GS Receiver Qualifier field 392, 411
GS Sender Qualifier field 392, 411
GSRouting/receiverQualifier EDITPA variable 122
when routingMode is GSOnly 122
GSRouting/routingMode
and GSRouting/receiverQualifier 122
and GSRouting/senderQualifier 121
GSRouting/routingMode EDITPA variable 120
senderQualifier variable 121
GSRouting/senderQualifier EDITPA variable 121
when routingMode is GSOnly 121
GXS VAN 336

H

- HL segments, multiple consecutive HL segments with same name 105
home page, EDI Module
Configuration menu 26, 58, 94, 149, 366, 380
Configure Field Formats 26, 29
Configure Properties 58, 94, 149, 366, 380
Control Numbers menu 178, 181
Create FA Reports 368
Define Instance ID Queries 354
Doc Exchange menu 104, 108
EDI Processes menu 354, 357
FA menu 368
Format Settings screen 26, 29
Install External ID Types 114
Install TN Document Types 104, 108
Manage Control Numbers 178, 181
Manage Interchange Info 391, 403, 406
Partner Set Up menu 114, 391, 403, 406, 407
Properties Settings screen 58, 94, 149, 366, 380
Query Envelope IDs 407
Run a Demo 357

- HTTP 188
 HTTP client
 logic to include 33
 non-mainframe EDI documents 187
 requirements for 33
 wm.tn:receive service 188
 HTTP method
 and EDI line break omission 34
 submitting EDI documents to IS 33
 HTTP POST method 33, 34
 to wm.tn:receive service 188
 HTTPS client
 non-mainframe EDI documents 187
 wm.tn:receive service 188
- I**
- IC mode, defined 424
 IC&GP mode, defined 420
 ICC.NET VAN 336
 IHeaderInfo EDITPA variable,
 wm.b2b.editn.batch:batchProcess service 328, 429
 IHeaderInfo/GS/GS07 EDITPA variable 143
 IHeaderInfo/ISA/ISA01 EDITPA variable 142
 IHeaderInfo/ISA/ISA02 EDITPA variable 142
 IHeaderInfo/ISA/ISA03 EDITPA variable 143
 IHeaderInfo/ISA/ISA04 EDITPA variable 143
 IHeaderInfo/ISA/ISA11 EDITPA variable 143
 IHeaderInfo/UNB/UNB01 EDITPA variable 144
 IHeaderInfo/UNB/UNB06 EDITPA variable 144
 IHeaderInfo/UNB/UNB08 EDITPA variable 144
 ICValues variable 67
 ID Pair Information section 178
 ID Qualifiers
 See also EDI ID qualifiers
 Receiver ID 391, 403, 406, 407
 Sender ID 391, 403, 406, 407
 IData objects
 converting 25
 EDI documents to 98
 EDI transaction set to 46
 from large documents iteratively 344
 internal-format document to String format 46
 creating from EDI documents 51
 group of segments converting to 51
 segment converting to 51
 unparsed section converting to 52
 validating internal-format document 91
 IDOC
 flat file schema from 20, 23
 using to create flat file schemas 23
 inbound EDI documents
 and FA reconciliation 136
 and generating FAs 56
 and mapping 72
 control number validation 166
 flat file schemas to parse and validate 103
 Inbound Information - Control Number Validation fields 393
 Inbound Information - FA Generation fields 395
 Inbound Information fields 392
 non-standard processing 196, 222
 processing rules 206, 227
 processing using Trading Networks 194, 220
 receiving and processing 40
 retrieving from VAN 336
 sample logic for processing 43
 standard processing 196, 222
 variables for processing 196, 222
 Inbound Information fields 392
 Control Number Validation 393
 FA Generation 395
 increment, control number
 default 172
 defining 177
 definition 172
 setting 177
 Initialization Service, EDITPA 118, 152
 input variables
 controlNumber 304
 createEDIBizDoc Service 303
 ctlFromTable 290
 delimiters 428
 docProcessingSvcName variable 83
 documentTypeName 84
 encoding 304
 getReport 336

- grpCtlNumber 290
mode 428
N1Segment service, example of 74
oneBatchQueue 310, 321, 427
 illustration of MULTIPLEOUTPUTS setting 324
 illustration of SINGLEOUTPUT setting 322
OutboundEDIDoc 303
PGPEnable 337
pub.client:ftp service 36
pub.client:http service 34, 188
receiver 303
receiverID 315, 316, 317, 318, 323, 428
receiverIDQualifier 315, 316, 318, 323, 428
receiverQualifier 303
saveInboundToTN 336
sender 303
senderID 315, 316, 317, 318, 323, 428
senderIDQualifier 315, 316, 318, 323, 428
senderQualifier 303
Set Inputs for scheduled delivery queue 330
standard 303
Tutorial.EDItoXML:dateParse service 77
VAN.VANConnectivity:getFromVAN service 336
VAN:VANConnectivity:getFromVAN service, conventions 340
wm.b2b.edi.util:generateFA service 69
wm.b2b.edi:envelopeProcess service 47
wm.b2b.editn.batch:batchProcess service 326
InputStream
 application/EDI stream content type 32, 41, 187
 converting to IData object 46
 sending documents as 97
installing TN document types and flat file schemas for transactions 104
instance ID 353
 defining query for Transaction document 354
 defining query for Transaction document, prerequisite 354
FAs 356
Group document 353
Interchange document 353
query 346
Transaction document 353
Integration Server
 client 32
 sending EDI documents to for processing 32
interchange control number
 compliance check 47
 validating 124
Interchange documents
 conversation ID 198, 346
 DocumentID system attribute 197
 Execute a Service processing action 210
 GroupID system attribute 197
 GS Only variable 120
 instance IDs 353
 OFF variable 121
 processing 110
 system attribute 197
 UN/EDIFACT business process 347
 UN/EDIFACT FA reconciliation 362
 viewing 376
interchange envelopes
 creating batch EDI document with multiple 310
 creating batch EDI document with one 310
 persistMultipleDocEnvelope EDITPA variable 123
 validation 47
 for processing document iteratively 52
wm.b2b.edi.util:addICEnvelope service 90
wm.b2b.edi:envelopeProcess service 47
interchange headers
 batching when not headers not present in documents 315
 control numbers 238
 control numbers, validating 124
 final batch EDI document 320
 information for outbound EDI document 284
Interchange Received FA status, FA reconciliation 365
interchange segments, splitOption variable 119, 388
interchange sender/receiver pairs
 and group-level sender/receiver pairs 404
 defining 390
 defining for non-standard processing 390
 editing and deleting 403
 partner information to define 112
 settings that affect inbound processing 411
Interchange setting, splitOption EDITPA variable
 outbound processing of EDI documents 293
 setting up Trading Networks to deliver EDI document 296
interchanges, UN/EDIFACT
 FA status, determining 65, 271

- interconnectivity to another VAN 341
- internal-format documents
- converting 24
 - converting from IData object to String object 46
 - creating
 - EDI document from 88
 - flat file schemas for 24
 - flat file schema that defines structure 40
- IData object 46, 91
- mapping
- data to format 41
 - information from line item segments into, example of 50
 - setting up Trading Networks to process 291
- internalFormatString, format service 28
- invoking 363
- individual segment-mapping services 73
 - mapping service 83
 - services to add group and interchange headers 290
- wm.b2b.edi.util:generateFA service after
- wm.b2b.edi:envelopeProcess service 67
- wm.b2b.edi.util:generateFA service before
- wm.b2b.edi:envelopeProcess service 66
- wm.tn.doc.xml:routeXml service 298
- IS document types
- and mapping EDI documents 72
 - creating for transaction 355
 - EDITPA 117, 118, 151, 152
 - for internal-format document structure 88
 - instance ID query 355
 - Tutorial.Records:POSource 85
 - wm.b2b.editn.rec:batchFailDocument 320
 - wm.b2b.editn.TPA:EDITPA 117, 118, 151, 152
 - wm.tn.rec:BizDocEnvelope IS 211, 232
- IS documents
- converting to 25
 - creating document lists from 85
- ISA01 field 402
- ISA011 field 402
- ISA012 field 402
- ISA014 field 402
- ISA02 field 402
- ISA03 field 402
- ISA04 field 402
- isEnabled, format service 27
- iteratively
- converting documents to IData objects 94, 98
 - converting large documents to IData objects 344
 - processing document, segment by segment 42, 47, 68
- iterator variable
- converting EDI documents to IData objects iteratively 98
 - processing document segment by segment 47
- J**
- Java service, converting field values 77
- K**
- keys, server log 374
- L**
- large documents
- and the EDI Module configuration file 97
 - converting to IData objects iteratively 344
 - iterator variable, convertToValues service 47
 - memory constraints 94
 - processing 94
 - using Trading Networks 344
 - services in WmEDI package 344
 - with Trading Networks 344
 - wm.b2b.edi:envelopeProcess service 97
- line breaks, omitting from EDI documents 34
- logging information
- activity log 374
 - server log 374
- logic to include in
- FTP client 35
 - HTTP client 33
- sample to process
- inbound EDI document 43
 - Transaction document 213
- service
- that Execute a Service action invokes 210, 231
 - to form EDI document 89
 - to process EDI document iteratively 50
 - to process EDI documents 41
 - to validate internal-format document before converting to EDI document 91

- service that forms the EDI document to route it through Trading Networks 302
- service that forms the EDI document to submit it to Trading Networks 298
- to add UN/EDIFACT envelopes to EDI document 91
- logical error status
- and FA status for group 64, 270
 - and FA status for transaction 63, 268
 - and FA status for UN/EDIFACT interchange 65, 271
 - description 61, 265
- Logical Error Status field 266, 398
- looping
- example of in N1Segment service 75
 - sample code for processing entire inbound EDI document at one time 44
- M**
- mapping
- and IS document types 72
 - ANSI X12 812 document, example of 50
 - changing the properties file 86
 - concatenating strings 78
 - data
 - prerequisites 72
 - to format an internal-format document 41
 - description 72
 - EDI document data to another format 73
 - EDI-to-XML mapping examples 73
 - element referencing tips 80
 - examples of XML data to EDI format 85
 - invoking mapping service 83
 - preventing redundancies 86
 - replacing control characters 86
 - segment mapping, *See* segment mapping
 - service, *See* mapping service
 - transforming source data before 77
 - truncating characters 85
 - when using addlCEnvelope service 289
 - XML documents to EDI format 84
- mapping service
- ANSI X12 850 document, example of 80
 - best practices
 - from EDI to XML 79
 - from XML to EDI 85
- EDI document 73
- EDI to XML 80
- segment, *See* segment-mapping services
- specifying input/output 79
- XML to EDI 84, 85
- maximum number of transaction errors 58
- Maximum Results
- control number information 184
 - group-level sender/receiver pairs 403, 406, 407
- MCI VAN 336
- memory constraints, large documents 94
- migrating
- 4.x or earlier templates to flat file schemas 24
 - earlier templates to flat file schemas 23
- minimum, control number
- default 172
 - defining 177
 - definition 172
 - setting 177
- mode input variable of `wm.b2b.editn.batch:batchProcess` service 428
- Modifiable Data Status of EDITPA 116, 150
- modifying default EDITPA 116, 150
- monitoring business process 357
- Monitoring Directory field 37
- multiple queues, batched EDI documents 419
- MULTIPLEOUTPUTS
- illustration 324
 - oneBatchQueue variable 310
- Mutually-Defined external ID type, Trading Networks 114
- N**
- N = Not Allowed validation code 69
- N1 segment of an ANSI X12 850 document 73
- N1Segment flow service, example of 74
- name
- FA reconciliation report file 368
 - format service 26
- namespace convention
- flat file schema
 - ANSI X12, UN/EDIFACT 105
- namespace directory (ns) 35
- NEW processing status on Transaction Analysis screen 378

- None FA status, FA reconciliation 365
 non-standard EDI formats, and flat file schema 21
 non-standard processing
 advantages of 111
 defined 110, 386
 defining interchange-level sender/receiver pairs 390
 disadvantages of 111
 partner information to define 386
 sample code for forming document 413
 sender/receiver pairs 390, 404
 settings for interchange sender/receive pairs 411
 Not Allowed validation code for FAs 69
 Not Rejected setting
 FAGeneration/processDocument EDITPA variable 274, 275
 Process Document field 396
 NR validation code 69
 NR, validation code for FAs 69
 number of scheduled delivery queues 329
- O**
- Off setting
 Auto Generate FA field 395
 FAGeneration/autoGenerate EDITPA variable 260
- On setting
 Auto Generate field 395
 FAGeneration/autoGenerate EDITPA variable 259
- oneBatchQueue input variable 321
 description of settings 310
 illustration of SINGLEOUTPUT setting 322, 324
 MULTIPLEOUTPUTS 310
 null 310
 SINGLEOUTPUT 310
 wm.b2b.editn.batch:batchProcess service 427
- Only Accepted setting
 FAGeneration/processDocument EDITPA variable 274, 275
 Process Document field 396
- Out of sequence control number action field 394, 409, 411
 outbound EDI documents
 adding group and interchange headers to 290
 and FA reconciliation 136
 and queue 338
 BizDocEnvelope 303
 built-in services to form 284
- composite elements separator 139
 control numbers 168, 290
 control numbers, TRADACOMS 168, 291
 created by batching feature 138
 delimiters and header information 414
 delimiters/record variable 138
 delivering
 batch documents when using 6.0.1 batching 426
 delivery methods 291
 directly from service that forms it 292
 to VANs 336
 field separator 138
 flow service examples 92
 mapping documents 72
 receiver ID and envelopelIdentifier/sender/ID variable 141
 release character 139
 routing through Trading Networks 301
 segment terminator 138
 sender ID and envelopelIdentifier/sender/ID variable 140
 sending to Trading Networks from internal applications 284
 submitting back to Trading Networks 293
 VANFTP service signed and PGP encrypted 339
 variables for processing 293
 variables for processing TRADACOMS documents 295
 wm.b2b.editn:trackEDIdocs service to record in
 EDITRACKING table 136
- Outbound Information
 Delimiters field 401
 Envelope Information field 402
 FA Generation fields 400
- OutboundEDIDoc input variable 303
 out-of-sequence control numbers
 defining action for 171, 172
 definition 176
 reprocessing documents with 250
 service to reprocess documents with 252
 user status 242
- outputs
 bizdoc variable 304
 for N1Segment service, example of 74
 from wm.b2b.edi.util:generateFA service 69
 Tutorial.EDItoXML:dateParse service 77
 overriding pre-processing settings with processing rule settings 208, 229

P

P = Partially Accepted validation code 69

packages

WmEDI, *See* WmEDI package

WmEDIsamples, *See* WmEDIsamples package

WmFlatFile 27

Partially Accept FA status, FA reconciliation 365

Partially Accepted setting

Child Transaction Rejected Status field 399

Partially Accepted validation code for FAs 69

partner information

non-standard processing 386

standard processing 112, 146

Partner Set Up menu of EDI Module home page 114, 391, 403, 406, 407

partner-specific EDITPA 118, 152

controlling batchProcess service when using version 6.0.1
batching 429

defining 117, 151, 388

Per Document setting

Auto Generate FA field 395

FAGeneration/autoGenerate EDITPA variable 260

PERL scripts 33

persistMultipleDocEnvelope EDITPA variable 123, 158

PGPEnable input variable to getFromVAN service 337

PGP-encryption 337

pipeline

after setting scope for the INVOKE routeXml flow operation
300

Attributes variable 211, 232

bizdoc variable 211, 232

BizDocEnvelope 211, 232

diagram of, after setting scope for INVOKE routeXML flow
operation 300

DocumentID variable 211, 232

edidata variable

Errors variable 211, 232

information in that your service can access 211, 232

using scope to temporarily limit pipeline variables 300

port 191

8021 for FTP 35

File Polling listener 191

FTP listening 189

number 5555 34, 188

of URL service to process EDI document 34

POST an EDI document to

a service 34

wm.tn:receive service 188

pre-processing actions

default 208, 229

when invalid control number 249

prerequisites for

batching EDI documents

using 6.0.1 version 426

using Trading Networks 326

creating service to

form EDI documents 88, 285

process EDI documents 40

defining instance ID query for transaction 354

generating functional acknowledgments (FAs) 57

mapping data 72

processing EDI documents 20

using Trading Networks 102, 195, 221

private queues 330

procedures

adding

control number information 177

external ID types to Trading Networks for EDI ID qualifiers
114

associating

custom attribute with EDI document 204, 225

EDI format services with EDI data types 29

configuring

EDI Module for large document handling 94

how to convert field values in documents 26

maximum number of transaction errors 58

creating

customized services to connect to other VANs 341

FA reconciliation reports 368

flat file schema from a SAP IDOC 23

flat file schema from a SEF file 22

IS document type for transaction set from flat file schema
356

partner-specific EDITPA 117, 151

profiles 113, 148

customizing the EDI flat file schema for iterative processing 48
defining

an instance ID query for transaction 354

control number settings 177

- group-level sender/receiver pair associations 406
- information for interchange-level sender/receiver pair 391
- processing rule 206, 227
 - batch EDI documents 331
 - deliver final batch EDI document 332
- scheduled delivery queues 329, 432
- scheduled delivery service for delivering EDI documents to VANs 338
- editing and deleting interchange-level sender/receiver pair information 403
- migrating 4.x or earlier templates to flat file schemas 24
- modifying default EDITPAs 116, 150
- querying for existing group-level sender/receiver pair associations 407
- searching for existing control numbers 181
- to run sample business process 357
- viewing
 - EDI documents using Trading Networks Console 377
 - information about queued EDI documents 379
 - server log 374
 - Trading Networks activity log 374
- Process Document field 275, 396
- process models 346
 - defining 347
- Process Normally setting
 - action when invalid control number 170, 241
 - Duplicate control number action field 393
 - Out of sequence control number action field 393
- processing
 - EDI documents, *See* processing EDI documents
 - Group documents, sample code 215
 - inbound EDI documents using Trading Networks 194, 220
 - internal-format documents with Trading Networks 291
 - iteratively 47
 - large documents 94
 - using Trading Networks 344
 - non-standard, *See* non-standard processing
 - standard, *See* standard processing
 - status, viewing on Transaction Analysis screen 377
 - TRADACOMS File documents, sample code 234, 236
 - Transaction document, sample code 213
- processing actions 209, 230
 - documents with unacceptable FA statuses 281
 - outbound FA 279
 - when invalid control number 249
- processing EDI documents
 - entire document at one time 42, 43, 66
 - advantages of 42
 - disadvantages of 42
 - sample code 43
 - inbound document iteratively 50
 - iteratively, segment by segment 42, 47, 68
 - prerequisites 20
 - with Trading Networks 102
 - segment by segment 42, 47, 68
 - advantages of 42
 - disadvantages of 42
 - using Trading Networks, prerequisites 195, 221
 - ways to 41
- processing rules 194, 220
 - and EDI Processing Mode custom attributes 122
 - and GSRouting variables 120
 - and Trading Networks attributes 196, 222
 - batch EDI documents 331
 - batching
 - EDI documents 331
 - EDI documents when using version 6.0.1 batching 433
 - criteria 207, 228
 - outbound FA 278
 - when invalid control number 248
 - deliver EDI documents to VANs 340
 - delivering batch EDI document 332
 - delivering batch EDI document when using version 6.0.1 batching 437
 - documents with unacceptable FA statuses 279
 - Execute a Service action 204, 225
 - inbound EDI documents 194, 220
 - overriding pre-processing settings 208, 229
 - pre-processing actions when invalid control number 249
 - process document with invalid control number 247, 279
 - process inbound EDI documents 206, 227
 - process outbound FAs 278
 - processing actions 209, 230
 - documents with unacceptable FA statuses 281
 - outbound FA 279
 - when invalid control number 249
 - setting values of attributes at run time 204, 225
 - Trading Networks 194, 220
 - use custom attribute as criteria 205, 226
- Processing Service field 37

processingMode EDITPA variable 122
ProcessNormally setting of ControlNumberManagement
 duplicateControlNumberAction EDITPA variable 389
 outOfSequenceControlNumberAction EDITPA variable 389
Production Mode
 and IMB11 field of UN/EDIFACT document 123
 and ISA015 field of ANSI X12 document 123
 control number information 179, 183
 custom attribute 214, 218
 envelope information 391
 group-level sender/receiver pairs 406
 interchange-level sender/receiver pairs 403
 processingMode variable 122
profiles
 creating 113, 148
 defining 113, 147, 387
program code conventions in this document 15
properties file and redundant mapping 86
Properties Settings screen
 configuring large document handling 94
 configuring maximum number of errors in FA reports 58
 configuring TRADACOMS external ID types 149, 380
 controlling FA status for documents sent multiple times 366
properties.cnf file, EDI Module
 EDIBigDocThreshold property 95
 EDIMaxFATransactionErrors property 58
 EDIResolveDuplicates property 366
properties.cnf file, Trading Networks
 tn.tspace.location property 96
 tn.tspace.max property 96
Proposed, EDITPA Agreement Status 116, 150
pub.client:ftp service 36, 190
pub.client:http service 33
 example of, with wm.tn:receive service 188
pub.io:streamToBytes service
 processing File documents 236
pub.list:appendToDocumentList service
 processing File documents 235
pub.schema:validate service 91
pub.string:bytesToString service
 processing File documents 236
pub.string:stringToBytes service 305
pub.xml
 documentToXMLString service 84

pub.xml:
 loadXMLNode service 85
 xmlNodeToDocument service 85
 xmlStringToXMLNode service 85
public queue
 and defining scheduled delivery queue 329
 associated with scheduled deliver service 338
 public scheduled delivery queue, *See* scheduled delivery
 queues
 settings 329, 338
 VANS 337
publishBatchFailEvent EDITPA variable 140, 159, 319, 327

Q

Qualifier for group-level sender/receiver pairs 407
querying for existing group-level sender/receiver pair
 associations 407
Queue Name 338
 scheduled delivery queue 330
QUEUED processing status on Transaction Analysis screen 378
queues
 multiple, for batched EDI documents 419
 private 330
 public, *See* public queue
 scheduled, *See* scheduled delivery queues

R

R = Rejected validation code 69
Random setting
 FAGeneration/generateControlNumber EDITPA variable 261
 Generate Control Number field 397
Receiver
 EDITPA 116, 118, 151, 152
 processing rule critieria 207, 228
receiver
 and GSRouting/routingMode variable 120
 input variable for createEDIBizDoc Service 303
 Interchange, Group, Transaction documents 120
 interchange-level 390
 viewing on Transaction Analysis screen 377
Receiver field, Outbound Information - FA Generation 400

- Receiver ID
 adding control number information 179, 182, 410
 FA reconciliation report field 368, 371
 group-level sender/receiver pairs 406, 407
 interchange-level sender/receiver pairs 391, 403
- Receiver Qualifier
 adding control number information 179, 183, 410
 group-level sender/receiver pairs 406, 407
 interchange-level sender/receiver pairs 391, 403
- receiverID input variable 315, 316, 317, 318, 323
`wm.b2b.edtn.batch:batchProcess` service 428
- receiverIDQualifier input variable 315, 316, 318, 323, 428
- receiverQualifier input variable 303
- Recognition Errors processing rules critiera 208, 228
- reconciliation messages, TRADACOMS
 creating in outbound batched EDI documents 295
- Reject FA status, FA reconciliation 365
- Reject setting
 action when invalid control number 170, 242
 ControlNumberManagement
`duplicateControlNumberAction` EDITPA variable 389
`outOfSequenceControlNumberAction` EDITPA variable 389
 Duplicate control number action field 394
 Out of sequence control number action field 394
- Rejected setting
 Child Transaction Rejected Status field 399
`FAGeneration/rejectionRules/logicalErrorStatus` EDITPA variable 266
`FAGeneration/rejectionRules/syntaxErrorStatus` EDITPA variable 264
 Logical Error Status field 398
`logicalErrorStatus` input variable of `generateFA` 61
 Syntax Error Status field 397
`syntaxErrorStatus` input variable of `generateFA` 60
- Rejected validation code for FAs 69
- Related Doc ID, FA reconciliation report field 372
- related documents flag, setting 380
- release character 139
- Release field 401
- Report template, FA reconciliation report field 368
- reportFileName, FA reconciliation report file 368
- reports, FA *See* FAs
- requirements for
 HTTP client 33
 submitting EDI documents to Trading Networks via HTTP 188
 trading partner relationships, met in EDITPA 116, 150
- retrieving inbound EDI documents from a VAN 336
- routing outbound EDI documents through Trading Networks 301
- routingMode variable 120
- run time, setting values of custom attributes 204, 225
- S**
- sample
 business process 357
 code, *See* sample code
 FA reconciliation report 370
 process models, *See* sample process models
- sample code
 creating service when using non-standard processing 413
 delivering outbound EDI document from service that forms it 292
 forming EDI document using standard processing 286
- processing
 entire document at one time 43
 inbound EDI document iteratively 50
 Transaction document 213
- processing Group document 215
- submitting outbound EDI document back to Trading Networks 298
- sample process models
 ANSI X12 documents 347
 UN/EDIFACT Interchange documents 350
- sampleServices:
 folder, examples 53
 Iterator810 service 53
 UNEDIFACTToValues 45
 UNEDIFACTtoValues service 53
 X12toValue service
 diagram of 83
 X12toValues service 53, 83
`docProcessingSvcName` input variable 83
- SAP IDOC 20, 23
 files, *See* IDOC (SAP) files

- Save Document to Database pre-processing action 208, 229
 EDI documents 376
 setting when invalid control number 249
- savelnboundToTN, input variable to getFromVAN service 336
- Schedule, scheduled delivery queue 330
- scheduled delivery queues
 batching
 defining 329, 432
 delivery task 379
 number to define 329
 number to define when using version 6.0.1 batching 431
 VANs 337
 viewing information about EDI documents in queue 379
- scheduled delivery service associated with public queue 338
- searching for control numbers 181
- security header information 284
- SEF file
 creating flat file schema from 20
 flat file schema and `wm.b2b.edi:SEFParse` service 29
- segment
 converting to `IData` object 51
 counts, compliance check 47
 instance ID query 354
 mapping, *See* segment mapping
 processing document by 42, 47, 68
- Segment field 401
- segment mapping
 advantages of 73
 defined 73
 EDI-to-XML mapping examples 73
 services, *See* segment-mapping services
 XML documents 84
- Segment setting
 FA Level field 395
 FAGeneration/FA Level EDITPA variable 260
- segment terminator in outbound EDI document 138
- segmentMapping folder 74
- segment-mapping services 74
 EDI to XML 80
 N1 segment of EDI document 74
 XML to EDI 85
- selecting
 EDI documents to view on the Transaction Analysis screen 378
 processing rule, criteria 331, 332
- Sender
 EDITPA 116, 118, 151, 152
 processing rule criteria 207, 228
- sender
 and GSRouting/routingMode variable 120
 input variable for `createEDIBizDoc` service service 303
 Interchange, Group, Transaction documents 120
 interchange-level 390
 viewing on Transaction Analysis screen 377
- Sender field, Outbound Information - FA Generation 400
- Sender ID
 adding control number information 178, 182, 410
 FA reconciliation report field 368, 371
 group-level sender/receiver pairs 406, 407
 interchange-level sender/receiver pairs 391, 403
- Sender Qualifier
 adding control number information 178, 182, 410
 group-level sender/receiver pairs 406, 407
 interchange-level sender/receiver pairs 391, 403
- senderID input variable 315, 317, 318, 323
 `wm.b2b.editn.batch:batchProcess` service 428
- senderID input variables 316
- senderIDQualifier input variable 315, 316, 318, 323, 428
- senderQualifier input variable 303
- sending
 EDI documents to Integration Server 32
 using File Polling 37
 outbound document to Trading Networks from internal applications 284
- separator
 composite elements 139
 field 138
- server log 374
 keys 374
 viewing 374

services
 check envelope compliance 47
 create
 BizDocEnvelope and add outbound EDI document 303
 flat file schemas from SAP IDOCs 23
 customized to connect to another VAN 341
 delivery 330
 EDI Batch delivery service 330
 EDI Module, validate EDI envelopes 47
 form EDI document 88
 format, *See* format services
 getFromVAN service 336
 Initialization Service for EDITPA 118, 152
 large document in WmEDI package 344
 mapping, *See* mapping service
 processing 215, 218
 EDI documents 40
 Transaction and Group documents 210, 231
 pub.client:
 ftp service 36, 190
 http service 33
 pub.io:
 streamToBytes service 236
 pub.list:
 appendToDocumentList service 235
 pub.schema:validate service 91
 pub.string:
 bytesToString service 236
 pub.string:stringToBytes service 305
 pub.xml:
 loadXMLNode service 85
 xmlNodeToDocument service 85
 xmlStringToXMLNode service 85
 pub.xml:documentToXMLString service 84
 sampleServices:
 Iterator810 service 53
 UNEDIFACTtoValues service 53
 X12toValues service 53, 83
 segment-mapping, *See* segment-mapping services
 set attribute values at runtime 204, 225
 to process EDI documents when using File Polling 37
 Tutorial.EDItoXML.segmentMapping:N1Segment service 77
 Tutorial.EDItoXML:

EDItoPOXMLMap service 83
 processEDI850_4010 service 83
 processEDI850_4101 service 53
 Tutorial.XMLtoEDI.properties:getProperties service 86
 Tutorial.XMLtoEDI:
 appendSegment service 85
 conditionalTruncate service 85
 processXMLSource service 85, 92
 replaceControlChar service 86
 XMLPOtoEDIMap service 85
 validate EDI envelopes 47
 VAN.VANConnectivity:
 getFromVAN service 336, 341
 putToVAN (VANFTP) service 338, 341
 wm.b2b.edi.migration:getTemplate service 24
 wm.b2b.edi.tradacoms.compose:
 addToTradacomsTransmission service 290
 endTradacomsTransmission service 236, 290
 startTradacomsBatch service 290
 startTradacomsTransmission service 236, 290
 wm.b2b.edi.tradacoms.doc:
 getContentPart service 235
 getDocumentPartInfo service 235
 getFFSchemaNames service 235
 isFileEnvelope service 235
 wm.b2b.edi.tradacoms:
 convertToString service 25
 convertToValues service 25
 wm.b2b.edi.tradacoms:convertToString service 288
 wm.b2b.edi.tradacoms:convertToValues service 235, 287
 wm.b2b.edi.util:
 addGroupEnvelope service 90, 290
 addGroupEnvelopeEDIFACT service 290
 addIEEnvelope service 90, 288, 290, 298, 415
 addIEEnvelopeEDIFACT service 90, 91, 290
 EDIconcat service 51
 generateFA service 56, 69, 92
 getEDIstring service 45, 217
 invoke service 83
 wm.b2b.edi:
 convertToString service 25, 214, 288, 413
 migration 24

- convertToValues service 24, 25, 46, 48, 50, 89, 94, 97, 214, 217, 287, 415
createDOCtemplate service 23
envelopeProcess service 43, 47, 92, 97, 216
SEFParse service 21, 29
wm.b2b.edi:convertToValues service 218
wm.b2b.editn.batch:batchProcess service 310, 326, 332
wm.b2b.editn.crossRef:getEnvInfo service 414
wm.b2b.editn.TPA:
 getEDITPAData 287
 initService 118, 152
wm.b2b.editn.util.reprocess:
 listInSequence service 252
 listUnprocessDoc service 250
 nextInSequenceDoc service 256
 reprocessDocument service 251, 255
wm.b2b.editn.util:
 getContentPartDataAsInputStream service 235
wm.b2b.editn:
 ediPartnerIDToTNPartnerID service 305
 getTspace service 213, 216
 trackEDIdocs service 136, 292, 301, 302
wm.b2b.util.documentList:
 addDocToDocumentList service 235
wm.tn.doc.xml:routeXml service 298, 301
wm.tn.doc:
 addContentPart service 308
 createNewEnvelope service 307
 view service 253
wm.tndoctype:view service 306
wm.tn.route:routeBizdoc service 302
wm.tn:receive service 188
Set Inputs
 public queue setting 339
 scheduled delivery queue 330
setting up
 File Polling 37
 to batch EDI documents 329
Trading Networks to deliver EDI document based on
 splitOption variable 296
Trading Networks to deliver EDI documents 302
Trading Networks to process internal-format documents 291
version 6.0.1 batching 430
settings
 applyFormat (of format service) 28
 externalFormatString (of format service) 28
 for interchange sender/receiver pairs 411
 internalFormatString (of format service) 28
 isEnabled (of format service) 27
 name (of format service) 26
 public queue 329, 338
single pass, version 6.0.1 batching 419
 defining processing rules 433
 diagram of 419
 IC&GP mode 420
SINGLEOUTPUT
 illustration 322
 oneBatchQueue variable 310
Sort By
 FA reconciliation report field 370
 field for control number search 183
 group-level sender/receiver pairs 406, 407
 interchange-level sender/receiver pairs 403
specifying
 pre-processing actions when invalid control number 249
 processing actions 209, 230
 documents with unacceptable FA statuses 281
 outbound FA 279
 when invalid control number 249
 processing rules criteria 207, 228
 outbound FA 278
 when invalid control number 248
splitOption EDITPA variable 119, 388
 and FAReconciliation 119
 and Trading Networks 194
 business process 346
 documents saved to Trading Networks database 376
 Execute a Service processing action 209
 not used when invalid control number 242
 outbound processing of EDI documents 293
 possible values 119
 processing rule for inbound EDI documents 206
 use during outbound processing 293
ST/AK1/AK102 instance ID 356
Standard Exchange Format (SEF) file, *See* SEF file
Standard field, Control Number Information 179, 183

- standard input variable 303
- standard processing
 - advantages of 111
 - defined 110, 386
 - disadvantages of 111
 - forming EDI document, sample code 286
 - partner information to define 112, 146
 - turning control number validation on or off 168
- startTradacomsBatch service 290
- startTradacomsTransmission service 290
- Status field, FA reconciliation report 370
- statuses
 - delivery tasks associated with documents being batched 319
 - FA, *See* FA statuses
 - processing, view on Transaction Analysis screen 378
- statuses, FA
 - acceptable 274
 - FA reconciliation report
 - Accept 365
 - Accept with Errors 365
 - Disable 365
 - Duplicate 365
 - Duplicate FA/Errors 365
 - Error 365
 - FA Errors 365
 - Interchange Received 365
 - None 365
 - Partially Accept 365
 - Reject 365
 - processing rule to handle documents with unacceptable 279
 - reported in FAs
 - Accepted 56, 263
 - Accepted, But Errors Were Noted 56, 263
 - FA 56, 263
 - Not Allowed 56, 263
 - Not Required 56
 - Partially Accepted 56, 263
 - Rejected 56, 263
 - unacceptable 274
- String 46
 - application/EDI content type 41
 - application/UNEDIFACT content type 41
 - application/X12 content type 41
- concatenating multiple using variable substitution 78
- converting
 - EDI document to 32, 187
 - internal-form document from IData object to String format 46
 - to IData object 46
- structure
 - EDI document, array 85
 - of element definition 29
 - XML data, flat 85
- SubField 401
- submitting EDI documents
 - batched documents to Trading Networks for delivery 426
 - FTP method 35
 - HTTP method 33
 - to Trading Networks
 - File Polling 191
 - FTP method 189
 - HTTP method 188
 - HTTPS method 188
 - to deliver it 293
- success delivery task status 319
- syntax error status
 - and FA status for group 64, 270
 - and FA status for transaction 63, 268
 - and FA status for UN/EDIFACT interchange 65, 271
 - description 60, 264
- Syntax Error Status field 265, 397
- system attributes
 - ConversationID 198
 - defined 196, 222
 - DocumentID 197, 223, 304
 - GroupID 197, 223
 - GroupId 304
 - that EDI Module sets 197, 223
 - User Status 208, 228
- T**
- targetSchema 22
- Tasks screen 379
- templates
 - for webMethods EDI Module 4.x or earlier 23
 - migrating 4.x or earlier to flat file schemas 24

- Testing Mode
and IMB11 field of UN/EDIFACT document 123
and ISA015 field of ANSI X12 document 123
custom attribute 215, 218
envelope information 391
processingMode variable 122
- TN document types
assigning custom document attributes to 204, 225
FA reconciliation report field 371
flat file schema 105
installing 104
instance ID query 354
Trading Networks processing 194, 220
`tn.tspace.location` configuration property 96
`tn.tspace.max` configuration property 96
To date, FA reconciliation report field 369
TPA, *See* EDITPA
TRADACOMS EDI documents
adding outbound envelopes and obtaining control numbers 290
and storageOption variable 154
TRADACOMS/ControlNumberManagement
duplicateControlNumberAction EDITPA variable 155
outOfSequenceControlNumberAction EDITPA variable 156
validateInboundBatchControlNumbers EDITPA variable 154
validateInboundFileControlNumbers EDITPA variable 155
validateInboundTransmissionControlNumbers EDITPA variable 154
TRADACOMS/enforceApplicationReference EDITPA variable 157
TRADACOMS/Outbound/applicationReference EDITPA variable 296
TRADACOMS/Outbound/applicationReferenceFromFile EDITPA variable 296
TRADACOMS/Outbound/createReconciliationMessage EDITPA variable 295
TRADACOMS/Outbound/recipientTransmissionReference EDITPA variable 295
TRADACOMS/splitOption EDITPA variable 153, 295
and Trading Networks 220
documents saved to Trading Networks database 376
Execute a Service processing action 230
processing rule for inbound EDI documents 227
- TRADACOMS/storageOption EDITPA variable 154
- Trading Networks 194, 220
activity log 374
and EDI recognizer 194, 220, 291, 293
Console, *See* Trading Networks Console
database 376
delivering EDI document
 Group splitOption variable 297
 Interchange splitOption variable 296
delivery tasks and batching EDI documents 311, 315, 319
DocumentID system attribute 197, 223
DUNS external ID type 114
EDIFFSchema folder 104, 105
EDITPA variables 196, 222
FAs 363
flat file schemas 103
GroupID system attribute 197, 223
large document handling 344
Mutually-Defined external ID type 114
processing EDI documents 102, 186
processing inbound EDI documents 194, 220
providing EDI ID qualifiers 114
public queue, *See* public queue
routing outbound EDI documents through 301
submitting documents to via File Poling 191
TN document types 104, 194, 220
- Trading Networks attributes 196
and EDI documents 196, 222
custom attributes 196, 222
system attributes 196, 222
- Trading Networks Console
creating default EDITPA 116, 150
creating EDITPA 117, 151
profiles, creating 113, 148
Tasks screen 379
Transaction Analysis screen 376
- trading partners
creating profiles 113, 148
defining profiles 113, 147, 387
- transaction
control numbers, compliance check 47
counts, compliance check 47
FA status of FA 60, 264
FA status, determining 63, 268

- IS document types for 355
- Not Allowed FA status 59
- processing 110
- Transaction Analysis screen 376
- Transaction CntrlNum Status custom attribute 224
- Transaction documents
 - business processes involving EDI documents 347
 - conversation ID 198, 346
 - defining instanceID query 354
 - defining instanceID query, prerequisite 354
 - DocumentID system attribute 197
 - Execute a Service processing action 210
 - GroupID system attribute 198
 - GS Only variable 120
 - GS&ISA variable 121
 - instance IDs 353
 - OFF variable 121
 - processing, sample code 213
 - system attribute 197
 - viewing 376
- transactionDocument variable 212, 233
- TransactionSet setting
 - FA Level field 395
 - FAGeneration/FA Level EDITPA variable 260
- transforming source data before mapping to target 77
- transmission control number
 - validating 154
- Transmission documents
 - Execute a Service processing action 231
- Transmission documents, TRADACOMS
 - DocumentID system attribute 223
 - Execute a Service processing action 231
 - GroupID system attribute 223
 - system attribute 223
 - viewing 376
- transmission envelopes
 - persistMultipleDocEnvelope EDITPA variable 158
- transmission headers
 - control numbers, validating 154
 - validating transmission control numbers 154
- transmission segments, splitOption variable 153, 295
- transmission sender/receiver pairs
 - defining profiles for 146
- triggers, to handle notification of batching failure 320
- troubleshooting information 16
- true, setting
 - applyFormat format service 28
 - FAReconciliation EDITPA variable 363, 364
 - isEnabled format service 27
- truncating characters when mapping from XML to EDI 85
- tspace 94
- Tutorial.EDItoXML folder
 - segment mapping example 73
- Tutorial.EDItoXML.segmentMapping:N1Segment service
 - diagram of 77
 - example of 80
- Tutorial.EDItoXML:dateParse service
 - diagram of 78
 - inputs and outputs 77
- Tutorial.EDItoXML:EDItoPOMap service, diagram of 81
- Tutorial.EDItoXML:EDItoPOXMLMap service 83
- Tutorial.EDItoXML:EDItoXMLPOMap service
 - diagram of 82
 - diagram of final map 82
 - example of 80
- Tutorial.EDItoXML:processEDI850_4010 service 83
 - diagram of 84
- Tutorial.EDItoXML:processEDI850_4101 service 53
- Tutorial.Records:POSource IS document type
 - structure of XML data 85
- Tutorial.XMLtoEDI
 - processXMLSource service 92
- Tutorial.XMLtoEDI folder
 - examples of mapping XML documents to EDI format 85
- Tutorial.XMLtoEDI.properties:getProperties service 86
- Tutorial.XMLtoEDI:appendSegment service 85
- Tutorial.XMLtoEDI:conditionalTruncate service 85
- Tutorial.XMLtoEDI:processXMLSource service 85
- Tutorial.XMLtoEDI:replaceControlChar service 86
- Tutorial.XMLtoEDI:XMLPOtoEDIMap service 85

two passes, version 6.0.1 batching 421, 422
 defining processing rules 433
 diagram of 422
 GP mode 424
 IC mode 424
`wm.b2b.edtn.batch:batchProcess` service 424
 typographical conventions in this document 15

U

`UCI/03` instance ID 356
UN/EDIFACT
 business process 347
 standard, `CONTRL` message 56
UN/EDIFACT CONTRL
 message 56, 57
`UCI/03` instance ID 356
UN/EDIFACT EDI document
 adding envelope 91
 EDI Processing Mode, custom attribute 123
 FA reconciliation 362
`wm.b2b.edi.util:addICEnvelopeEDIFACT` service 90
UN/EDIFACT Interchange document
 FA status 365
 FAReconciliation 364
 sample process models 350
UN/EDIFACT UNB header 402
`UNAmode` EDITPA variable 138, 328, 429
 UNB01 field 402
 UNB06 field 402
 UNB07 field 402
 UNB08 field 402
 UNB09 field 402
 UNB10 field 402
 UNB11 field 402
 unDefData element 97
 unparsed section, converting to `IData` object 52
URL
`myEDIServices.v1:processEDIDoc` service 33
`wm.tn:receive` service 188
user status
 document with duplicate control number 242
 document with out-of-sequence control number 242
User Status system attribute, not used 208, 228
 user-defined format, and flat file schema 21

V

Validate inbound envelope control numbers field 168, 393, 408, 411
Validate Structure pre-processing action 208, 229
 setting when invalid control number 249
validate variable, for `wm.b2b.edi:envelopeProcess` service 47
validation
 codes for FAs 69
 control numbers 174, 238
 defining action to take when invalid number 170
 turning on or off 168
EDI envelopes, mechanism 47
interchange service 47
 internal-format document before converting to EDI document 91
Value Added Networks (VANs), *See* `VANS`
Values variable 67
`VAN.VANConnectivity:getFromVAN` service 336, 341
 input variables 336
`VAN.VANConnectivity:putToVAN (VANFTP)` service 338, 341
VANs
 connecting to another VAN by customizing built-in services 341
 connectivity 336
 conventions of different VANs 340
 defining processing rules 340
 retrieving inbound EDI documents from 336
 scheduled delivery queue 337
 Trading Networks public queue 337
 viewing delivery progress of 379
variables
`bizdoc/Attributes/EDI Processing Mode` 214, 218
`ControlNumberManagement`
`duplicateControlNumberAction` EDITPA variables 125, 127, 389
`outOfSequenceControlNumberAction` EDITPA variables 389
`useReverseRouting` EDITPA variables 127
`validateInboundEnvelopeControlNumbers` EDITPA variables 124
`validateInboundGroupControlNumbers` EDITPA variables 125
`ctlFromTable` 289
`delimiters/field` EDITPA variable 138

delimiters/record EDITPA variable 138
 delimiters/release EDITPA variable 139
 delimiters/subfield EDITPA variable 139
 docID 212, 233
 docTypeID 212, 233
 docTypeName 212, 233
 edidata, *See* edidata pipeline variable
 EDITPA, *See also* EDITPA variables
 envelopeDocuments 211, 232
 envelopIdentifier/receiver/ID EDITPA variable 141
 envelopIdentifier/receiver/qualifier EDITPA variable 142
 envelopIdentifier/sender/ID EDITPA variable 140
 envelopIdentifier/sender/qualifier EDITPA variable 141
 FAGeneration
 addGroup EDITPA variable 129
 autoGenerateFA EDITPA variable 129
 FALevel EDITPA variable 129
 generateControlNumber EDITPA variable 130
 processDocument EDITPA variable 135
 rejectionRules/childTransactionRejectedStatus EDITPA
 variable 133
 rejectionRules/logicalErrorStatus EDITPA variable 132
 rejectionRules/syntaxErrorStatus EDITPA variable 131
 FAReconciliation EDITPA variable 136
 groupDocuments 212, 233
 GSRouting/routingMode EDITPA variables 120
 IHeaderInfo/GS/GS07 EDITPA variable 143
 IHeaderInfo/ISA/ISA01 EDITPA variable 142
 IHeaderInfo/ISA/ISA02 EDITPA variable 142
 IHeaderInfo/ISA/ISA03 EDITPA variable 143
 IHeaderInfo/ISA/ISA04 EDITPA variable 143
 IHeaderInfo/ISA/ISA11 EDITPA variable 143
 IHeaderInfo/UNB/UNB01 EDITPA variable 144
 IHeaderInfo/UNB/UNB06 EDITPA variable 144
 IHeaderInfo/UNB/UNB08 EDITPA variable 144
 input, *See* input variables
 iterator, *See* iterator variable
 persistMultipleDocEnvelope EDITPA variables 123, 158
 processingMode EDITPA variable 122
 publishBatchFailEvent EDITPA variable 140, 159
 routingMode variable 120
 splitOption 119, 388
 substitution, and concatenating strings 78
 TRADACOMS/ControlNumberManagement
 duplicateControlNumberAction EDITPA variables 155
 outOfSequenceControlNumberAction EDITPA variables
 156
 validateInboundBatchControlNumbers EDITPA variables
 154
 validateInboundFileControlNumbers EDITPA variables 155
 validateInboundTransmissionControlNumbers EDITPA
 variables 154
 TRADACOMS/enforceApplicationReference 157
 TRADACOMS/Outbound/applicationReference 296
 TRADACOMS/Outbound/applicationReferenceFromFile 296
 TRADACOMS/Outbound/createReconciliationMessage 295
 TRADACOMS/Outbound/recipientTransmissionReference
 295
 TRADACOMS/splitOption 153, 295
 TRADACOMS/storageOption 154
 transactionDocuments 212, 233
 UNAmode EDITPA variable 138
 verification, VAN.VANConnectivity:getFromVAN service 337
 Verify Digital Signature pre-processing action 209, 229
 setting when invalid control number 249
 Version field
 control number information 179, 183
 FA reconciliation report 370
 versions, adding a new version of an EDI standard 107
 viewing
 EDI documents on Transaction Analysis screen 376
 FA reconciliation report 370
 group-level sender/receiver pair associations 407
 server log 374
 Trading Networks activity log 374

W

webMethods Modeler, designing process models 346
 webMethods Monitor, business processes 357
 window, control number
 default 173
 defining 177
 definition 173
 example of using to determine duplicate 176
 example of using to determine out-of-sequence 176
 setting 177

wm.b2b.edi.migration:getTemplate service 24
wm.b2b.edi.tradacoms.compose:addToTradacomsTransmission service 290
 adding Transmission and Batch envelopes 290
wm.b2b.edi.tradacoms.compose:endTradacomsTransmission service 290
 adding Transmission and Batch envelopes 290
 processing File documents 236
wm.b2b.edi.tradacoms.compose:startTradacomsBatch service 290
 adding Transmission and Batch envelopes 290
wm.b2b.edi.tradacoms.compose:startTradacomsTransmission service 290
 adding Transmission and Batch envelopes 290
 processing File documents 236
wm.b2b.edi.tradacoms.doc:getContentPart service
 processing File documents 235
wm.b2b.edi.tradacoms.doc:getDocumentPartInfo service
 processing File documents 235
wm.b2b.edi.tradacoms.doc:getFFSchemaNames service
 processing File documents 235
wm.b2b.edi.tradacoms.doc:isFileEnvelope service
 processing File documents 235
wm.b2b.edi.tradacoms:convertToString service 25
 forming EDI document using standard processing 288
wm.b2b.edi.tradacoms:convertToValues service 25
 forming EDI document using standard processing 287
large document handling 344
processing
 File documents 235
wm.b2b.edi.util:addGroupEnvelope service 90
 obtaining control numbers for outbound processing 290
wm.b2b.edi.util:addGroupEnvelopeEDIFACT service
 obtaining control numbers for outbound processing 290
wm.b2b.edi.util:addICEnvelope service 90, 298
 forming EDI document using non-standard processing 415
 forming EDI document using standard processing 288
 obtaining control numbers for outbound processing 290
wm.b2b.edi.util:addICEnvelopeEDIFACT service 90, 91
 obtaining control numbers for outbound processing 290

wm.b2b.edi.util:EDIconcat service, logic to processing an inbound EDI document iteratively 51
wm.b2b.edi.util:generateFA service 56, 92
 determining FA status 59
 generating functional acknowledgments (FA) 56
 input variables for 69
 invoking
 after wm.b2b.edi:envelopeProcess service 67
 before wm.b2b.edi:envelopeProcess service 66
 large document handling 344
 output 69
wm.b2b.edi.util:getEDIstring service
 processing Group documents 217
 sample code for processing entire inbound EDI document at one time 45
wm.b2b.edi.util:invoke service 83
wm.b2b.edi:convertToString service 24, 25
 forming EDI document using non-standard processing 413
 forming EDI document using standard processing 288
processing
 Group documents 218
 Transaction documents 214
wm.b2b.edi:convertToValues service 25
converting
 EDI documents to IData objects iteratively 98
creating service to form EDI document 89
customizing EDI flat file schema for iterative processing 48
EDI Module configuration file 97
forming EDI document using non-standard processing 415
forming EDI document using standard processing 287
iterator input variable
large document handling 94, 97, 344
logic to processing an inbound EDI document iteratively 50
migration 24
processing
 document segment by segment 47
 Group documents 217
 Transaction documents 214
sample code for processing entire inbound EDI document at one time 46
validating internal-format document 91
wm.b2b.edi:createIDOCtemplate service 23

- wm.b2b.edi:envelopeProcess service 47, 92
 complianceCheck variable 47
 EDI Module configuration file 97
 invoking after wm.b2b.edi.util:generateFA service 66, 67
 large document handling 97, 344
 processing Group documents 216
 sample code for processing entire inbound EDI document at one time 43
 validate variable 47
- wm.b2b.edi:pad service 85
- wm.b2b.edi:SEFParse service
 assigning format services 29
 creating flat file schemas 21
- wm.b2b.editn.batch:batchProcess service 310
 assigning scheduled delivery queue to 330
 create BizDocEnvelope for final batch EDI document 321
 EDITPA variables 327, 328
 final batch EDI document 332
 forming batch document 326
 input variables 326
 oneBatchQueue input variable 310
 publishing IS document to notify of batching failure 311, 315
 replacing delimiters to match those in output batch EDI document 313
 single pass, version 6.0.1 batching 420
 steps to extract and sort transactions to be batched 311, 315
 two passes through Trading Networks using version 6.0.1 batching 424
- wm.b2b.editn.crossRef:getEnvInfo service, forming EDI document using non-standard processing 414
- wm.b2b.editn.FAReport:generateFA service 363
- wm.b2b.editn.rec:batchFailDocument IS document type 320
- wm.b2b.editn.TPA:EDITPA IS document type 117, 118, 151, 152
- wm.b2b.editn.TPA:getEDITPAData service 287
- wm.b2b.editn.TPA:initService 118, 152
- wm.b2b.editn.util.reprocess:listInSequence service, reprocessing out-of-sequence documents 252
- wm.b2b.editn.util.reprocess:listUnprocessedDocuments service, locating documents with duplicate control numbers 250
- wm.b2b.editn.util.reprocess:nextInSequenceDoc service, reprocessing out-of-sequence documents 256
- wm.b2b.editn.util.reprocess:reprocessDocument service
 reprocessing duplicate documents 251
 reprocessing out-of-sequence documents 255
- wm.b2b.editn.util:getContentPartDataAsInputStream service
 processing File documents 235
- wm.b2b.editn:ediPartnerIDToTNPartnerID service, retrieving Trading Networks internal ID 305
- wm.b2b.editn:getTspace service
 processing Group documents 216
 processing Transaction documents 213
- wm.b2b.editn:trackEDIdocs service 136
 delivering EDI document directly from the service that forms it 292
 routing EDI document through Trading Networks 301, 302
- wm.b2b.util.documentList:addDocToDocumentList service
 processing File documents 235
- wm.pub.flow:debugLog service, processing Transaction documents 215, 218
- wm.tn.doc.xml:routeXml service
 preparing pipeline to invoke 298, 301
 sending multiple EDI documents 191
 submitting document to Trading Networks for document recognition 298
- wm.tn.doc:addContentPart service, adding outbound EDI document to BizDocEnvelope 308
- wm.tn.doc:createNewEnvelope service, creating BizDocEnvelope for outbound EDI document 307
- wm.tn.doc:view service, obtain document attribute when reprocessing documents 253
- wm.tndoctype:view service, determining Trading Networks internal ID 306
- wm.tn.rec:BizDocEnvelope IS document type 211, 232
- wm.tn.route:routeBizdoc service, routing document through Trading Networks 302
- wm.tn:receive service
 and pub.client:ftp service 190
 and pub.client:http service 188
 client invokes 188
 example of 190
 File Polling 191
 path of for FTP 190
 port number 5555 for HTTP 188
 POST EDI document to 188
 sending
 documents directly to Trading Networks 188
 EDI documents to Trading Networks via FTP 189
 EDI documents to Trading Networks via HTTP 188
 multiple EDI documents 191
 URL of for HTTP 188

WmEDI package
 EDI Module 40
 EDICOR server log key 374
 large document handling 344
 large document processing 344
 wm.b2b.edi.tradacoms.compose 290
WmEDI\config directory 199
WmEDIforTN package
 EDIFFSchema folder 104, 105
 EDIFFSchema.Standard.Vversion:Ttransaction 355
 EDIFTN server log key 374
 EDITPA 116, 150
WmEDIforTN\Pub\FAReports directory 370
WmEDIsamples package
 downloading 45
 outbound flow services 92
 sampleServices folder examples 53
 sampleServices:UNEDIFACTToValues 45
 Tutorial.EDItoXML folder examples 73
 Tutorial.XMLtoEDI service 85
 Tutorial.XMLtoEDI:processXMLSource service 85, 92
 Tutorial.XMLtoEDI:XMLPOtoEDIMap service 85
WmFlatFile package configuration 27

X

XML documents
 converting into outbound EDI (ANSI X12), example of 92
 EDI-to-XML mapping examples 73
 mapping to EDI format 84
 XML-to-EDI flow service example 92