

webMethods Module for EDI Installation and User's Guide

Version 9.12

September 2017

This document applies to webMethods Module for EDI 9.12 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2000-2023 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <https://softwareag.com/licenses/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

Document ID: ESTD-EDI-IUG-912-20231207

Table of Contents

About this Guide	7
Document Conventions.....	8
Online Information and Support.....	8
Data Protection.....	9
1 Installing, Upgrading, and Uninstalling webMethods Module for EDI	11
Overview.....	12
Requirements.....	12
Installing Module for EDI.....	12
Upgrading the Module for EDI from 9.0 to 9.12.....	17
Uninstalling Module for EDI.....	18
2 Configuring webMethods Module for EDI	21
Overview.....	22
Assigning User-Specific Permissions in My webMethods.....	22
Defining Module for EDI Properties.....	24
Setting Up Items Required for Structural Validation and Conversion.....	29
Defining TN EDI Document Types.....	34
Creating a TN EDI Document Type Based on a Transaction.....	40
Flat File Schema Namespace Conventions.....	41
Managing TN EDI Document Types.....	42
Managing Document Attributes.....	45
Overwriting Document Attributes Extracted from Inbound Documents.....	46
Adding New Versions of EDI Standards.....	49
3 Defining Trading Partner Information	53
Overview.....	54
Partner Information You Need to Define.....	54
Defining Trading Partner Profiles.....	55
Adding External ID Types for EDI ID Qualifiers.....	55
Adding and Overriding EDI ID Qualifiers.....	56
Partner Profile ID Types Mapping for EDI Documents.....	57
Defining EDI Trading Partner Agreements.....	59
4 Defining Trading Partner Information When Using TRADACOMS	97
Overview.....	98
Defining Trading Partner Profiles When Using TRADACOMS.....	99
Defining EDI Trading Partner Agreements When Using TRADACOMS.....	101
5 Defining Control Number Information for Trading Partners	113
Overview.....	114
Managing Control Number Information.....	114

Turning Inbound Control Number Validation On and Off.....	121
Defining Control Number Information for Processing Documents.....	123
Defining Actions Module for EDI Can Take for Invalid Control Numbers.....	124
Searching for Existing Control Number Settings.....	127
6 Creating Clients that Send EDI Documents to Integration Server.....	131
Overview.....	132
Defining the Content Type to Use.....	132
Before Creating the Service to Form an EDI Document.....	132
Creating the Service the Client Invokes to Handle EDI Documents.....	133
Sending EDI Documents to Integration Server via HTTP.....	136
Sending EDI Documents to Integration Server via FTP.....	138
Sending EDI Documents to Integration Server via File Polling.....	139
7 Creating Clients that Send EDI Documents to Trading Networks.....	141
Overview.....	142
Content Type to Use.....	142
Service the Client Invokes.....	143
Sending EDI Documents to Trading Networks via HTTP/S.....	143
Sending EDI Documents to Trading Networks via FTP.....	144
Sending EDI Documents to Trading Networks via File Polling.....	145
Sending EDI Documents to Trading Networks via My webMethods.....	146
Using a Service to Send Multiple EDI Documents to Trading Networks.....	147
8 Handling Large Documents When Using Trading Networks.....	149
Overview.....	150
Configuring Module for EDI for Large Document Handling.....	150
Converting Large Documents to IData Objects Iteratively.....	152
Troubleshooting Memory Constraint Problems.....	152
9 Generating Acknowledgments.....	153
Overview.....	154
Generating Functional Acknowledgments.....	154
Before You Can Generate a Functional Acknowledgment.....	156
Determining How Module for EDI Reports FA Status.....	156
When to Invoke the generateFA Service.....	163
Output from the generateFA Service.....	166
Tracking Late Functional Acknowledgments.....	167
Generating Interchange Acknowledgments.....	169
10 Before Creating the Service to Process EDI Documents.....	171
Overview.....	172
Inputs to Your Service.....	172
Logic to Include in the Service to Process EDI Documents.....	172
Samples.....	182
11 Preparing to Process Inbound Non-TRADACOMS Documents Using Trading Networks.....	185

Overview.....	186
Preparing to Receive EDI Documents.....	187
Specifying EDITPA Variables that Affect Inbound Processing.....	188
Trading Networks Attributes and EDI Documents.....	189
Defining Processing Rules to Process Inbound EDI Documents.....	198
Coding Services to Process Transaction and Group Documents.....	202
12 Processing Inbound TRADACOMS Documents Using Trading Networks.....	211
Overview.....	212
Before Setting Up Processing of Inbound EDI Documents When Using TRADACOMS....	213
Specifying EDITPA Variables that Affect Inbound Processing When Using TRADACOMS.	213
Trading Networks Attributes and EDI Documents When Using TRADACOMS.....	214
Defining Processing Rules to Process Inbound EDI Documents When Using TRADACOMS.....	218
Coding Services to Process File and Batch Documents When Using TRADACOMS.....	221
13 Optional Inbound Processing When Using Trading Networks.....	229
Overview.....	230
Validating Inbound Control Numbers.....	230
Defining Processing Rules to Handle Documents with Invalid Control Numbers.....	236
Reprocessing EDI Documents with Invalid Control Numbers.....	238
Automatically Generating Functional Acknowledgments.....	246
Automatically Generating Interchange Acknowledgments.....	253
Performing Optional Processing of Inbound EDI Documents.....	254
14 Forming EDI Documents to Send Outbound When Using Trading Networks.....	257
Overview.....	258
Preparing Trading Networks to Process Internal-Format Documents.....	258
Creating the Service to Form the EDI Document.....	258
Delivering the Outbound EDI Document.....	264
15 Batching EDI Documents.....	275
Overview.....	276
Indicating How Many Batch EDI Documents to Create.....	276
How Documents Are Combined to Create the Batch EDI Document.....	277
Extracting Transactions and Sorting the Batched Documents.....	277
Recombining the Transactions into the Batch EDI Document.....	284
Before You Can Batch EDI Documents.....	289
Controlling How the batchProcess Service Forms the Batch Document.....	290
Preparing to Batch EDI Documents.....	291
Delivering the Batch EDI Document.....	293
Viewing Batched EDI Documents.....	294
16 Retrieving and Delivering EDI Documents from and to VANs.....	297
Overview.....	298
Preparing to Retrieve Documents from a VAN.....	298
Preparing to Deliver Documents to VANs.....	299

Handling Conventions Required by Specific VANs.....	301
Customizing Built-In Services to Connect to Another VAN.....	301
17 Reconciling Functional Acknowledgments.....	303
Overview.....	304
Turning FA Reconciliation On or Off.....	305
Information Module for EDI Records to Reconcile FAs.....	305
18 Viewing Log Information for EDI Documents.....	311
Overview.....	312
Viewing the Server Log.....	312
Viewing the Trading Networks Activity Log.....	312
19 Viewing Information about EDI Documents.....	315
Overview.....	316
EDI Documents You Can View.....	316
Viewing EDI Documents.....	317
Viewing Information about EDI Documents in Queues.....	319
Viewing Related Documents with Relationship Labels.....	319
20 Including Documents in a Business Process.....	323
Overview.....	324
Designing the Process Model.....	324
Assigning Conversation IDs to EDI Documents.....	328
Sample Business Process.....	330
Monitoring Your Business Processes.....	330
A Non-Standard Processing.....	331
Overview.....	332
Defining Partner Information When Using Non-Standard Processing.....	332
Defining Control Number Information for Partners When Using Non-Standard Processing.....	350
Processing Inbound EDI Documents using Trading Networks and Non-Standard Processing.....	352
Forming EDI Documents to Send Outbound When Using Trading Networks and Non-Standard Processing.....	353
B Using Module for EDI Decoupled from Trading Networks.....	355
Overview.....	356
Configuring Module for EDI Decoupled from Trading Networks.....	356
Preparing Module for EDI Decoupled from Trading Networks.....	358
Creating Flat File Schemas for EDI Documents.....	358
Forming EDI Documents and Sending Them Outbound.....	361
Example.....	363

About this Guide

■ Document Conventions	8
■ Online Information and Support	8
■ Data Protection	9

This guide describes how to use webMethods Module for EDI to process EDI documents, including:

- How to install and configure Module for EDI and install webMethods Trading Networks document types
- Define processing rules
- Create services to process inbound and outbound EDI documents

For an overview of Module for EDI features, see *webMethods Module for EDI Concepts Guide*.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.softwareag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://techcommunity.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/u/softwareag> and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Installing, Upgrading, and Uninstalling webMethods Module for EDI

■ Overview	12
■ Requirements	12
■ Installing Module for EDI	12
■ Upgrading the Module for EDI from 9.0 to 9.12	17
■ Uninstalling Module for EDI	18

Overview

This chapter explains how to install, upgrade, and uninstall webMethods Module for EDI. The instructions use the Software AG Installer and the Software AG Uninstaller wizards. For complete information about the wizards or other installation methods, or to install other webMethods products, see the *Installing webMethods Products On Premises* for your release.

Requirements

For a list of the operating systems and webMethods products that Module 9.12 for EDI supports, see *webMethods eStandards Modules System Requirements*, available in the webMethods area of the Software AG Documentation website.

Installing Module for EDI

Note: Module for EDI stores data in Trading Networks tables. If you are installing the module in a clustered environment, you must install Module for EDI and Trading Networks on each Integration Server in the cluster, and each installation must be identical. For information on running Trading Networks in a clustered environment, see the *webMethods Trading Networks Administrator's Guide* for your release.

> To install Module for EDI

1. If you are installing Module for EDI on an already installed Integration Server, shut down Integration Server.
2. Download Software AG Installer from the Empower Product Support website at <https://empower.softwareag.com>.
3. Start the Software AG Installer wizard.
4. Choose the supported webMethods version that corresponds to Integration Server that you are installing. Click **Next**.
5. Specify the installation directory as follows:
 - If you are installing Module for EDI on an existing Integration Server, specify the Software AG installation directory that contains the host Integration Server.
 - If you are installing both the host Integration Server and Module for EDI for the first time, specify the installation directory to use. (The default is SoftwareAG.)
6. In the product selection list, navigate to **eStandards > webMethods Module 9.12 for EDI**. Install any other required products as indicated in *webMethods eStandards Modules System Requirements*.

Software AG Installer installs the following packages in the *Integration Server_directory\instances\instance_name\packages* directory:

Note:

If you select the top level check box for a schema package, then all the versions for that package are selected automatically. If you don't want to install all the versions for a schema package, then deselect the top level check box, and select individual versions for a package or individual packages instead.

Note:

Starting with webMethods Module for EDI 9.12, XML schemas are introduced for use with XML based document types. XML schemas provide the following advantages:

- View the code list details of a segment for a document type.
- View the alternate name and description of a segment for an XML schema.

Note:

With webMethods Module for EDI 9.12, SEF schema also displays the alternate name and description of a segment.

Prerequisite: Install fix EDI_9.12_X12_SEF_Fix1 and refer to the corresponding Fix Readme for more details.

Installer Options		Packages	Versions for SEF	Versions for XML
Program Files		WmEDI WmEDIforTN		
Schema	EANCOM	WmEDI_EANCOM	01B, 1, 2, 3, 93A, and 96A	01B, 1, 2, 3, 93A, and 96A
	ODETTE	WmEDI_ODETTE	3 and 94	3 and 94
	TRADACOMS	WmEDI_TRADACOMS	2, 3, 4, 5, 6, 8, and 9	2, 3, 4, 5, 6, 8, and 9
	UCS	WmEDI_UCS	4010, 4020, 4030, and 5010	4010, 4020, 4030, and 5010
	UNEDIFACT > Release 1990	WmEDI_UNEDIFACT_1990	901 and 902	901 and 902
	UNEDIFACT > Release 1991	WmEDI_UNEDIFACT_1991	911 and 912	911 and 912
	UNEDIFACT > Release 1992	WmEDI_UNEDIFACT_1992	921	921

Installer Options	Packages	Versions for SEF	Versions for XML
UNEDIFACT > WmEDI_UNEDIFACT_1993 Release 1993		93A, 2932, and S93A	93A
UNEDIFACT > WmEDI_UNEDIFACT_1994 Release 1994		94A and 94B	94A and 94B
UNEDIFACT > WmEDI_UNEDIFACT_1995 Release 1995		95A and 95B	95A and 95B
UNEDIFACT > WmEDI_UNEDIFACT_1996 Release 1996		96A and 96B	96A and 96B
UNEDIFACT > WmEDI_UNEDIFACT_1997 Release 1997		97A and 97B	97A and 97B
UNEDIFACT > WmEDI_UNEDIFACT_1998 Release 1998		98A and 98B	98A and 98B
UNEDIFACT > WmEDI_UNEDIFACT_1999 Release 1999		99A and 99B	99A and 99B
UNEDIFACT > WmEDI_UNEDIFACT_2000 Release 2000		00A and 00B	00A and 00B
UNEDIFACT > WmEDI_UNEDIFACT_2001 Release 2001		01A, 01B, and 01C	01A, 01B, and 01C
UNEDIFACT > WmEDI_UNEDIFACT_2002 Release 2002		02A and 02B	02A and 02B
UNEDIFACT > WmEDI_UNEDIFACT_2003 Release 2003		03A and 03B	03A and 03B
UNEDIFACT > WmEDI_UNEDIFACT_2004 Release 2004		04A and 04B	04A and 04B
UNEDIFACT > WmEDI_UNEDIFACT_2005 Release 2005		05A and 05B	05A and 05B
UNEDIFACT > WmEDI_UNEDIFACT_2006 Release 2006		06A and 06B	06A and 06B
UNEDIFACT > WmEDI_UNEDIFACT_2007 Release 2007		07A and 07B	07A and 07B
UNEDIFACT > WmEDI_UNEDIFACT_2008 Release 2008		08A and 08B	08A and 08B
UNEDIFACT > WmEDI_UNEDIFACT_2009 Release 2009		09A and 09B	09A and 09B

Installer Options	Packages	Versions for SEF	Versions for XML
UNEDIFACT > Release 2010	WmEDI_UNEDIFACT_2010	10A and 10B	10A and 10B
UNEDIFACT > Release 2011	WmEDI_UNEDIFACT_2011	11A and 11B	11A and 11B
UNEDIFACT > Release 2012	WmEDI_UNEDIFACT_2012		12A
UNEDIFACT > Release 2013	WmEDI_UNEDIFACT_2013	13A and 13B	13A and 13B
UNEDIFACT > Release 2014	WmEDI_UNEDIFACT_2014	14A and 14B	14A and 14B
UNEDIFACT > Release 2016	WmEDI_UNEDIFACT_2016	16A and 16B	16A and 16B
UNEDIFACT > Release 2017	WmEDI_UNEDIFACT_2017	17A and 17B	
UNEDIFACT > Release 2018	WmEDI_UNEDIFACT_2018	18A and 18B	
UNEDIFACT > Release 2019	WmEDI_UNEDIFACT_2019	19A and 19B	
UNEDIFACT > Release 2020	WmEDI_UNEDIFACT_2020	20A and 20B	
UNEDIFACT > Release 2021	WmEDI_UNEDIFACT_2021	21A and 21B	
UNEDIFACT > Release 2022	WmEDI_UNEDIFACT_2022	22A and 22B	
UNEDIFACT > Miscellaneous	WmEDI_UNEDIFACT_Misc	2, 3, 40, and 41	2, 2932, 3, 40, and 41
VDA	WmEDI_VDA	4905_4, 4906_2, 4907_2, 4908_3, 4911_1, 4913_4, 4913_5, 4915_2, 4916_1, 4918_1, 4919_1, 4920_1, 4921_1, 4927_3, 4938_1, 4984_1, 4986_12 and 4987_1	4905_4, 4906_2, 4907_2, 4908_3, 4911_1, 4913_4, 4913_5, 4915_2, 4916_1, 4918_1, 4919_1, 4920_1, 4921_1, 4927_3, 4913_5, 4915_2, 4916_1, 4918_1,

Installer Options	Packages	Versions for SEF	Versions for XML
			4919_1, 4920_1, 4921_1, 4927_3, 4938_1, 4984_1, 4986_12, and 4987_1
VICS	WmEDI_VICS	3010, 4010, 4020, 4030, 4050, and 5010	3010, 4010, 4020, 4030, 4050, and 5010
X12 > Version 20xx	WmEDI_X12_20xx	2000, 2001, 2001FORD, 2002, 2002FORD, 2003, 2003GM, 2040, and 2040CHRY	2000, 2001, 2001FORD, 2002, 2002FORD, 2003, 2003GM, 2040, and 2040CHRY
X12 > Version 30xx	WmEDI_X12_30xx	3010, 3020, 3030, 3040, 3041, 3050, 3051, 3060, and 3070	3010, 3020, 3030, 3040, 3041, 3050, 3051, 3060, and 3070
X12 > Version 40xx	WmEDI_X12_40xx	4010, 4010RIFMAT, 4020, 4030, 4040, 4050, and 4060	4010, 4010RIFMAT, 4020, 4030, 4040, 4050, and 4060
X12 > Version 50xx	WmEDI_X12_50xx	5010, 5020, 5030, 5040, and 5050	5010, 5020, 5030, 5040, and 5050
X12 > Version 60xx	WmEDI_X12_60xx	6010, 6020, 6040, and 6050	6010, 6020, 6040, and 6050
X12 > Version 70xx	WmEDI_X12_70xx	7010, 7030, 7050, and 7060	7010, and 7030
X12 > Version 80xx	WmEDI_X12_80xx	8010	

Note: Module for EDI uses the Trading Networks JDBC pool associated with Integration Server. The module database scripts are installed as part of the Trading Networks script installation using the Database Component Configurator. For more information about Trading Networks script installation, see the *Installing webMethods Products On Premises* for your release.

7. After Installer completes the installation, close the wizard.
8. Start the host Integration Server.

Upgrading the Module for EDI from 9.0 to 9.12

Upgrade the module from 9.0 to 9.12 by installing Module 9.12 for EDI in a separate installation directory, editing the properties in the config files, and migrating the data from webMethods Trading Networks.

Important:

Module 9.12 for EDI requires the supported Trading Networks version. You must upgrade all your webMethods products at the same time. For more information about upgrading your webMethods products, see the *Upgrading Software AG Products* for your release.

➤ To upgrade the module from 9.0 to 9.12

1. Shut down all webMethods products and all other applications that are running on the machine on which you want to install Module 9.12 for EDI. In addition, if any business processes are running, wait for them to complete normally or use Monitor to stop them.

Important:

If all products, applications, and business processes are not shut down, Installer cannot replace the key files that are locked by the operating system.

2. Back up the following 9.0 installation directories:
 - ... \packages\WmEDI
 - ... \packages\WmEDIforTN
3. Install Module 9.12 for EDI as described in “[Installing Module for EDI](#)” on page 12. Specify the Module 9.12 for EDI installation directory for the installation directory.
4. Copy the ... \packages\WmEDI\config\properties.cnf file from your backup installation to your 9.12 installation.
5. Migrate the data and TPAs from Trading Networks. For instructions about migrating data from Trading Networks, see *Upgrading Software AG Products*.

6. If you modified a schema in the WmEDIforTN\ms\EDIFFSchema folder for the WmEDIforTN package in the ... \packages\WmEDIforTN installation directory, then copy the updated schema to the same installation directory. If you used a different package for the flat file schemas, then copy the package to the Module for EDI 9.12 installation using the Integration Server migration utility. For more information, see *Upgrading Software AG Products*.

Note:

If you have not copied the schemas to the installation directory, then perform the following actions:

- Create the flat file schemas that are used in Module for EDI 9.0 in Module for EDI 9.12. For information about creating flat file schemas, see [“Flat File Schema Namespace Conventions” on page 41](#).
- To use the migrated document types; open the document type in My webMethods Server and click **Save and Close**. This registers the document type with Integration Server and is ready for use.

Uninstalling Module for EDI

Before You Begin

Uninstalling Module 9.12 for EDI removes all components in the Module for EDI packages. To keep certain records or services from the existing Module for EDI packages on Integration Server, export them to a new package.

➤ **To export a module package or service**

1. In Designer, select the package or the service you want to export.
2. Select **File > Export**.

If you select a package, the entire package is exported. If you select a service, only the selected service is exported.

Uninstalling Module 9.12 for EDI

➤ **To uninstall Module for EDI**

1. Shut down the Integration Server that hosts Module for EDI.
2. Start Software AG Uninstaller, selecting the Software AG installation directory that contains the host Integration Server. In the product selection list, select **eStandards > webMethods Module 9.12 for EDI** and any other products and items you want to uninstall.
3. After Software AG Uninstaller completes, restart the host Integration Server.

4. Software AG Uninstaller removes all Module 9.12 for EDI-related files that were installed into the *Integration Server_directory*\instances*\$instance_name*\packages directory. However, Uninstaller does not delete files you created after you installed the module (for example, user-created files or configuration files), nor does it delete the module directory structure. You can navigate to the *Integration Server_directory*\instances*\$instance_name*\packages directory and delete the WmEDI-related directories.

2 Configuring webMethods Module for EDI

■ Overview	22
■ Assigning User-Specific Permissions in My webMethods	22
■ Defining Module for EDI Properties	24
■ Setting Up Items Required for Structural Validation and Conversion	29
■ Defining TN EDI Document Types	34
■ Creating a TN EDI Document Type Based on a Transaction	40
■ Flat File Schema Namespace Conventions	41
■ Managing TN EDI Document Types	42
■ Managing Document Attributes	45
■ Overwriting Document Attributes Extracted from Inbound Documents	46
■ Adding New Versions of EDI Standards	49

Overview

Before you can process EDI documents using webMethods Module for EDI (Module for EDI), you must first configure the module. The following table summarizes the configuration tasks and where you can find more information about those tasks.

Task	Where to Find More Information
Assign user-specific permissions	“Assigning User-Specific Permissions in My webMethods” on page 22
Define the Module for EDI properties	“Defining Module for EDI Properties” on page 24
Set up the items required for parsing and validating the structure of EDI documents and converting documents to a different format	“Setting Up Items Required for Structural Validation and Conversion” on page 29 For more information about schemas and formatting related services, including how to specify format services for fields in flat file schemas and dictionaries, see <i>Flat File Schema Developer’s Guide</i> .
Define the TN EDI document types that you want to process	“Defining TN EDI Document Types” on page 34 For more information about TN document types, profiles, and TPAs and how Trading Networks uses them, see the <i>webMethods Trading Networks Administrator’s Guide</i> for your release.
Configure Module for EDI to support a new version of any EDI standard	“Adding New Versions of EDI Standards” on page 49
Define trading partner information	“Defining Trading Partner Information” on page 53
Define settings for inbound control number validation	“Defining Control Number Information for Trading Partners” on page 113
Optionally, prepare to use Module for EDIINT to transport documents through EDIINT	<i>webMethods Module for EDIINT Installation and User’s Guide</i>

Assigning User-Specific Permissions in My webMethods

My webMethods is a web-based administration and monitoring user interface for managing your webMethods components. Use My webMethods to perform functions such as managing your trading partners, configuring how documents are exchanged through your network, and performing real-time monitoring and analysis.

The functionality available in the My webMethods user interface is based on the permissions assigned to your My webMethods roles. For more information about using My webMethods, see the *Administering My webMethods Server* for your release.

The My webMethods administrator and the system administrator can manage permissions for users, groups, and roles in whatever combination is needed.

Functional privileges define the actions a role can perform within My webMethods. Data-level security defines the data on which a role can act.

The following table describes the Trading Networks general functional permissions specific to Module for EDI that you can assign to grant access to Trading Networks functionality in My webMethods. The table lists the data governed by each general functional permission and the actions a role can perform when assigned the permission.

Functional permission	Actions a role can perform when assigned this functional permission
Submit Documents to TN	Use the Administration > Integration > B2B Settings > Submit Documents page to submit EDI documents to Trading Networks.
Manage EDI Control Number Sequences	Use the Administration > Integration > B2B Settings > Control Numbers page to manage (add, edit, delete) control numbers.

To set data permissions, you must first define data sets. After defining a data set, you then specify the roles that can act on the data in the data set. For each role, define permissions that indicate the actions the role can perform on the data.

For more information about data permissions, general functional privileges, managing permissions, and data and actions governed by data permissions, see the *webMethods Trading Networks Administrator's Guide* for your release.

The following table describes the module-specific Trading Networks data permissions that you can assign to a role.

Data permission	Actions a role can perform when assigned this data permission
Edit EDI Control Number Sequences	<p>Edit the control number sequence associated with partner profiles in the data set</p> <p>Note: You need Manage EDI Control Number Sequences functional permission to use the Control Number tab in the Administration > Integration > B2B > Partner Profiles > Partner Profile page.</p>
Edit EDI FA Status Attributes	Edit EDI FA Status attributes of the transactions in the data set

Defining Module for EDI Properties

Module for EDI properties are defined in the `properties.cnf` configuration file. This file contains properties that specify how to handle large document thresholds, encoding, FA transaction errors, related documents, control number sequence, duplicate documents, and the extraction of conversation IDs.

➤ To define Module for EDI properties

1. Navigate to the `Integration Server_directory\instances\${instance_name}\packages\WmEDI\config` folder.
2. Take a backup of the `sample-properties.cnf` file and rename the file to `properties.cnf`.
3. Open the `properties.cnf` file in a text editor and set the properties as indicated in the following table. If a property is not listed in the `properties.cnf` file, you can add it manually.

Property	Description
EDIBigDocThreshold	<p>Determines the size at which Module for EDI considers a document to be large.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> ■ <i>0</i> or <i>-n</i>. Zero or any negative whole number specifies that Module for EDI should consider no documents as large. The module processes all documents in the traditional manner, reading the document content into memory during processing. The default is <i>-1</i>. ■ <i>n</i>. A positive whole number specifies the number of bytes over which Module for EDI treats a document as large. For example, if the value is 1000000, the module considers all documents greater than 1,000,000 bytes as large. <p>Note: You must specify a threshold number that is at least three times the size of a document. For example, if the document size is 1 MB, set <code>EDIBigDocThreshold</code> to at least 3 MB. Otherwise, Module for EDI still treats the document as large even though its size is smaller than the threshold number.</p>
EDICheckEncodingLength	<p>Specifies the length of each portion of data to be validated for encoding. The Module for EDI validates an EDI document by dividing it into smaller portions</p>

Property	Description
	of data where each portion of data is validated for encoding. You can add a suffix of K or M to the value. The default value is 20K.
EDlencoding	Specifies the character encoding for encoding EDI documents. The default is UTF-8. Module for EDI uses this property if it does not encounter an encoding variable in the pipeline.
EDIEncodingsToTry	<p>Specifies a comma-separated list of character encoding to use if the default encoding standard property (EDlencoding) fails.</p> <p>If both the EDlencoding property and the EDIEncodingsToTry property fail, then the value for the file.encoding java system property is used. If this value is UTF-8, then ISO8859-1 is used.</p>
EDIMaxFATransactionErrors	<p>Specifies the maximum number of errors to report per FA transaction. Valid values are:</p> <ul style="list-style-type: none"> ■ <i>n</i>. Specifies the maximum number of errors that can be reported for any one FA transaction. The default is 100. ■ -1. Specifies that any number of errors can be reported for any one FA transaction.
EDIPartnerOriginalIDAttributesEnabled	<p>Specifies whether to enables the Original Sender ID, Original Receiver ID, Original Sender ID Type and Original Receiver ID Type attributes extraction.</p> <ul style="list-style-type: none"> ■ <code>true</code>—Extracts the Original Sender ID, Original Receiver ID, Original Sender ID Type and Original Receiver ID Type attributes. ■ <code>false</code>—Does not extract, a null value appears for the Original Sender ID and the Original Receiver ID. The Original Sender ID Type and the Original Receiver ID Type attributes do not appear on the transaction attributes section.
EDIUseNewRelationshipLabel	<p>Indicates whether to display related documents identified by their control numbers or by their relationship labels on the Transaction Analysis page in My webMethods. Valid values are:</p> <ul style="list-style-type: none"> ■ <code>true</code>—Displays documents by relationship labels.

Property	Description
	<ul style="list-style-type: none"> ■ <code>false</code>—Displays documents by control numbers. This is the default.
EDIAllowCustomControlNumberSequences	<p>Specifies the control number sequence. Valid values are:</p> <ul style="list-style-type: none"> ■ <code>false</code>—Module for EDI looks up the next control number using the corporation names of the sender and receiver. This is the default. ■ <code>true</code>—Module for EDI looks up the next control number using the sender and receiver IDs and qualifiers. If no record is found using this criteria, the module attempts to identify the sequence based on corporation names. If no sequence is found, the module creates a new sequence based on partner corporation names. ■ <code>strictCompatibility</code>—Specifies whether to use either the external IDs of partners or their corporation names to define the control number sequence. Use this value for backward compatibility with Module for EDI 6.5.2 or if you want to adopt the sequence that a particular operation uses. (For example, with the batching feature, Module for EDI uses corporation names when generating outbound batches but uses sender/receiver IDs and qualifiers in services that add envelopes and groups.)
EDIResolveDuplicates	<p>Specifies how Module for EDI assigns FA status when sending (or receiving) a document multiple times before the receiver returns an FA.</p> <ul style="list-style-type: none"> ■ <code>false</code>—When sending 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 Module for EDI has not yet received an FA. When the receiver returns an FA for one of these documents, the status of each document is updated to Duplicate (“120”), indicating that multiple documents exist in the EDITRACKING table that match this FA. This is the default. ■ <code>true</code>—When sending a document the first time, its status is set to None (“100”) in the EDITRACKING table. This indicates that Module for EDI has not

Property	Description
	<p>yet received an FA. If the same document is sent again before the receiver returns an FA:</p> <ul style="list-style-type: none"> ■ The status of the first document changes to Duplicate (“120”), indicating that multiple documents exist in the EDITRACKING table that match this FA. ■ The status of the second document is set to None. This behavior occurs for each subsequent document sent before an FA is returned. <p>When the receiver returns an FA, the FA acknowledges only the document that was sent last. The status of this last document is set according to the FA status reported in the FA document. The status of all prior documents remains Duplicate.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: This behavior also applies to received documents.</p> </div>
disableCIDSupport	<p>Specifies whether to disable the extraction of conversation IDs from EDI documents for Envelope and Group document types.</p> <ul style="list-style-type: none"> ■ <code>true</code>—Disables the extraction of conversation IDs from EDI documents for Envelope and Group document types. ■ <code>false</code>—Enables the extraction of conversation IDs from EDI documents for Envelope and Group document types. This is the default. <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: For VDA documents, if you set this property to <code>false</code>, you must also add a package dependency from WmTN to WmPRT and then reload WmTN in order to extract the conversation ID correctly. For details, see the <i>webMethods Service Development Help</i> for your release.</p> </div>
ignoreFAStatusUpdateForOutboundBatch	<p>Controls whether Module for EDI checks and updates the EDI FA Status of incoming EDI documents during batching.</p> <ul style="list-style-type: none"> ■ <code>true</code>—The module does not check and update the EDI FA Status.

Property	Description
	<ul style="list-style-type: none"> ■ <code>false</code>—The module checks and updates the EDI Status. This is the default.
PersistUnchangedInboundEDIDocument	<p>Specifies whether to persist EDI documents in Trading Networks exactly as they are received, without removing empty records and spaces between records.</p> <ul style="list-style-type: none"> ■ <code>true</code>—EDI documents are persisted exactly as they are received. ■ <code>false</code>—EDI documents are persisted after removing empty records and spaces between records. This is the default.
EDIUseConvertToString61Behavior	<p>Specifies whether Module for EDI performs less stringent validation during implementation of the <code>wm.b2b.edi:convertToString</code> service. Software AG recommends setting this property to <code>false</code>.</p> <ul style="list-style-type: none"> ■ <code>true</code>—Module for EDI performs less stringent validation against the flat file schema during implementation of the <code>wm.b2b.edi:convertToString</code> service. Specifically, Module for EDI prevents sorting the edidata from the <code>wm.b2b.edi:convertToString</code> service when the service's input data does not follow the same order as the flat file schema and the <code>sortInput</code> parameter is set to <code>false</code>. ■ <code>false</code>—Module for EDI performs stringent validation against the flat file schema during implementation of the <code>wm.b2b.edi:convertToString</code> service when the <code>sortInput</code> parameter is set to <code>false</code>. This is the default.
EDIDisableCONTRLVersion	<p>Specifies whether both the release number and the version number of the UNEDIFACT message are used to determine the version. Valid values are:</p> <ul style="list-style-type: none"> ■ <code>true</code>—Module for EDI considers only the release number of the UNEDIFACT message to determine the version of the UNEDIFACT message. ■ <code>false</code>—Module for EDI considers both the release number and the version number when determining the version of the UNEDIFACT message. This is the default.
EDISchemaSource	<p>Specifies whether the flat file schemas are created using either the SEF or XML files. The default is XML.</p>

Property	Description
	<p>Note:</p> <ul style="list-style-type: none"> ■ After changing the properties.cnf file from XML to SEF, reload the wm package for SEF in Integration Server > Management > WmEDI. ■ Ensure that the required schema packages are installed.
<p>InstallSEFSchemaWithDescription</p>	<p>This property is applicable only when the EDISchemaSource configuration is set to <code>sef</code>. If InstallSEFSchemaWithDescription is set as <code>true</code>, while creating a Flat File Schema from a SEF file. The SEF file is read from <code>packages\WmEDI\pub\SEFS\X12_SefDesc</code> instead of <code>packages\WmEDI\pub\SEFS\X12</code>. The default value is <code>false</code>, which reads SEF files from <code>packages\WmEDI\pub\SEFS\X12</code>.</p>

Note:

Any properties you set in the properties.cnf file, will appear in My webMethods. You can edit these properties in the Module for EDI Configuration Properties section of the **Administration > Integration > B2B Settings > Configure Properties** page.

4. Click **Save**.
5. Restart Integration Server.

Setting Up Items Required for Structural Validation and Conversion

Module for EDI uses flat file schemas to parse and validate the structure of inbound EDI documents and to convert documents from EDI format into the format used by your internal applications and vice versa. Module for EDI automatically creates the flat file schemas when you define TN EDI document types as described in [“Defining TN EDI Document Types” on page 34](#).

However, you might still need to perform the following flat file-related tasks:

Task	Where to Find More Information
Configure how you want format services to convert field values in documents	“Configuring Format Services to Convert Field Values” on page 30
Specify how to associate the format service to fields defined in a flat file schema for an EDI document	“Associating the EDI Format Services with EDI Data Types” on page 32

Task**Where to Find More Information**

Create flat file schemas for EDI documents and your internal-format documents [“Creating Flat File Schemas for Internal-Format Documents” on page 34](#)

Configuring Format Services to Convert Field Values

A format service formats the value of a field and optionally validates the field against restrictions defined in the format service. Module for EDI provides format services to convert field values into a different format as required by your system. For example, a format service can change the format of a date field from YYYYMMDD to MMDDYYYY or data from the EDI standard into a different format used by your internal application. You associate format services with specific fields in a flat file schema.

The following EDI format services run automatically when defined as the conversion service for a flat file input schema:

- `wm.b2b.edi:convertToValues` (or `wm.b2b.edi.tradacoms:convertToValues`) service. Converts an EDI transaction set document to an IS document (IData object), which contains the internal document format.
- `wm.b2b.edi:convertToString` (or `wm.b2b.edi.tradacoms:convertToString`) service. Converts 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 assigned format service does not execute.

For more information about the `convertToValues` and `convertToString` services, see *webMethods Module for EDI Built-In Services Reference*. For more information about how to use these services when processing EDI documents, see [“Logic to Include in the Service to Process EDI Documents” on page 172](#).

The format services use the settings in the following configuration file to determine how to perform field conversions:

```
Integration Server_directory\instances\$instance_name\packages\WmEDI\config\format.xml
```

The `format.xml` configuration file specifies the format required by the EDI standard (external format) and your internal application (internal format). By default, the external formats in this configuration file match the EDI ANSI standard format. You must update the configuration so that it reflects the EDI standard that you are using and the formats of your internal application in order for the system to accurately convert field values from one format to another.

To apply conversion configurations, perform the following procedure to update the configuration in memory so changes are applied immediately, as well as in the `format.xml` file.

➤ **To configure field conversion values in documents**

1. In My webMethods: **Administration > Integration > B2B Settings > Configure Properties**.
2. In the Module for EDI Format Settings panel, 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:

```
<FormatService
name="name_of_format_service"
isEnabled="true_or_false"
internalFormatString="format"
externalFormatString="format"
applyFormat="true_or_false"/>
```

For example:

```
<FormatService
name="formatDate6"
isEnabled="true"
internalFormatString="MM/dd/yy"
externalFormatString="yyMMdd"
applyFormat="true"/>
```

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

Setting	Description
name	<p>The name of the format service that the entry governs. Do not change name.</p> <p>The entry <code>formatImpliedDecimal</code> affects all implied decimal services (<code>formatN0-formatN9</code>), which format fields with EDI data types <code>N0 - N9</code>. For a complete description of these and other EDI format services, see <i>webMethods Module for EDI Built-In Services Reference</i>.</p>
isEnabled	<p>Indicates whether the format service is enabled. Specify either true or false.</p> <ul style="list-style-type: none"> ■ true—The format service performs its processing to convert and/or validate field values. ■ false—The format service immediately returns without performing any conversion or validation. <p>Note: You can also configure whether to enable format services in the configuration of the <code>WmFlatFile</code>.</p> <p>Note: If the configuration for the <code>WmFlatFile</code> package specifies not to use format services, the <code>convertToValues</code> and <code>convertToString</code> services will not invoke the format service, even if the <code>isEnabled</code></p>

Setting	Description
	configuration setting is set to true. For more information about how to configure the WmFlatFile package, see <i>Flat File Schema Developer's Guide</i> .
internalFormatString	<p>The format required by your internal application (for example, your back-end system).</p> <p>Specify a string that follows the conventions described in the java classes <code>java.text.DecimalFormat</code> and <code>java.text.SimpleDateFormat</code>. For more information about these classes, see Oracle's documentation.</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 classes <code>java.text.DecimalFormat</code> and <code>java.text.SimpleDateFormat</code>. For more information about these classes, see Oracle's documentation.</p>
applyFormat	<p>Whether the <code>convertToValues</code> or <code>convertToString</code> services should apply the converted value.</p> <ul style="list-style-type: none"> ■ <code>true</code>—The service validates and updates the document to reflect the converted value. ■ <code>false</code>—The service validates, but does not update the document with the converted value.

3. 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 the section on format services in the chapter on creating and editing flat file schemas and dictionaries in *Flat File Schema Developer's Guide*.

Associating the EDI Format Services with 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 this 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 a data type of "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 359.

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:

```
Integration Server_directory\instances\$instance_name\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. In My webMethods: **Administration > Integration > B2B Settings > Configure Properties**.
2. In the Module for EDI Format Settings panel, update each entry in the `format.xml` configuration file as needed. 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 provided with Module for EDI, see the description of the `wm.b2b.edi.util.formatServices` folder in *webMethods Module for EDI 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` with 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"/>
```

3. Resequence the `DatatypeFormat` entries, if necessary.

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

4. Click **Save**.

Creating Flat File Schemas for Internal-Format Documents

If you want to use EDI built-in services to convert internal-format documents (for example, 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. Use the functionality in the WmFlatFile package in Software AG Designer to create the flat file schemas. For more information, see *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, Software AG recommends using flat file schemas.

Defining TN EDI Document Types

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

When you define a TN EDI document type, Module for EDI 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 for Non-TRADACOMS Standards"](#) on page 41 or ["Flat File Schema Namespace Conventions When Using TRADACOMS"](#) on page 42.

Note:

In a clustered environment, you can optionally deploy the EDIFFSchema folder in all the nodes so that the installed document types are available across the cluster.

If you have set the tn.docType.EnableLogDocumentType property to true, Trading Networks logs the details of the new document type in the activity log. For more information about setting Trading Networks configuration properties, see the *webMethods Trading Networks Administrator's Guide* for your release.

Note:

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

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

These are the following three high-level steps for defining TN EDI document types:

Note:

You must have administrator privileges and data permission to define and manage document types in Trading Networks. For information about defining data permissions and role-based access, see the *webMethods Trading Networks Administrator's Guide* for your release.

Step 1: Define Document Type Information

You can use My webMethods to define document type information.

Defining Document Type Information Using My webMethods

This procedure uses My webMethods to define information for the supporting TN EDI document types for each transaction set or file you want to exchange with trading partners. Trading Networks checks documents against this information to determine whether the document matches a defined TN EDI document type.

➤ To define document type information using My webMethods

1. In My webMethods: **Administration > Integration > B2B > Document Types Administration > Document Types**.
2. Click **Add Document Types**.
3. Select **EDI**.
4. In the EDI Document Type dialog box, select the appropriate values for the following fields, and then click **Create**.

Field	Description
Standard	The EDI standard
Version	The version of the selected standard 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 (for example, v4).
Transaction	The transaction set (or TRADACOMS file) of the selected standard and version

Field	Description
	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 (for example, v4).

Module for EDI generates the name of the TN EDI document type based on the standard, the version, and the transaction type you selected. This name is displayed in the **Name** field on the EDI Document Type Details page. You cannot edit the name of the TN EDI document type.

In addition to generating the name of the TN EDI document type, the module installs a flat file schema. For information about flat file schema namespace conventions, see [“Flat File Schema Namespace Conventions” on page 41](#).

- In the EDI Document Type Details page, specify the following:

Field	Description
Description	Optional. The description for the TN EDI document type.
Status	<p>Whether you want the TN EDI document type to be enabled or disabled for identifying a document that passes through your system.</p> <ul style="list-style-type: none"> ■ Enabled—The TN EDI document type is enabled. This is the default. ■ Disabled—The TN EDI document type is disabled. For information about the effects of disabling a TN document type, see the <i>webMethods Trading Networks Administrator’s Guide</i> for your release. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: You can keep the TN EDI document type disabled until all the information for the TN EDI document type is specified.</p> </div>

Step 2: Define Attributes to be Extracted from Documents

You must specify the attribute values to extract from the document's *TN_parms* parameter. You can extract both system attributes and custom attributes (the document data). In addition, you can use built-in or custom transformations to transform extracted attribute values before Trading Networks stores them in the BizDocEnvelope.

For VDA documents, the extraction query specified in the document type should be able to extract any field value in the header of the message. You must specify the sender and receiver attributes that you want Trading Networks to extract from the documents.

For TRADACOMS documents, you cannot extract attributes for TRADACOMS headers, messages, and trailers separately. Trading Networks does not separate these sections in the attribute extraction dialog box. You can extract attributes only from the message section of a TRADACOMS document. For more information about the attributes that Module for EDI stores for TRADACOMS document types, see [“Attributes Stored for TRADACOMS Document Types” on page 38](#).

For more information about document attributes, see [“Trading Networks Attributes and EDI Documents” on page 189](#) and the *webMethods Trading Networks Administrator’s Guide* for your release.

For information about overwriting attributes, see [“Overwriting Document Attributes Extracted from Inbound Documents” on page 46](#).

➤ **To define the attributes that you want Trading Networks to extract from the document**

1. Complete the steps in [“Step 1: Define Document Type Information” on page 35](#).
2. On the EDI Document Type Details page, click the **Extract** tab. The document tree view displays the contents of the installed TN EDI document type.
3. For each attribute that you want Trading Networks to extract from documents identified by this TN EDI document type:
 - a. Expand the document tree view, and select a node that represents the attribute that you want Trading Networks to extract.
 - b. Do one of the following to display the Add Extracted Attribute dialog box:
 - Click  next to the **Query** field, and then select **Add Attribute**.
 - In the EDI document tree view panel, click  in the **Action** column, and then select **Add Attribute**.
 - c. In the Add Extracted Attribute dialog box, specify the following:

Field	Description
Query	<p>The XQL query that instructs Trading Networks how to extract the attribute for the particular TN EDI document type.</p> <p>Trading Networks fills the Query field with the XQL query for the node that you selected.</p> <p>For information about extracting attributes specifying an XQL query of a node in the document tree, see “Extracting a Document Attribute Using an XQL Query” on page 38.</p>
Name	<p>The name of the attribute (for example, SenderID, ReceiverID, or ConversationID). Select a name from the list.</p> <p>Trading Networks fills in the Type and Description fields based on the document attribute definition.</p>
Required	<p>Whether you want Trading Networks to log an error message in the activity log when it cannot extract the attribute.</p>

Field	Description
Transformation	How Trading Networks handles attribute data before storing it in the Trading Networks database. The data type of the attribute (shown in the Type field) determines the built-in or custom transformation options available.

For VDA documents, you must also select the type of the external IDs (for example, DUNS) from the **Built-in** list for both SenderID and ReceiverID attributes so that Trading Networks can match the external IDs in SenderID and ReceiverID with the external IDs in the profiles. This is how Trading Networks verifies the identities of the sender and receiver.

- d. Click **OK**.

For information about using an XQL query of a node in a document tree to extract a document attribute, see [“Extracting a Document Attribute Using an XQL Query” on page 38](#).

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 of these document types.

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)

Extracting a Document Attribute Using an XQL Query

You can use an XQL query of a node in the document tree to extract a document attribute by following this procedure. For examples, see [“Examples of Extracting a Document Attribute Using an XQL Query” on page 39](#).

➤ **To extract a document attribute by specifying an XQL query of a node in the document tree**

1. In My webMethods: **Administration > Integration > B2B > Document Types Administration > Document Types**.
2. Select the document type for which you want to specify attributes to extract and then click the **Action** tab.
3. On the **Extract** tab, in the document tree view, select a node that represents the attribute that you want Trading Networks to extract.
4. Click  either next to the **Query** field or in the **Action** column in the EDI document tree view panel.
5. Select **Add Attribute**.
6. In the Add Extracted Attribute dialog box, specify the values, as described in [“Step 2: Define Attributes to be Extracted from Documents”](#) on page 36.
7. Click **OK**.

Examples of Extracting a Document Attribute Using an XQL Query

The examples provided are based on the following EDI document:

```
ST*850*0001!
BEG*01*BK*99AKDF9DAL393*39483920193843*20021106*AN3920943*AC*IBM*02*AE*02*BA!
CUR*AC*USA*.2939*SE*USA*IMF*002*20021106*1615*021*20021106*0718*038*20021106*195
2*002*20021106*1615*021*20021106*0718!
REF*AB*3920394930203*GENERAL
PURPOSE*BT:12345678900987654321768958473:CM:500:AB:3920394930203!
REF*BC*3920394930203*GENERAL
PURPOSE*BT:12345678900987654321768958473:CM:500:AB:3920394930203!
...
SE*196*0001!
```

Example 1. To extract the first field, 01, in the BEG record of the sample EDI document, specify the following XQL query: `/ST/BEG/BEG01/code`

Example 2. To extract the first field, AB, in the first REF record of the sample EDI document, specify the following XQL query: `/ST/REF[0]/REF01/code`

Example 3. To extract the first field, BC, in the second REF record of the sample EDI document, specify the following XQL query: `/ST/REF[1]/REF01/code`

Step 3: Define Processing Options

Once you have identified the attributes that you want Trading Networks to extract from the document, you need to define the processing options you want Trading Networks to consider when matching a document to a TN EDI document type. Use the following procedure to define options.

➤ **To define processing options for a TN EDI document type**

1. Complete the steps in “[Step 2: Define Attributes to be Extracted from Documents](#)” on page 36.
2. Select the document type for which you want to define options and click the **Options** tab.
3. On the EDI Document Type Details page, specify the following:

Field	Description
Enable Processing Rule Routing	Whether you want Trading Networks to search for a processing rule and use the rule to process the document.
Validate Structure (pre-processing action)	Whether you want Trading Networks to validate the structure of the document.
Save (pre-processing action)	<p>Whether you want Trading Networks to save the document to the database.</p> <p>If you select to save the document to the database, select one of the following from the list:</p> <ul style="list-style-type: none">■ All documents■ Only unique documents <p>If you choose to save the document to the database, select the options to specify whether you want Trading Networks to save the following document information to the database:</p> <ul style="list-style-type: none">■ Content■ Attributes■ Activity Log <p>Trading Networks performs this pre-processing action before it performs the processing action defined by the processing rule.</p>

4. Do one of the following:
 - To save the changes you have made and continue defining the TN EDI document type, click **Save**.
 - To save the changes and close the TN EDI document type, click **Save and Close**.

Creating a TN EDI Document Type Based on a Transaction

You can create an EDI document type based on a transaction that you have saved in Trading Networks.

➤ **To create a TN EDI document type based on a transaction saved in the Trading Networks**

1. In My webMethods: **Monitoring > Integration > B2B > Transactions**.
2. Click  on the row containing the 'Unknown' document that you want to use for creating the document type.
3. Select **Create Document Type**.
4. In the Create Document Type dialog box, click **EDI** to select the document type category.
5. In the EDI Document Type dialog box, select the appropriate values for **Standard, Version,** and **Transaction**.
6. Click **Create**.
7. In the EDI Document Type Details page, specify the details.
8. Do one of the following:
 - Click **Save** to save the changes you have made and continue defining the TN EDI document type.
 - Click **Save and Close** to save the changes and close the TN EDI document type.
 - Click **Cancel** if you do not want to save the TN EDI document type.

Flat File Schema Namespace Conventions

When you define a TN EDI document type, Module for EDI creates a flat file schema for the same EDI transaction set or TRADACOMS file. Trading Networks stores flat file schemas in the EDIFFSchema folder of the WmEDIforTN package, using the namespace conventions described below.

Flat File Schema Namespace Conventions for Non-TRADACOMS Standards

For non-TRADACOMS standards, Trading Networks uses the following namespace convention for flat file schemas:

```
EDIFFSchema.Standard.VVersion:TTransactionSet
```

where *Standard* is the name of the EDI standard, *version* is the version number of the EDI standard, and *TransactionSet* is the EDI transaction set. For example:

- For a TN EDI document type for version 4010 of an ANSI X12 850, Trading Networks creates the following flat file schema in the WmEDIforTN package: EDIFFSchema.X12.V4010:T850

- For a TN EDI document type for version 99A of the UN/EDIFACT standard with a transaction set of ORDERS, Trading Networks creates the following flat file schema in the WmEDIforTN package: `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 When Using 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 38.

Managing TN EDI Document Types

You can perform the following tasks on TN EDI documents types:

- Enable and disable
- Search
- View
- Update

Viewing the Code List Details

➤ **To view the code list descriptions of a segment for a document type using My webMethods**

Note:

- The code list descriptions are available only for the document types based on the XML schemas for the EDI schema packages available in the installer. By default, Module for EDI installs the document types from the XML schemas which are available in the EDI schema packages.
- The code list descriptions in My webMethods are displayed only for the default values of the installed EDI standards. If these values are modified and any additional values are added to these standards in Designer, then the changes do not reflect in My webMethods.

1. In My webMethods: **Administration > Integration > B2B > Document Types Administration > Document Types**.
2. Click on an EDI Document Type.
3. Navigate to the **EDI Document Type Details > Extract** tab and expand the document type tree at the **UUID** level.
4. Click on the corresponding **Lookup** link for a document type.

The CodeList Lookup dialog box lists all the code IDs, names, and descriptions for the document type. You can also search for a specific code ID using the **Search Code ID > ID** field.

Viewing Alternate Name and Description of Segments

➤ To view the alternate name and description of a segment for an XML schema

1. In **Software AG Designer: Package Navigator > WmEDIforTN > EDIFFSchema > Standard**, expand any version folder and double click on a transaction.
2. In the **Properties** tab, click on a segment to view the properties.

The **Alternate name** and **Description** properties of the segment are displayed.

Updating TN EDI Document Types

This section describes how to update TN EDI document types. For more information about performing any of the other tasks listed, see the *webMethods Trading Networks Administrator's Guide* for your release.

You can perform the following update tasks on TN EDI document types:

Task	Where to Find More Information
Update the description and status of the TN EDI document type	“Updating the Description or Status of a TN EDI Document Type” on page 44

Task	Where to Find More Information
Update the processing options of the TN EDI document type	“Updating the Processing Options Specified in the Document Type” on page 44
Update the TN EDI document type based on a transaction	“Updating a TN EDI Document Type Based on a Transaction” on page 45

You cannot update the name, standard, and version of a TN EDI document type.

When you update a TN EDI document type, the changes are logged in the activity log if you have set the `tn.docType.EnableLogDocumentType` property to true. For more information about setting Trading Networks configuration properties, see the *webMethods Trading Networks Administrator’s Guide* for your release.

Updating the Description or Status of a TN EDI Document Type

Use this procedure to update the description or status of a TN EDI document type. For more information about the description and the status of a TN EDI document type, see [“Defining TN EDI Document Types” on page 34](#).

➤ To update the description or status of a TN EDI document type

1. In My webMethods: **Administration > Integration > B2B > Document Types Administration > Document Types**

Trading Networks displays all the document types.

2. In the row containing information about the TN EDI document type that you want to update, click  **Edit**.
3. In the EDI Document Type Details page, edit the **Description** and **Status** fields as required, and then click **OK**.

Updating the Processing Options Specified in the Document Type

➤ To update the options specified in the document type

1. In My webMethods: **Administration > Integration > B2B > Document Types Administration > Document Types**.
2. Click the **Options** tab, and then select the required options.

For more information about these options, see [“Defining TN EDI Document Types” on page 34](#).

3. Do one of the following:
 - To save the changes you have made and continue defining the TN EDI document type, click **Save**.
 - To save the changes and close the TN EDI document type, click **Save and Close**.

Updating a TN EDI Document Type Based on a Transaction

➤ To update a TN EDI document type based on a transaction saved in Trading Networks

1. In My webMethods: **Monitoring > Integration > B2B > Transactions**.
2. Click  on the row containing the document that you want to use for editing the document type. Select **Edit Document Type**.
3. In the Document Type Details page, edit the fields in the **Extract** and **Options** tabs as required.

For more information about updating TN EDI document types using a saved transaction, see the *webMethods Trading Networks Administrator's Guide* for your release.

4. Do one of the following:
 - To save the changes you have made and continue editing the document type, click **Save**.
 - To save the changes and close the TN EDI document type, click **Save and Close**.

Managing Document Attributes

You can perform the following tasks on document attributes:

Task	Where to Find More Information
Add a document attribute	“Adding a Document Attribute” on page 45
Edit a document attribute	“Editing a Document Attribute” on page 46
Delete a document attribute	“Deleting a Document Attribute” on page 46

Adding a Document Attribute

➤ To add a document attribute

1. In My webMethods: **Administration > Integration > B2B > Document Types Administration > Document Attributes**.
2. Click **Add Document Attributes**.

3. In the **Name** field, enter the name of the attribute.
4. In the **Type** field, select the appropriate value from the drop-down list.
5. In the **Description** field, enter a description for the document attribute.
6. Click **OK**.

Editing a Document Attribute

> To edit a document attribute

1. In My webMethods: **Administration > Integration > B2B > Document Types Administration > Document Attributes**.
2. In the row containing information about the document attribute that you want to update, click  **Edit**.
3. Edit the **Name** and **Description** fields as required.
4. Click **OK**.

Deleting a Document Attribute

> To delete a document attribute

1. In My webMethods: **Administration > Integration > B2B > Document Types Administration > Document Attributes**.
2. Click the row containing the document attribute you want to delete, and then click **Delete**.

Overwriting Document Attributes Extracted from Inbound Documents

To simplify document recognition by Trading Networks, you can overwrite attribute values that are extracted from inbound documents with values of your own. To overwrite any custom or system attribute, add the attribute with the new value in the *TN_parms* parameter of the *wm.tn:receive* service. Trading Networks extracts the value from *TN_parms* and saves it to the database to use later. For more information about the *wm.tn:receive* service, see the *webMethods Trading Networks Built-In Services Reference* for your release.

You can overwrite the values of system and custom attributes for Transaction, Group, and Envelope document types depending on the type of transaction. Trading Networks displays the new value of the attribute on the **Attributes** tab in My webMethods.

For more information about document attributes, see “[Trading Networks Attributes and EDI Documents](#)” on page 189 and the *webMethods Trading Networks Administrator’s Guide* for your release.

EDI Transaction

The following tables show the system and custom attributes that you can overwrite for an EDI transaction, where:

- Yes means that the value of the attribute you specify overwrites the default value.
- No means that you cannot overwrite the attribute.
- NA means that the attribute is not present in the document.

System Attributes

Attribute	Transaction Document	Group Document	Envelope Document
ConversationID	Yes	Yes	Yes
DocumentID	Yes	Yes	Yes
GroupID	Yes	Yes	Yes

Custom Attributes

Attribute	Transaction Document	Group Document	Envelope Document
EDI Batch	No	Yes	Yes
EDI Group Type	NA	Yes	Yes
EDI Outbound FA	NA	NA	Yes
EDI Processing Mode	No	Yes	Yes
EDI Status	No	Yes	Yes
EDI Version	NA	Yes	Yes
Envelope CntrlNum Status	No	No	No
Group CntrlNum Status	No	No	NA
EDI FA Status	No	Yes	NA

TA1 Transaction

The following tables show the system and custom attributes that you can overwrite for a TA1 transaction, where:

- Yes means that the value of the attribute you specify overwrites the default value.

- No means that you cannot overwrite the attribute.
- NA means that the attribute is not present in the document.

System Attributes

Attribute	Transaction (TA1) Document	Envelope Document
ConversationID	No	Yes
DocumentID	No	Yes
GroupID	No	Yes

Custom Attributes

Attribute	Transaction (TA1) Document	Envelope Document
EDI Batch	NA	Yes
EDI Group Type	NA	Yes
EDI Outbound FA	NA	Yes
EDI Processing Mode	NA	Yes
EDI Status	NA	Yes
EDI Version	NA	Yes
Envelope CntrlNum Status	NA	No
Group CntrlNum Status	NA	NA
TA1 Status	NA	No
TA1 Code	NA	No
EDI FA Status	NA	NA

FA Transaction

The following tables show the system and custom attributes that you can overwrite for an FA transaction, where:

- Yes means that the value of the attribute you specify overwrites the default value.
- No means that you cannot overwrite the attribute.
- NA means that the attribute is not present in the document.

System Attributes

Attribute	Transaction (TA1) Document	Envelope Document	Group Document
ConversationID	Yes	Yes	Yes
DocumentID	Yes	Yes	Yes
GroupID	Yes	Yes	Yes

Custom Attributes

Attribute	Transaction (TA1) Document	Envelope Document	Group Document
EDI Batch	No	Yes	Yes
EDI Group Type	NA	Yes	Yes
EDI Outbound FA	NA	Yes	NA
EDI Processing Mode	No	Yes	Yes
EDI Status	No	Yes	Yes
EDI Version	NA	Yes	Yes
Envelope CntrlNum Status	No	No	No
Group CntrlNum Status	No	NA	No
LateFA	NA	NA	No
EDI FA Status	No	NA	Yes

Adding New Versions of EDI Standards

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

Important:

webMethods Module for EDI with Trading Networks Server on Microservices Runtime works with Integration Server 10.15 Core Fix 2 or higher.

➤ To add a new version of an EDI standard from SEF

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. The file must contain all the transaction sets that the version supports.

- In Designer's navigation panel, run the following built-in service located in the WmEDIforTN package: `wm.b2b.editn.util.VersionSupport:addNewEDIVersion`. This service creates a new TN document type file in the config directory of the WmEDIforTN package. The new version will be available when defining TN document types. Specify the following input parameters for the service:

Input Parameter	Description
<i>SEF fileName</i>	The fully qualified path of the SEF file.
<i>Replace?</i>	Whether to overwrite an existing SEF file of the same name and its associated TN document type file. yes — Overwrite an existing SEF file of the same name and its associated TN document type file. no — Do not overwrite an existing SEF file of the same name and its associated TN document type file. This is the default.

Note:

The service does not update the envelope segments structure or the validation for individual fields. However, the service does update the following fields in the IS document type `wm.b2b.editn.rec:EDIHEADERS` so that Trading Networks recognizes the new version during validation:

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.editn.util.VersionSupport:addNewEDIVersion` service, see *webMethods Module for EDI Built-In Services Reference*.

- Reload the WmEDIforTN package.
- Install the new TN EDI document type for the transaction set, as described in [“Defining TN EDI Document Types” on page 34](#). Module for EDI creates the TN EDI document type and the SEF files.

**Module for EDI
creates...**

In the following directory...

TN EDI document type WmEDIforTN\config\EDI_Standard\docType.xml

where *EDI_Standard* is X12, UNEDIFACT, ODETTE, EANCOM, VICS, or UCS.

SEF file

Integration_Server_directory\instances*Instance_name*\packages\WmEDI\pub\SEF\EDI_Standard\using.sef

Note:

Integration_Server_directory\instances*Instance_name*\packages\WmEDI\pub\SEF\EDI_Standard\using.sef
When **InstallSEFSchemaWithDescription** is set to true.

Note:

For UN/EDIFACT, ODETTE, and EANCOM, the system adds a CONTRL message in the .SETS section of the SEF file.

Important:

After you install a new version of an EDI standard and the corresponding TN EDI document type, *do not* re-install that version of the EDI standard. If you re-install it, the internal ID of the document type will change, and any references to that ID will be incorrect. In addition, any Trading Networks processing rules that refer to the document type will also be incorrect.

3 Defining Trading Partner Information

- Overview 54
- Partner Information You Need to Define 54
- Defining Trading Partner Profiles 55
- Adding External ID Types for EDI ID Qualifiers 55
- Adding and Overriding EDI ID Qualifiers 56
- Partner Profile ID Types Mapping for EDI Documents 57
- Defining EDI Trading Partner Agreements 59

Overview

To process documents when you are using webMethods Trading Networks with webMethods Module for EDI, you must define both trading partner profiles and trading partner agreements (EDITPAs) for the partners with whom you will trade EDI documents. How you define information for your partners depends on whether you want Module for EDI to use standard or non-standard processing. For more information, see *webMethods Module for EDI Concepts Guide*.

Important:

The chapters in this guide describe how to use standard processing. When you want to use non-standard processing, certain tasks must be performed differently. Such tasks are indicated by the  graphic, as well as a reference to the corresponding section in “[Non-Standard Processing](#)” on page 331. When you are using standard processing, you can ignore these references.

Partner Information You Need to Define

The following table lists the information you must define for interchange and group sender/receiver pairs with whom you will trade EDI documents.

Note:  For more information about defining partners when using non-standard processing, see “[Defining Partner Information When Using Non-Standard Processing](#)” on page 332. For more information about the difference between standard and non-standard processing, see *webMethods Module for EDI Concepts Guide*.

For...	Define...
Interchange sender/receiver pairs	Trading partner profiles for each interchange sender and receiver. For more information, see “ Defining Trading Partner Profiles ” on page 55.
Group sender/receiver pairs	Trading partner profiles for each group sender and receiver. For more information, see “ Defining Trading Partner Profiles ” on page 55.
Tailoring how Module for EDI processes documents	<ul style="list-style-type: none"> ■ A default EDITPA, which defines the settings that you want to use for most partner pairs. ■ A partner-specific EDITPA for each interchange sender/receiver pair for which you want to override the default EDITPA settings. <p>For more information, see “Defining EDI Trading Partner Agreements” on page 59.</p>
How Module for EDI validates inbound control numbers	<ul style="list-style-type: none"> ■ Whether to validate control numbers. ■ Actions Module for EDI should take when it encounters an invalid control number.

For...	Define...
	<ul style="list-style-type: none"> ■ Control number validation settings (that is, maximum, minimum, increment, and window). <p>For more information, see “Validating Inbound Control Numbers” on page 230.</p>

For more information about how Module for EDI works with Trading Networks, see *webMethods Module for EDI Concepts Guide*.

Defining Trading Partner Profiles

You must set up profiles for all the trading partners with whom you want to exchange documents. 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.

Note: **Non-standard** When you use non-standard processing, you must create profiles only for partners at the group level. For more information, see [“Defining Trading Networks Profiles When Using Non-Standard Processing”](#) on page 333. For more information about the difference between standard and non-standard processing, see *webMethods Module for EDI Concepts Guide*.

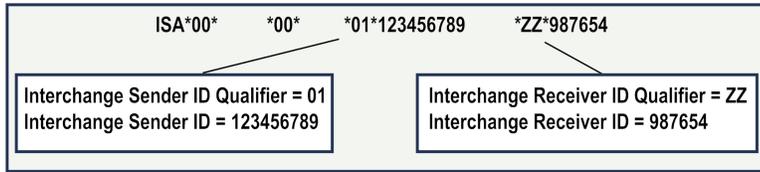
When you define trading partner profiles, you need to specify the external IDs your partners use in their documents. The external IDs correspond to the standard EDI ID qualifiers. Trading Networks does not provide external IDs for all EDI ID qualifiers; however, Module for EDI provides a way for you 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. For more information, see [“Adding External ID Types for EDI ID Qualifiers”](#) on page 55.

For instructions about creating trading partner profiles, see the *webMethods Trading Networks Administrator’s Guide* for your release.

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 example shows an ANSI X12 interchange envelope header.



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. To Trading Networks, you can add 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.

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

1. In My webMethods: **Administration > Integration > B2B > Partner Administration > Partner Profiles**.
2. Either click the desired profile or click  **Edit** next to the desired profile.
3. On the **External IDs** tab, click **Add ID Type**.
4. Type the name of the external ID type and click **OK**.
5. Do one of the following:
 - To assign a value to the new external ID type, select the ID type from the **ID Type** list and then type a value in the **Value** box.
 - To add a new external ID, click **Add ID**, select the ID type from the **ID Type** list, type a value in the **Value** box, and click **OK**.

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.”

Module for EDI 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 in the following table.

Step	Description
1	In any user-defined package, create a service that overrides the value of the ID qualifier code and name it <code>custom.EDIIDMapping:addEDIIDMapping</code> .
2	In your flow, create the <code>custom.EDIIDMapping:addEDIIDMapping</code> service.
3	To the <code>custom.EDIIDMapping:addEDIIDMapping</code> service, add the <i>qualifier</i> input parameter and the <i>idType</i> output parameter. They are both of data type String.
4	In the <code>custom.EDIIDMapping:addEDIIDMapping</code> service, branch on <i>qualifier</i> 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 <i>idType</i> with any value, such as "User Defined 1."

For more information about creating custom flow services, see the *webMethods Service Development Help* for your release. For a sample service, see the Technical community area of the Empower Product Support website at <http://communities.softwareag.com/>.

Partner Profile ID Types Mapping for EDI Documents

The following table lists the ID type mapping found in EDI documents to represent the identity types in partner profiles:

Standard Code	ID Type
01	DUNS
02	SCAC
03	FMC
04	IATA
5	INSEE SIRET
07	GLN
08	UCC ID
09	X.121
10	DoD Code
11	DEA
12	Phone
13	UCS Code
14	EAN

Standard Code	ID Type
15	PAS Code
16	DUNS+4
17	ABA Routing
18	AIAG
19	EDICA ID
20	HIN
21	IPEDS
22	INSEE SIREN
23	NCES
24	ATP
25	4-Digit Code List of Postsecondary Institutions
26	Stat of List of Postsecondary Institution
27	Carrier ID by HCFA
28	Fiscal Intermediary ID by HCFA
29	Medicare Provider and Supplier ID by HCFA
30	ISO 6523
31	DIN
32	FEIN
33	BfA
34	Medicare Provider and Supplier ID by states
35	Stat College Student Info System Institution Codes
36	Stat University Student Info System Institution Codes
37	Society of Property Info Compilers and Analysts
38	6-Digit Code List of Secondary Institutions
51	GEIS
54	BdDB
55	Bank ID

Standard Code	ID Type
91	seller agent
92	buyer agent
AM	AMECOP ID
NR	NRMA
SA	ID by SAFER
SN	Standard Address Number
ZZ, ZZZ	Mutually Defined

Defining EDI Trading Partner Agreements

An EDI trading partner agreement (EDITPA) is a set of variables that you provide to tailor how Module for EDI exchanges documents between two trading partners. The module supports partner-specific EDITPAs and a single default EDITPA. Each EDITPA contains variables that Module for EDI uses only when processing documents that match the values specified in the EDITPA.

- **A partner-specific EDITPA** is unique for a specific sender and receiver pair and contains variables that the module uses only when processing documents that match the specified sender/receiver pair.
- **A sender-specific EDITPA** specifies a sender but leaves the receiver defined as “unknown.” The value “unknown” equates to “any,” thus matching all document types to the EDITPA.
- **A receiver-specific EDITPA** identifies the receiver but leaves the sender defined as “unknown.”
- **A default EDITPA** defines both the sender and receiver as “unknown.” A default EDITPA contains variables used by all trading partners when partner-specific information is not available. The settings in the default EDITPA should meet the requirements of the majority of your trading partner relationships. You can modify the default EDITPA. For more information, see [“Modifying the Default EDITPA” on page 60](#).

During document processing, Module for EDI identifies the EDITPA to use for the sender/receiver pair of a document as follows:

1. When a partner-specific EDITPA exists, the module uses the values specified in the partner-specific EDITPA.
2. When a partner-specific EDITPA does not exist and a sender specific EDITPA does exist, Module for EDI uses the values specified in the sender-specific EDITPA.
3. When neither a partner-specific EDITPA nor a sender-specific EDITPA exists, and a receiver-specific EDITPA does exist, the module uses the values specified in the receiver-specific EDITPA.

4. When a partner-specific EDITPA, a sender-specific EDITPA, and a receiver-specific EDITPA do not exist, the module uses the value from the default EDITPA.

Modifying 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 Integration Server, Module for EDI automatically creates the default EDITPA in Trading Networks. Using My webMethods, 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. For more information about modifying EDITPAs, see the *webMethods Trading Networks Administrator's Guide* for your release.

Note:

Do not disable the default EDITPA. Module for EDI uses the settings in the default EDITPA when values are missing in the partner-, sender-, or receiver-specific EDITPAs.

When you modify the default EDITPA, in My webMethods, specify the following settings on the Administration > Integration > B2B > Trading Partner Agreements page:

For this 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 wm.b2b.editn.TPA:EDITPA IS document type. For more information, see " wm.b2b.editn.TPA:EDITPA IS Document Type " on page 62.

Defining a Partner-Specific EDITPA

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

Using My webMethods, you can create a partner-specific EDITPA in one of the following ways:

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

For instructions about how to create TPAs (either by duplication or from scratch), see the *webMethods Trading Networks Administrator's Guide* for your release.

When creating a partner-specific EDITPA, specify the following fields on the Administration > Integration > B2B > Trading Partner Agreements > Trading Partner Agreement Details page:

Field	Description
Agreement ID	EDITPA
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.
Control Number	The value that the application using the TPA expects (only relevant if the application uses the Control Number field). The value for this field must be an integer. The default is zero (0).
Description	(Optional) A description for the TPA. You can use up to 1024 characters of any type.
Data Status	Whether to modify the values of the TPA data of the IS document type. This field only applies when the agreement status is Agreed.
Initialization Service	A service that you created to define the default values for the <code>wm.b2b.editn.TPA:EDITPA</code> IS document type. (Or, leave this field blank.) Software AG recommends that you do not use the initialization service (<code>wm.b2b.editn.TPA:initService</code>) that is provided with Module for EDI. The <code>wm.b2b.editn.TPA:initService</code> service is used to populate the values of the default EDITPA.
Validation Service	(Optional) A service to validate the TPA. For more information about the validation service, see the <i>webMethods Trading Networks Administrator's Guide</i> for your release.
Export Service	The service to convert the data in the Trading Networks TPA to an industry-format trading partner agreement. For more information about the export service, see the <i>webMethods Trading Networks Administrator's Guide</i> for your release.
IS Document Type	<code>wm.b2b.editn.TPA:EDITPA</code> For information about setting values for the <code>wm.b2b.editn.TPA:EDITPA</code> variables, see " wm.b2b.editn.TPA:EDITPA IS Document Type " on page 62.

Note:

You can disable partner-specific EDITPAs. When you disable a partner-specific EDITPA, Module for EDI uses the values in the default EDITPA.

Note: **Non-standard** When you are using non-standard processing, define partner-specific EDITPAs for group-level sender/receiver pairs. For more information, see [“Defining a Partner-Specific EDITPA When Using Non-Standard Processing”](#) on page 333. For more information about the difference between standard and non-standard processing, see [“Defining a Partner-Specific EDITPA When Using Non-Standard Processing”](#) on page 333.

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

At run time, Module for EDI uses a default value for a variable when the value for that variable is neither specified in a partner-specific EDITPA nor the default EDITPA. When the `wm.b2b.editn.TPA:initService` service initially creates the default EDITPA, the module sets the default values.

Review the variable descriptions to determine the value you want to specify in the default EDITPA 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. The descriptions and default values for the variables of the `wm.b2b.editn.TPA:EDITPA IS` document type are listed below.

Tip:

It is helpful to understand how Module for EDI processes EDI documents and uses the variables in the EDITPA. For more information, see *webMethods Module for EDI Concepts Guide*.

For details about specific EDITPA variables, see the following locations:

EDITPA Variable	Where to Find More Information
<i>GSRouting</i>	“GSRouting Variables” on page 63
<i>persistMultipleDocEnvelope</i>	“PersistMultipleDocEnvelope Variable” on page 65
<i>processingMode</i>	“processingMode Variable” on page 66
<i>splitOption</i>	“splitOption Variable” on page 67
<i>FARconciliation</i>	“FARconciliation Variable” on page 68
<i>UNAmode</i>	“UNAmode Variable” on page 69
<i>publishBatchFailEvent</i>	“publishBatchFailEvent Variable” on page 70
<i>delimiters</i>	“delimiters Variables” on page 70
<i>envelopeIdentifier</i>	“envelopeIdentifier Variables” on page 71
<i>IHeaderInfo</i>	“IHeaderInfo Variables” on page 73
<i>FAGeneration</i>	“FAGeneration Variables” on page 75
<i>ControlNumberManagement</i>	“ControlNumberManagement Variables” on page 83

EDITPA Variable	Where to Find More Information
<i>useReverseRoute</i>	“useReverseRoute Variable” on page 86
<i>BatchCriteria</i>	“BatchCriteria Variables” on page 87
<i>EDITransportMedia</i>	“EDITransportMedia Variable” on page 89
<i>persistOriginalEnvelope</i>	“persistOriginalEnvelope Variable” on page 89
<i>documentPersistOrder</i>	“documentPersistOrder Variable” on page 90
<i>LateFA</i>	“LateFA Variables” on page 91
<i>X12TA1Generation</i>	“X12TA1Generation Variables” on page 93
TRADACOMS	“Defining Trading Partner Information When Using TRADACOMS” on page 97
EANCOMAUTACK	“EANCOMAUTACK Variables” on page 94

GSRouting Variables

Module for EDI creates Interchange, Group, and Transaction documents based on the *splitOption* variable in the EDITPA. The *GSRouting* variables indicate the value the module uses for the sender and receiver in these documents.

The values that you select for the *GSRouting* variables determine:

- How you define criteria for Trading Networks processing rules when you create them for the Interchange, Group, and Transaction documents. You can use the sender and receiver as criteria in the processing rules.
- The external ID types and values that you define in the partner profiles. For Trading Networks to match the sender and/or receiver criteria in a processing rule, you must define a partner profile with an external ID type and value that match the value that the module will put in the Interchange, Group, and Transaction documents.

For more information about creating processing rules and defining external IDs in trading partner profiles, see the *webMethods Trading Networks Administrator’s Guide* for your release.

GSRouting/routingMode EDITPA Variable

Default: OFF

The *routingMode* variable indicates how Module for EDI obtains the sender and receiver for each type of document. The following table provides the possible values for this 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
```

Value	Description
GSOnly	<p>Module for EDI determines the sender and receiver as follows:</p> <ul style="list-style-type: none">■ For Interchange documents, the module uses 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 module uses 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.
GS&ISA	<p>Module for EDI determines the sender and receiver as follows:</p> <ul style="list-style-type: none">■ For Interchange documents, the module uses 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 module derives the sender by concatenating ISA05, ISA06, and GS02 fields, using a colon (:) separator, and 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 always ZZ (Mutually Defined) when you use GS&ISA.
OFF	<p>Module for EDI determines the sender and receiver as follows:</p> <p>For Interchange, Group, and Transaction documents, the module uses the interchange header.</p> <p>For example, the sender uses the EDI ID qualifier "01" with the value "123456789" and the receiver uses the EDI ID qualifier "ZZ" with the value "987654321."</p> <div data-bbox="386 1570 1268 1776" style="background-color: #f0f0f0; padding: 10px;"><p>Note: This setting does not apply when using non-standard processing. When <i>routingMode</i> is OFF and you are using non-standard processing, Module for EDI uses the GSOOnly setting. For more information about group level processing, see <i>webMethods Module for EDI Concepts Guide</i>.</p></div>

***GSRouting/senderQualifier* EDITPA Variable**
Default: *

The *senderQualifier* variable indicates the EDI ID qualifier that Module for EDI uses for the sender when the *GSRouting/routingMode* variable is *GSonly*.

Value	Description
*	Module for EDI uses the EDI ID qualifier for the sender from the interchange header (for example, 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 documentation for your EDI standard and version.

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 55](#).

GSRouting/receiverQualifier EDITPA Variable

Default: *

The *receiverQualifier* variable indicates the EDI ID qualifier that Module for EDI uses for the receiver when the *GSRouting/routingMode* variable is *GSOnly*.

Value	Description
*	Module for EDI uses the EDI ID qualifier for the receiver from the interchange header (for example, 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.

Merk:

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 55](#).

PersistMultipleDocEnvelope Variable

persistMultipleDocEnvelope EDITPA Variable

Default: true

The *persistMultipleDocEnvelope* variable indicates whether Module for EDI saves the original document in the Trading Networks database. The original EDI document usually contains multiple interchange segments. Module for EDI only uses the *persistMultipleDocEnvelope* variable from the default EDITPA.

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

Value	Description
true	Module for EDI saves the original EDI document in the Trading Networks database. The document is saved with both the sender and receiver set to “unknown.”
false	Module for EDI does not save the original EDI document in the Trading Networks database. When you specify false, there is no way to retrieve the original EDI document later.

processingMode Variable

processingMode EDITPA Variable

Default: Testing

The *processingMode* variable indicates whether the partners using the EDITPA are in testing, production, or custom mode. When processing Interchange, Group, and Transaction documents, Module for EDI includes the custom attribute EDI Processing Mode in the BizDocEnvelope and sets the value of this attribute based on the *processingMode* EDITPA variable.

In your processing rules for Interchange, Group, and Transaction documents, you can use the EDI Processing Mode custom attribute to customize processing according to the partner's processing mode. For example, you might set up two processing rules: one for when partners are in testing mode and another for when partners are in production mode.

For more information about custom attributes and using them when creating processing rules, see the *webMethods Trading Networks Administrator's Guide* for your release.

Value	Description
Testing	Use when testing the exchange of documents between two partners. For example, when the production mode is Testing, you might create processing rules that accept the documents and do all processing except passing the document to production applications.
Production	Use when you are confident that your logic for exchanging documents is successful.
Custom	Use when you want to define your own setting.
Interchange Define	Module for EDI sets the value of EDI Processing Mode based on the processing mode defined in the interchange header. <ul style="list-style-type: none">■ For an ANSI X12 document, the module defines the processing mode based on the value of ISA015, as follows:<ul style="list-style-type: none">■ T or others—Sets the EDI processing mode attribute to Testing■ P—Sets the processing mode attribute to Production

Value	Description
	<ul style="list-style-type: none"> ■ I—Sets the processing mode attribute to Custom ■ For a UN/EDIFACT document, the module defines the processing mode attribute based on the value of UNB11, as follows: <ul style="list-style-type: none"> ■ 1,2,3,4 or others—Sets the processing mode attribute to Testing ■ Empty—Sets the processing mode attribute to Production

splitOption Variable

splitOption EDITPA Variable

Default: Transaction

The *splitOption* variable defines how Module for EDI splits an interchange segment within an EDI document. The module uses *splitOption* to determine which of the following document types to create from the interchange segment:

- Interchange documents—Contain a single interchange envelope, group segments, and transaction sets
- Group documents—Contain a single group segment and transaction sets
- Transaction documents—Contain a single transaction set

To perform processing on the transactions in an inbound document, set *splitOption* to Transaction or Group. To use Trading Networks to route the inbound EDI document without processing individual transactions, set the *splitOption* to Interchange.

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

Value	Description
Interchange	<p>Module for EDI creates only the Interchange document.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: When <i>splitOption</i> is Interchange and the <i>FARconciliation</i> EDITPA variable is set to true, the module splits the document at the Group level. For more information, see “FARconciliation Variable” on page 68.</p> </div> <p>Non-standard When you are using non-standard processing and you specify Interchange, the module splits the document at the Group level. For more information, see “wm.b2b.edi.TPA:EDITPA IS Document Type When Using Non-Standard Processing” on page 334.</p>

Value	Description
Group	Module for EDI creates the Interchange document and a Group document for each group segment in the interchange segment.
Transaction	Module for EDI 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.

FAReconciliation Variable

FAReconciliation EDITPA Variable

Default: false

The *FAReconciliation* variable indicates whether to enable FA reconciliation so Module for EDI performs both of the following:

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

To reconcile FAs, Module for EDI makes a record of each Group/Interchange EDI document that it sends and receives through Trading Networks. For ANSI EDI documents, the module records each Group document it sends or receives. For UN/EDIFACT EDI documents, the module records each Group document if the EDI document contains Group-level documents; if it does not, the module records Interchange-level document. Module for EDI records the information about these documents to the EDITTRACKING table, which is a Module for EDI-specific table in the Trading Networks database. For more information about the EDITTRACKING table, see [“Routing the Outbound EDI Document to Trading Networks” on page 270](#).

Note: When 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 EDITTRACKING table.

When FA reconciliation is enabled, Module for EDI updates the status for each Group/Interchange document in the EDITTRACKING table when it sends or receives the Group/Interchange document's corresponding FA. For more information about FA reconciliation, see [“Reconciling Functional Acknowledgments” on page 303](#).

Value	Description
true	Module for EDI enables FA reconciliation. <ul style="list-style-type: none"> ■ When the module sends or receives a Group/Interchange document through Trading Networks, it adds an entry to the EDITTRACKING table the document and sets the FA status to None.

Value	Description
	<ul style="list-style-type: none"> ■ When the module receives or sends the corresponding FA, it attempts to locate the corresponding Group/Interchange document in the EDITRACKING table. ■ When the module locates a matching entry, it updates the FA status. For example, if the module locates one matching document and the FA has the "A" (Accept) status, the module updates the FA status to Accept. For a list of all possible statuses, see “FA Statuses” on page 306. ■ When the module cannot find a matching entry, it adds an entry for the FA to the EDITRACKING table.
false	<p>Module for EDI disables FA reconciliation.</p> <ul style="list-style-type: none"> ■ When the 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 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"> ■ When the module locates a matching entry and the status is Disable, it does nothing. ■ When the module locates a matching entry and the status is None, it updates the status to Disable. ■ When the module cannot find a matching entry, it adds an entry for the FA to the EDITRACKING table.

UNAmode Variable

UNAmode EDITPA Variable

Default: auto

The *UNAmode* variable indicates whether you want to create a UNA segment to precede the interchange in an outbound UN/EDIFACT EDI document.

When you use the Module for EDI batching feature to create an outbound EDI document, the *batchProcess* service honors this setting. For more information, see [“Batching EDI Documents” on page 275](#).

Additionally, services you create to form outbound UN/EDIFACT 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* for your release.

Value	Description
yes	The module creates the UNA segment prior to the interchange in the outbound EDI UN/EDIFACT document.
no	The module does not create UNA segment prior to the interchange in the outbound EDI UN/EDIFACT document.
auto	The module creates the UNA segment prior to the interchange in the outbound EDI UN/EDIFACT document.

publishBatchFailEvent Variable

publishBatchFailEvent EDITPA Variable

Default: false

The *publishBatchFailEvent* variable indicates whether you want Module for EDI 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 `wm.b2b.editm.rec:batchFailRecord` IS document type. For more information, see *webMethods Module for EDI 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 for Failed Tasks” on page 283](#).

Value	Description
true	The module publishes an IS document when it is unable to include into the final batch EDI document, an EDI document that is queued for batching.
false	The module does not publish an IS document when it is unable to include into the final batch EDI document, an EDI document that is queued for batching.

delimiters Variables

delimiters/record EDITPA Variable

Default: (no default)

The *delimiters/record* 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 `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-In Services Reference* for your release.

When you use the Module for EDI batching feature to create an outbound EDI document, if you do not specify the *delimiters* input parameter 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 282](#).

delimiters/field EDITPA Variable

Default: (no default)

The *delimiters/field* 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 `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-In Services Reference* for your release.

When you use the Module for EDI batching feature to create an outbound EDI document, if you do not specify the *delimiters* input parameter 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 282](#).

***delimiters/subfield* EDITPA Variable**

Default: (no default)

The *delimiters/subfield* 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 `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-In Services Reference* for your release.

When you use the Module for EDI batching feature to create an outbound EDI document, if you do not specify the *delimiters* input parameter 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 282](#).

***delimiters/release* EDITPA Variable**

Default: (no default)

The *delimiters/release* variable indicates the release character to use for an outbound EDI document (for example, “\ \xd3). 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* for your release.

When you use the Module for EDI batching feature to create an outbound EDI document, if you do not specify the *delimiters* input parameter 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 282](#).

envelopeIdentifier Variables

The *envelopeIdentifier* variables consist of ID and qualifier variables for senders and receivers.

***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* for your release.

When using the Module for EDI batching feature:

- When the `batchProcess` service *oneBatchQueue* input parameter is NONE, the `batchProcess` service uses its input parameters 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 the *webMethods Trading Networks Built-In Services Reference* for your release

- When you are using batching with *oneBatchQueue* set to SINGLEOUTPUT or MULTIPLEOUTPUTS, do not specify the *envelopeIdentifier/sender/ID* variable.

***envelopeIdentifier/sender/qualifier* 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* for your release.

When using the Module for EDI batching feature:

- When the *batchProcess* service *oneBatchQueue* input parameter is NONE, the *batchProcess* service uses its input parameters 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 the *webMethods Trading Networks Built-In Services Reference* for your release.
- When you are using batching with *oneBatchQueue* set to SINGLEOUTPUT or MULTIPLEOUTPUTS, do not specify the *envelopeIdentifier/sender/ID* variable.

***envelopeIdentifier/receiver/ID* 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* for your release.

When using the Module for EDI batching feature:

- When the *batchProcess* service *oneBatchQueue* input parameter is NONE, the *batchProcess* service uses its input parameters 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 the *webMethods Trading Networks Built-In Services Reference* for your release.
- When you are using batching with *oneBatchQueue* set to SINGLEOUTPUT or MULTIPLEOUTPUTS, do not specify the *envelopeIdentifier/sender/ID* variable.

***envelopeIdentifier/receiver/qualifier* EDITPA Variable**

Default: (no default)

The *envelopeIdentifier/receiver/qualifier* variable indicates the EDI ID qualifier associated with the value you specify for *envelopeIdentifier/receiver/ID*. For example, if you specified a D-U-N-S number for the *envelopeIdentifier/receiver/ID*, specify 01 for *envelopeIdentifier/receiver/qualifier*, 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* for your release.

When using the Module for EDI batching feature:

- When the `batchProcess` service `oneBatchQueue` input parameter is `NONE`, the `batchProcess` service uses its input parameters 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 the *webMethods Trading Networks Built-In Services Reference* for your release.
- When you are using batching with `oneBatchQueue` set to `SINGLEOUTPUT` or `MULTIPLEOUTPUTS`, do not specify the `envelopeIdentifier/sender/ID` variable.

ICheaderInfo Variables

The *ICheaderInfo* variables indicate the values to use for ANSI X12 interchange and group headers and UN/EDIFACT UNB headers in outbound EDI documents.

ICheaderInfo/ISA/ISA01 EDITPA Variable

Default: 00

The *ICheaderInfo/ISA/ISA01* 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 `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-In Services Reference* for your release.

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

ICheaderInfo/ISA/ISA02 EDITPA Variable

Default: (no default)

The *ICheaderInfo/ISA/ISA02* 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 `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-In Services Reference* for your release.

When using the Module for EDI batching feature, the 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* for your release.

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

ICheaderInfo/ISA/ISA04 EDITPA Variable

Default: (no default)

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* for your release.

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

***IHeaderInfo/ISA/ISA11* EDITPA Variable**

Default: U

The *IHeaderInfo/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* for your release.

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

***IHeaderInfo/GS/GS07* EDITPA Variable**

Default: (no default)

The *IHeaderInfo/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* for your release.

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

***IHeaderInfo/UNB/UNB01* EDITPA Variable**

Default: (no default)

The *IHeaderInfo/UNB/UNB01* 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 `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-In Services Reference* for your release.

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

***IHeaderInfo/UNB/UNB06* EDITPA Variable**

Default: (no default)

The *IHeaderInfo/UNB/UNB06* 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 `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-In Services Reference* for your release.

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

***IHeaderInfo/UNB/UNB08* EDITPA Variable**

Default: (no default)

The *ICheaderInfo/UNB/UNB08* 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 `wm.tn.tpa:getTPA` service. For more information about this service, see the *webMethods Trading Networks Built-In Services Reference* for your release.

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

FAGeneration Variables

FAGeneration/autoGenerateFA EDITPA Variable

Default: Off

Note: Non-standard When you are using non-standard processing, Module for EDI does not use this EDITPA variable. Instead, it uses the **Auto Generate FA** setting available on the Interchange Information Detail panel of the module's home page. For more information about accessing and using this panel, see ["Defining Interchange-Level Sender/Receiver Pair Information" on page 335](#).

The *FAGeneration/autoGenerateFA* variable indicates whether you want Module for EDI to automatically generate FAs for an inbound EDI document.

Value	Description
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 (for example, ISA14 or UNB09).

FAGeneration/FALevel EDITPA Variable

Default: Default

Note: Non-standard When you are using non-standard processing, Module for EDI does not use this EDITPA variable. Instead, it uses the **FA Level** setting available on the Interchange Information Detail panel of the module's home page. For more information about accessing and using this panel, see ["Defining Interchange-Level Sender/Receiver Pair Information" on page 335](#).

The *FAGeneration/FALevel* variable defines the level of detail that the module acknowledges in the FAs that it generates.

Value	Description
Default	Acknowledges at the envelope level (group for ANSI X12 documents and interchange for UN/EDIFACT documents).
TransactionSet	Acknowledges at the transaction set level.
Segment	Acknowledges at the segment level.

Value	Description
Element	Acknowledges at the element level.

Merk:

When 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 156.

FAGeneration/processDocument EDITPA Variable

Default: All

Note: **Non-standard** When you are using non-standard processing, Module for EDI does not use this EDITPA variable. Instead, it uses the **Child Transaction Rejected Status** setting that you set from the Interchange Information Detail panel of the module’s home page. For more information about accessing and using this panel, see [“Defining Interchange-Level Sender/Receiver Pair Information”](#) on page 335.

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

- For acceptable FA statuses, the module processes a transaction, group, or UN/EDIFACT interchange using its normal processing.
- For unacceptable FA statuses, the module processes documents differently.

For more information about how Module for EDI processes documents based on FA status, see [“Defining Actions Module for EDI Takes Based on FA Status”](#) on page 249.

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 all Interchange, Group, and Transaction documents regardless of their FA status.	All	None
Only Accepted	Process only Interchange, Group, and Transaction documents that have the FA status "Accepted".	Accepted	<ul style="list-style-type: none"> ■ Not Allowed ■ Rejected ■ Partially Accepted ■ Accepted, But Errors Were Noted

<i>Set process</i>	When you want to:	Acceptable FA statuses	Unacceptable FA statuses
<i>Document to:</i>			
Not Rejected	Process all Interchange, Group, and Transaction documents unless they have the FA status "Rejected".	<ul style="list-style-type: none"> ■ Not Allowed ■ Partially Accepted ■ Accepted, But Errors Were Noted ■ Accepted 	Rejected

FAGeneration/generateControlNumber EDITPA Variable

Default: FromInboundDocument

Note: Non-standard When you are using non-standard processing, Module for EDI does not use this EDITPA variable. Instead, it uses the **Generate Control Number** setting that you set from the Interchange Information Detail panel of the module's home page. For more information about accessing and using this panel, see ["Defining Interchange-Level Sender/Receiver Pair Information"](#) on page 335.

The *FAGeneration/generateControlNumber* variable defines how Module for EDI is to generate the control numbers that it uses in the interchange and group headers of the FAs that it automatically generates.

Value	Description
FromInboundDocument	<p>Module for EDI uses the control numbers from the corresponding headers of the inbound EDI document that the FA acknowledges.</p> <p>The module uses the control number from the corresponding header of the inbound EDI document that the FA acknowledges. For example, if acknowledging a group, the module uses the control number from the corresponding group header in the inbound document.</p>
Random	Module for EDI randomly generates control numbers for the interchange and group headers of the FA.
FromControlNumberTable	<p>Module for EDI obtains the control numbers from the EDIControlNumber table.</p> <p>The module searches the EDIControlNumber table for the entry that matches the sender/receiver, EDI standard/version, production mode, and type (for example, "Envelope" or group type) identified in the corresponding interchange or group header of the inbound EDI document that the FA acknowledges. The module uses the next control number from this EDIControlNumber table entry for the FA. It also</p>

Value	Description
	increments the value of the next control number in the table entry, so it reflects the new next control number.

Note: Module for EDI always sets the control number for the transaction (997 or CONTRL) to 001.

FAGeneration/addGroup EDITPA Variable

Default: false

The *FAGeneration/addGroup* variable indicates whether you want Module for EDI to add a functional group to the UN/EDIFACT FA (for example, CONTRL).

Value	Description
true	Adds a functional group to the FA.
false	Does not add a functional group to the FA. This is the default.

FAGeneration/ctlNumberWleadingZero EDITPA Variable

Default: false

The *FAGeneration/ctlNumberWleadingZero* variable indicates whether you want Module for EDI to fill the control number field value with leading zeros. For example, if the control number is 34, and the length of the field is eight, if the *FAGeneration/ctlNumberWleadingZero* variable is set to true, the module will set the control number field value to 00000034.

Value	Description
true	Adds leading zeros to the control number field value.
false	Does not add zeros to the control number field value.

FAGeneration/RejectionRules/syntaxErrorStatus EDITPA Variable

Default: Rejected

Note: Non-standard When you are using non-standard processing, Module for EDI does not use this EDITPA variable. Instead, it uses the **Syntax Error Status** setting that you set from the Interchange Information Detail panel of the module's home page. For more information about accessing and using this panel, see ["Defining Interchange-Level Sender/Receiver Pair Information" on page 335](#).

The *FAGeneration/RejectionRules/syntaxErrorStatus* variable indicates how you want Module for EDI to report the syntax error status for a transaction, group, or UN/EDIFACT interchange.

Value	Description
Rejected	Use this setting when you want to reject the element (for example, transaction) because of syntax errors.

Value	Description
	<p>Reports the syntax error status as:</p> <ul style="list-style-type: none"> ■ “Accepted” when there are no syntax errors ■ “Rejected” when there are syntax errors
Accepted, But Errors Were Noted	<p>Use this setting when you want to know whether there were syntax errors, but you do not want to reject the element (for example, transaction) because of syntax errors.</p> <p>Reports the syntax error status as:</p> <ul style="list-style-type: none"> ■ “Accepted” when there are no syntax errors ■ “Accepted, But Errors Were Noted” when there are syntax errors
Accepted	<p>Use this setting when you do not want to check for syntax errors.</p> <p>Always reports the syntax error status as “Accepted” regardless of whether there are any syntax errors.</p>

FAGeneration/RejectionRules/logicalErrorStatus EDITPA Variable

Default: Rejected

Note: Non-standard When you are using non-standard processing, Module for EDI does not use this EDITPA variable. Instead, it uses the **Logical Error Status** setting that you set from the Interchange Information Detail panel of the module’s home page. For more information about accessing and using this panel, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#).

The *FAGeneration/RejectionRules/logicalErrorStatus* variable indicates how you want Module for EDI to report the logical error status.

Value	Description
Rejected	<p>Reports the logical error status as:</p> <ul style="list-style-type: none"> ■ “Accepted” when there are no logical errors ■ “Rejected” when there are logical errors <p>Use this setting when you want to reject the element (for example, transaction) because of logical errors.</p>
Accepted, But Errors Were Noted	<p>Reports the logical error status as:</p> <ul style="list-style-type: none"> ■ “Accepted” when there are no logical errors ■ “Accepted, But Errors Were Noted” when there are logical errors

Value	Description
	Use this setting when you want to know whether there were logical errors but do not want to reject the element (for example, transaction) because of logical errors.
Accepted	Always reports the logical error status as "Accepted" regardless of whether there are any logical errors. Use this setting when you do not want to check for logical errors.

FAGeneration/RejectionRules/childTransactionRejectedStatus EDITPA Variable

Default: Rejected

Note: Non-standard When you are using non-standard processing, Module for EDI does not use this EDITPA variable. Instead, it uses the **Child Transaction Rejected Status** setting that you set from the Interchange Information Detail panel of the module's home page. For more information about accessing and using this panel, see ["Defining Interchange-Level Sender/Receiver Pair Information"](#) on page 335.

The *FAGeneration/rejectionRules/childTransactionRejectedStatus* variable indicates how you want Module for EDI 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 ["childTransactionRejectedStatus Parameter"](#) on page 160.

Value	Description
Rejected	Reports the child transaction rejected status as: <ul style="list-style-type: none"> ■ "Rejected" when the FA status of any of the child transactions is either "Rejected" or "Accepted, But Errors Were Noted". ■ "Accepted" when the FA statuses of all the child transactions are "Accepted". <p>Use this setting when the FA status of all of the child transactions are "Rejected."</p>
Partially Accepted	Reports the child transaction rejected status as: <ul style="list-style-type: none"> ■ "Rejected" when the FA statuses of all of the child transactions are "Rejected". ■ "Partially Accepted" when 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".

Value	Description
	<ul style="list-style-type: none"> ■ “Accepted, But Errors Were Noted” when the FA statuses of the child transactions are either “Rejected” and/or “Accepted, But Errors Were Noted” AND no child transactions are “Accepted”. ■ “Accepted” when the FA statuses of all the child transactions are “Accepted”. <p>Use this setting when the FA status of at least one child transaction is “Accepted” and one is “Rejected”.</p>
Accepted, But Errors Were Noted	<p>Reports the child transaction rejected status as:</p> <ul style="list-style-type: none"> ■ “Rejected” when all the child transactions are “Rejected”. ■ “Accepted, But Errors Were Noted” when the FA statuses of the child transactions are “Rejected”, “Accepted, But Errors Were Noted”, and “Accepted”. ■ “Accepted” when the FA statuses of all the child transactions are “Accepted”. <p>Use this setting when the FA status of at least one child transaction is “Accepted, But Errors Were Noted”.</p>

***FAGeneration/MessageIdentifier/VersionNumber* EDITPA Variable**

Default: (no default)

The *FAGeneration/MessageIdentifier/VersionNumber* variable specifies the version number that the *wm.b2b.edi.util:generateFA* service should place in the CONTRL message when automatically generating FAs.

***FAGeneration/MessageIdentifier/ReleaseNumber* EDITPA Variable**

Default: (no default)

The *FAGeneration/MessageIdentifier/ReleaseNumber* variable specifies the release number that the *wm.b2b.edi.util:generateFA* service should place in the CONTRL message when automatically generating FAs.

***FAGeneration/MessageIdentifier/ControllingAgency* EDITPA Variable**

Default: (no default)

The *FAGeneration/MessageIdentifier/ControllingAgency* variable specifies the name of the controlling agency that the *wm.b2b.edi.util:generateFA* service should place in the CONTRL message when automatically generating FAs.

***FAGeneration/MessageIdentifier/AssociationCode* EDITPA Variable**

Default: (no default)

The *FAGeneration/MessageIdentifier/AssociationCode* variable specifies the association code that the *wm.b2b.edi.util:generateFA* service should place in the CONTRL message when automatically generating FAs.

***FAGeneration/MessageIdentifier/CodeListDirectoryVersionNumber* EDITPA Variable**

Default: (no default)

The *FAGeneration/MessageIdentifier/CodeListDirectoryVersionNumber* variable specifies the version number of the code list directory that the `wm.b2b.edi.util.generateFA` service should place in the CONTRL message when automatically generating FAs.

***FAGeneration/MessageIdentifier/MessageTypeSubFunctionIdentification* EDITPA Variable**

Default: (no default)

The *FAGeneration/MessageIdentifier/MessageTypeSubFunctionIdentification* variable specifies the code that identifies the sub function of a message type, which the `wm.b2b.edi.util.generateFA` service should place in the CONTRL message when automatically generating FAs.

***FAGeneration/DocumentSettings* EDITPA Variable**

Default: (no default)

The *FAGeneration/DocumentSettings* variables specify the auto FA generation settings for different group types (rather than for a sender/receiver pair). Click the **Add Row** icon to add the FA generation settings for a group type.

The *FAGeneration/DocumentSettings/groupType* variable is used to specify the group type. The remaining variables in *FAGeneration/DocumentSettings* function exactly the same as their corresponding *FAGeneration* variables, except that they pertain to a particular group type. For more information about setting the FA generation variables for each group type, see [“FAGeneration Variables” on page 75](#).

Module for EDI processes inbound messages as follows:

- When a document is received, the module identifies the TPA associated with the document and examines the *FAGeneration/DocumentSettings/groupType* variable to determine the FA generation setting for the group type of the inbound message.
- When the group type of the inbound message is listed and *autoGenerateFA* for that group type is set to On, the module generates the FA for that particular group type as per the settings.
- When the group type of the inbound message is not listed, the module examines the *FAGeneration/autoGenerateFA* variable to determine whether an FA should be generated as follows:
 - When the *FAGeneration/DocumentSettings/autoGenerateFA* variable is set to On, the module automatically generates an FA for the inbound message.
 - When the *FAGeneration/DocumentSettings/autoGenerateFA* variable is set to Per Document, the module automatically generates an FA based on the indicator flag in the interchange header (for example, ISA14 or UNB09).
 - When the *FAGeneration/DocumentSettings/autoGenerateFA* variable is set to Off, the module does not generate an FA for the inbound message.

Note:

When more than one record exists for a group type, Module for EDI considers the first record it encounters for the group when generating the FA.

Important:

Do not update and delete elements in the *DocumentSettings* array at the same time. Because no single identifier exists for these entries, there is no way for Module for EDI to identify which element was deleted and which one was modified. As a result, the message that the module logs to the Activity Log will not correctly reflect all of the changes that were performed. You should first delete the element to be deleted, save the TPA, and then modify the other elements as necessary.

***FAGeneration/userFFSchemaForEnvelope* EDITPA Variable**

Default: (no default)

The *FAGeneration/userFFSchemaForEnvelope* variable indicates the schema to be used while generating FA for an inbound message. You can customize the schema to be used for validating the envelope. When you copy the *wm.b2b.edi.EDIFFSchema:ICS4UNEDIFACTDEF_large* schema and modify the copied schema to customize the schema, there are some limitations. For more information about the limitations, see the *wm.b2b.edi.util:generateFA* service description in *webMethods Module for EDI Built-In Services Reference*.

ControlNumberManagement Variables

***ControlNumberManagement/validateInboundEnvelopeControlNumbers* EDITPA Variable**

Default: false

Note: Non-standard When you are using non-standard processing, Module for EDI does not use this EDITPA variable. Instead, it uses the **Validate inbound envelope control numbers** setting that you set from the Interchange Information Detail panel of the module's home page. For more information about accessing and using this panel, see "[Defining Interchange-Level Sender/Receiver Pair Information](#)" on page 335.

The *ControlNumberManagement/validateInboundEnvelopeControlNumbers* variable indicates whether you want the 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 189.

Value	Description
true	Module for EDI validates and tracks control numbers in the interchange headers of inbound EDI documents.
false	Module for EDI does not 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 Module for EDI 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 189.

Value	Description
true	Module for EDI validates and tracks control numbers in the group headers of inbound EDI documents.
false	Module for EDI does not validate or track control numbers in the group headers of inbound EDI documents.

ControlNumberManagement/duplicateControlNumberAction EDITPA Variable

Default: Error & Continue

Note: 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 panel of the Module for EDI home page. For more information about accessing and using this panel, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#).

The *ControlNumberManagement/duplicateControlNumberAction* variable indicates the action you want Module for EDI 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 189](#). For more information about each of the following actions, see [“Actions Module for EDI Can Take for Invalid Control Numbers” on page 234](#).

Value	Description
Error & Continue	Module for EDI logs the error and then continues to process the EDI document that contains the invalid control number normally.
ProcessNormally	Module for EDI logs a warning and processes the EDI document that contains the invalid control number normally.
Reject	Module for EDI logs the error and does not process the document normally. The module does not split the EDI document. Typically, the 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 module sends the document without splitting it to Trading Networks processing rules.

Additionally, Module for EDI 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 236](#).

Value	Description
	You can force processing of the duplicate document later, if you want. For more information, see “Reprocessing EDI Documents with Invalid Control Numbers” on page 238.

***ControlNumberManagement/outOfSequenceControlNumberAction* EDITPA Variable**
Default: Error & Continue

Note: **Non-standard** When you are using non-standard processing, this EDITPA is only used for duplicate 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 panel of the Module for EDI home page. For more information about accessing and using this panel, see [“Defining Interchange-Level Sender/Receiver Pair Information”](#) on page 335.

The *ControlNumberManagement/outOfSequenceControlNumberAction* variable indicates the action you want Module for EDI to take when it encounters an out-of-sequence 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 189. For more information about each of the following actions, see [“Actions Module for EDI Can Take for Invalid Control Numbers”](#) on page 234.

Value	Description
Error & Continue	Module for EDI logs the error, and then continues to process the EDI document that contains the invalid control number normally.
ProcessNormally	Module for EDI logs a warning and processes the EDI document that contains the invalid control number normally.
Reject	<p>Module for EDI logs the error and does not process the document normally. The module does not split the EDI document. Typically, the 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, Module for EDI sends the document without splitting it to Trading Networks processing rules.</p> <p>Additionally, the module sets the Trading Networks custom attribute EDI Status to Duplicate Control Number. You can use the custom attribute EDI Status in the 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 236.</p> <p>You can force processing of the duplicate document later, if you want. For more information, see “Reprocessing EDI Documents with Invalid Control Numbers” on page 238.</p>

ControlNumberManagement/VDA/validateStructure EDITPA Variable

Default: false

The *ControlNumberManagement/VDA/validateStructure* variable indicates whether you want Module for EDI to validate the structure of inbound VDA documents. For more information, see [“Defining How to Handle Invalid Control Numbers for VDA Documents” on page 126.](#)

Value	Description
true	Module for EDI validates the structure of inbound VDA documents.
false	Module for EDI does not validate the structure of inbound VDA documents.

useReverseRoute Variable***useReverseRoute* EDITPA Variable**

Default: false

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

Value	Description
false	Module for EDI 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	Module for EDI 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.
useInternalIds	Module for EDI uses the internal identification and internal sub-identification portions of the UN/EDIFACT Sender/Receiver ID to determine the partner ID of the document.

The qualifier is the normal Identification code qualifier, but the ID is constructed as follows:

<Internal identification>[:<internal sub-identification>]

Note that when no sub-identification value is present, this setting behaves the same way as the true setting.

Note:

Once your production system is running, do not change the value of the *useReverseRoute* variable. For example, if you send a document with *useReverseRoute* set to false, Module for EDI considers your partner ID to be the values of *ID* and *Qualifier*, and it stores these values in the EDITRACKING table. When you change *useReverseRoute* to true and then receive an FA from the receiver, the module tries to use the *ID* and *ReverseRoute Id* values to try to

determine the sender of the FA you received. In this case, the module cannot reconcile the FA with the document you sent.

BatchCriteria Variables

Batching involves grouping together transactions from all queued EDI documents. You can use the *BatchCriteria* variables to specify which fields Module for EDI uses to identify the collections and sub-collections by which to batch transactions.

BatchCriteria/useDefault EDITPA Variable

Default: true

The *BatchCriteria/useDefault* variable indicates whether to consider only the standard fields for sorting the documents in the batching process.

When this variable is not present in the TPA, Module for EDI considers only the standard fields for sorting the documents in the batching process.

Value	Description
true	Module for EDI considers only the standard fields for sorting the documents in the batching process.
false	Module for EDI considers the standard fields and all the other fields set to true in UNB, UNG, ISA, and GS records for sorting the documents in the batching process.

BatchCriteria EDITPA Variable

Default: (no default)

The *BatchCriteria* variable indicates which fields should be considered during the sorting of documents in the batch process.

When the *BatchCriteria/useDefault* variable is set to true, Module for EDI considers only the standard fields for sorting the documents in the batching process.

The module considers the following standard fields for sorting the EDIFACT documents to a collection area.

Value	Description
UNB/UNB02/S00202	The sender qualifier of the EDIFACT document.
UNB/UNB02/S00201	The sender ID of the EDIFACT document.
UNB/UNB03/S00302	The receiver qualifier of the EDIFACT document.
UNB/UNB03/S00301	The receiver ID of the EDIFACT document.
UNB/UNB11	The mode of the EDIFACT document.
UNG/UNG07/S00802	The version of the EDIFACT document.

The module considers the following standard fields for sorting the X12 documents to a collection area.

Value	Description
ISA/ISA05	The sender qualifier of the X12 document.
ISA/ISA06	The sender ID of the X12 document.
ISA/ISA07	The receiver qualifier of the X12 document.
ISA/ISA08	The receiver ID of the X12 document.
ISA/ISA15	The mode of the X12 document.
ISA/ISA12	The version of the X12 document.

The module considers the following standard fields for sorting the EDIFACT documents to a sub-collection.

Value	Description
UNG/UNG02/S00601	The group sender of the EDIFACT document.
UNG/UNG03/S00701	The group receiver of the EDIFACT document.
UNG/UNG06	The control agency code of the EDIFACT document.
UNG/UNG07/S00801	The message version number of the EDIFACT document.
UNG/UNG07/S00802	The message release number of the EDIFACT document.
UNG/UNG07/S00803	The message association assigned number code of the EDIFACT document.
UNG/UNG01	The group type of the EDIFACT document.

The module considers the following standard fields for sorting the X12 documents to a sub-collection.

Value	Description
GS/GS02	The group sender of the X12 document.
GS/GS03	The group receiver of the X12 document.
GS/GS08	The version of the X12 document.
GS/GS01	The group type of the X12 document.

- When the *BatchCriteria/useDefault* variable is set to false, in addition to using the standard fields, Module for EDI considers all the other fields set to true in UNB, UNG, ISA and GS records for sorting the documents. For example, if you want to use the routing address UNB/UNB03/S00303 in addition to the standard fields in the sorting criteria, set *BatchCriteria/UNB/UNB03/S00303* to true.
- *BatchCriteria/UNB* record fields identify collection areas for EDIFACT documents

- *BatchCriteria/ISA* record fields identify collection areas for X12 documents
- *BatchCriteria/UNG* record fields identify sub-collection areas for EDIFACT documents
- *BatchCriteria/GS* record fields identify sub-collection areas for X12 documents

Note:

Do not include S0010, the syntax identifier field, in the sorting process.

EDITransportMedia Variable***EDITransportMedia* EDITPA Variable**

Default: FTP

The *EDITransportMedia* variable indicates the protocol (for example, HTTP, HTTPS, FTP, AS1, and AS2) to be used to transport the EDI documents. Module for EDI uses this variable to display information on the Summary of EDI Activity by Transport Media and EDI Volume by Transport Media reports, if you have purchased the appropriate EDI reporting license.

Note:

Beginning with Integration Server 9.12, the *EDITransportMedia* variable is obsolete.

persistOriginalEnvelope Variable***persistOriginalEnvelope* EDITPA Variable**

Default: false

The *persistOriginalEnvelope* variable indicates whether to persist the original incoming EDI envelope in Trading Networks before Module for EDI starts processing the EDI envelope.

When the module receives an EDI document from a partner, it processes the transactions first, the groups next, and the envelope last. While processing a large document, if the module fails to process the entire inbound document, it cannot retrieve the original EDI document received from the partner because the envelope is processed last. Module for EDI cannot track these unprocessed inbound documents because the original EDI document received from the partner is not persisted in Trading Networks by default.

When you choose to persist the original incoming EDI document in Trading Networks before Module for EDI starts processing the document, even if the module fails to process the entire document, you can track the original document in Trading Networks and reprocess it in Module for EDI.

If persisting each and every EDI document in Trading Networks before Module for EDI starts processing it slows down your system performance, you can configure the module to process the envelope first by setting the *documentPersistOrder* variable to *EnvelopeFirst*. In this case, the module does not persist the original incoming EDI document in Trading Networks. If Module for EDI fails to process the entire inbound document, you can still retrieve the details of the original EDI document received from the partner because the module processes the envelope first and submits it to Trading Networks. For more information, see “[documentPersistOrder Variable](#)” on page 90.

Value	Description
true	<p>Before processing the document, Module for EDI persists the original incoming EDI document in Trading Networks with the following attribute values:</p> <pre>Processing Status = NEW User Status = Original EDI Envelope EDI Status = Persisted</pre> <p>When the module does not completely process the inbound document, the module will not delete the original document persisted in Trading Networks. Documents containing the attribute values mentioned above identify that the corresponding document was not processed completely. Resubmit these original documents to Trading Networks for reprocessing.</p> <p>When the module successfully processes the inbound document, that is, after all the transactions, groups, and envelope of the document are processed, the additional document persisted in Trading Networks is deleted.</p> <p>Note: When the value of the <i>documentPersistOrder</i> EDITPA variable is <i>EnvelopeFirst</i>, Module for EDI does not persist the original EDI envelope in Trading Networks before processing the document even if you set the value of the <i>persistOriginalEnvelope</i> variable to true.</p>
false	Module for EDI does not persist the original inbound EDI envelope in Trading Networks before processing the document. This is the default.

documentPersistOrder Variable

documentPersistOrder EDITPA Variable

Default: *EnvelopeLast*

The *documentPersistOrder* variable indicates the order in which the EDI documents are submitted to Trading Networks.

Value	Description
<i>EnvelopeLast</i>	<p>Module for EDI submits the envelope of the EDI document to Trading Networks after submitting the transactions and groups. This is the default.</p> <p>Module for EDI submits the EDI documents to Trading Networks in the following order:</p> <ol style="list-style-type: none"> 1. Transactions

Value	Description
	<ol style="list-style-type: none"> 2. Groups 3. Envelope
	<p>When the <i>persistOriginalEnvelope</i> variable is set to false, if the module fails while submitting large EDI documents, you cannot retrieve the original EDI document received from the partner because the module submits the envelope last. For more information, see “persistOriginalEnvelope Variable” on page 89.</p>
EnvelopeFirst	<p>Module for EDI submits the envelope of the EDI document to Trading Networks before submitting the groups and transactions.</p> <p>The module submits the EDI documents to Trading Networks in the following order:</p> <ol style="list-style-type: none"> 1. Envelope 2. Groups 3. Transactions <p>Even though the <i>persistOriginalEnvelope</i> variable is set to true, the module does not persist the original inbound EDI document in Trading Networks before processing the document. Because the module submits the envelope first, you can track the original inbound EDI document at any time until processing is complete.</p>

LateFA Variables

Use the *LateFA* variables for tracking the late functional acknowledgments for a partner pair. For more information, see [“Tracking Late Functional Acknowledgments” on page 167](#).

LateFA/LateFATime EDITPA Variable

Default: (no default)

The *LateFA/LateFATime* variable indicates the time in minutes by which Trading Networks must receive the FA for a partner pair. When Trading Networks receives the FA late, Module for EDI sets the **Late FA** attribute of the original document to Y in Trading Networks.

For a partner pair, Module for EDI checks the Trading Networks database to find the time difference between the time the document was sent and the time the sender receives the FA. When the time difference is more than the value of the *LateFA/LateFATime* variable, the module sets the **Late FA** attribute of the original document to Y in Trading Networks. When the FA is not late, the **Late FA** attribute of the original document is set to N.

- When the value of the *LateFA/LateFATime* variable is 0 or a negative integer, the module does not consider any FA as late for the partner pair.
- When the value of the *LateFA/LateFATime* variable is a positive integer, if Trading Networks receives the FA later than the value of the *LateFA/LateFATime* variable, the module sets the

Late FA attribute of the original document in Trading Networks to Y. If Trading Networks receives the FA within the time specified by the *LateFA/LateFATime* variable, the module sets the **Late FA** attribute to N.

- When the value of the *LateFA/LateFATime* variable is not set, the Late FA attributes of the original documents are not set.

***LateFA/LateFAForX12Group* EDITPA Variable**

Default: (no default)

The *LateFA/LateFAForX12Group* variable defines the late FA setting for the X12 groups specified in the *LateFA/LateFAForX12Group/groupType* variables.

Note:Module for EDI does not support tracking of late FAs for UN/EDIFACT groups.

***LateFA/LateFAForX12Group/groupType* EDITPA Variable**

Default: (no default)

The *LateFA/LateFAForX12Group/groupType* variables indicate the X12 groups used for tracking the late FAs.

When the original document does not contain any of the groups specified in *LateFA/LateFAForX12Group/groupType* variables, Module for EDI tracks the late FAs at the partner pair level by considering the time indicated in the *LateFA/LateFATime* variable.

***LateFA/LateFAForX12Group/LateFATime* EDITPA Variable**

Default: (no default)

The *LateFA/LateFAForX12Group/LateFATime* variable indicates the time in minutes by which Trading Networks must receive the FA for the specified group. When Trading Networks receives the FA late for the group, Module for EDI sets the **Late FA** attribute to Y for all the transactions in that group in Trading Networks.

For a group indicated in the *LateFA/LateFAForX12Group/groupType* variable of a partner pair, the module checks the Trading Networks database to find the time difference between the time the Group document was sent and the time the sender receives the FA. When the time difference is more than the value of the *LateFA/LateFAForX12Group/LateFATime* variable for the specified group, the module sets the **Late FA** attribute to Y for all the transactions of the specified group in Trading Networks. When the FA for the group is not late, the **Late FA** attribute is set to N for all the transactions in that group.

After Module for EDI sets the **Late FA** attribute for a document for that group, the module does not consider the duplicate FAs that Trading Networks may receive later.

- When the value of the *LateFA/LateFAForX12Group/LateFATime* variable is 0 or less, the module does not consider any FA as late for the group specified by the *LateFA/LateFAForX12Group/groupType* variable.
- When the value of the *LateFA/LateFAForX12Group/LateFATime* variable is a positive integer, the module sets the **Late FA** attribute of the original document to Y if Trading Networks receives the FA late for the group specified by the *LateFA/LateFAForX12Group/groupType* variable.
- When the value of the *LateFA/LateFAForX12Group/LateFATime* variable is not set, the Late FA attributes of the specified Group documents are not set.

X12TA1Generation Variables

The X12TA1Generation variables indicate the settings for interchange acknowledgment (TA1).

X12TA1Generation/autoGenerateTA1 EDITPA Variable

Default: Off

The *X12TA1Generation/autoGenerateTA1* variable indicates whether you want the Module for EDI to automatically generate TA1s for an inbound EDI document.

Value	Description
on	Always automatically generates TA1s for inbound EDI documents
off	Never automatically generates TA1s for inbound EDI documents

X12TA1Generation/Rules EDITPA Variable

The *X12TA1Generation/Rules* variable defines the rule settings for the TA1s the module generates.

X12TA1Generation/Rules[]/Name EDITPA Variable

Default: (no default)

The *X12TA1Generation/Rules[]/Name* variable specifies the name of the TA rule.

X12TA1Generation/Rules[]/ErrorCode EDITPA Variable

Default: (no default)

The *X12TA1Generation/Rules[]/ErrorCode* variable indicates the error code displayed when rule validation fails.

X12TA1Generation/Rules[]/Description EDITPA Variable

Default: (no default)

The *X12TA1Generation/Rules[]/Description* variable describes the validation rule.

X12TA1Generation/Rules[]/Enabled EDITPA Variable

Default: yes

The *X12TA1Generation/Rules[]/Enabled* variable indicates whether the validation rule is applied while generating the TA1.

Value	Description
yes	The rule is applied while generating the TA1.
no	The rule is not applied while generating the TA1.

X12TA1Generation/Rules[]/Order EDITPA Variable

Default: (no default)

The *X12TA1Generation/Rules[]/Order* indicates the chronological order in which rules are validated during TA1 generation. The lower the value, the higher the priority. Once validation of one rule fails, the TA1 is generated with the error that corresponds to the failed rule.

X12TA1Generation/Rules[]/Type EDITPA Variable

Default: E

The *X12TA1Generation/Rules[]/Type* variable indicates the valid values that can be used to define the TA1 status.

Value	Description
A	The transmitted interchange envelope was received without errors.
R	The transmitted interchange envelope was rejected.
E	The transmitted interchange envelope was received and accepted, but errors were noted.

***X12TA1Generation/Rules[]/validatorClass* EDITPA Variable**

Default: (no default)

The *X12TA1Generation/Rules[]/validatorClass* variable indicates the fully qualified java class name that Module for EDI uses internally while validating rules.

EANCOMAUTACK Variables

The *EANCOMAUTACK* variables indicate the configurations necessary for generating a secure Authentication and Acknowledgement (AUTACK) message.

Use the following *EANCOMAUTACK* variables to configure the settings for generating or processing the AUTACK messages:

***EANCOMAUTACK/autoGenerateAUTACK* EDITPA Variable**

Default: off

The *EANCOMAUTACK/autoGenerateAUTACK* variable indicates whether the AUTACK message containing the digital signatures are generated automatically.

Value	Description
on	Always generates the AUTACK message automatically.
off	Never generates the AUTACK message automatically.

***EANCOMAUTACK/processInboundAUTACK* EDITPA Variable**

Default: off

The *EANCOMAUTACK/processInboundAUTACK* variable indicates whether the AUTACK message containing the digital signatures are processed automatically.

Value	Description
on	Always processes the inbound AUTACK message automatically.
off	Never processes the inbound AUTACK message automatically.

***EANCOMAUTACK/configuration* EDITPA Variable**

Default: (no default)

The *EANCOMAUTACK/configuration* variables define the configuration settings required to generate the AUTACK message.

***EANCOMAUTACK/configuration/signingScope* EDITPA Variable**

Default: Interchange

The *EANCOMAUTACK/configuration/signingScope* variable allows you to sign an entire Interchange or Envelope, an entire Group, or each Transaction individually.

Value	Description
Interchange	Signs data from the UNB through UNZ segments, excluding the UNA segment.
Group	Signs data from the UNG through UNE segments.
Transaction	Signs data from the UNH through UNT segments.

***EANCOMAUTACK/configuration/digestAlgorithm* EDITPA Variable**

Default: SHA1

The *EANCOMAUTACK/configuration/digestAlgorithm* variable indicates the type of hash algorithm used for generating hash before signing the AUTACK message.

Value	Description
SHA1	Generates 160-bit hash using the SHA1 algorithm.
MD5	Generates 128-bit hash using the MD5 algorithm.
RIPMD160	Generates 160-bit hash using the RIPMD160 algorithm.

***EANCOMAUTACK/configuration/paddingAlgorithm* EDITPA Variable**

Default: ISO9796-2

The *EANCOMAUTACK/configuration/paddingAlgorithm* variable indicates the type of key algorithm that checks the robustness of the digital signature generated by the AUTACK message.

Value	Description
ISO9796-2	Uses ISO9796-2 padding algorithm before signing the AUTACK message.

***EANCOMAUTACK/configuration/signingAlgorithm* EDITPA Variable**

Default: RSA

The *EANCOMAUTACK/configuration/signingAlgorithm* variable indicates the type of signing algorithm for generating the digital signature.

Value	Description
RSA	Generates the digital signatures using the RSA algorithm.

***EANCOMAUTACK/configuration/filteringAlgorithm* EDITPA Variable**

Default: Hexadecimal

The *EANCOMAUTACK/configuration/filteringAlgorithm* variable indicates the type of filtering algorithm that converts digital signatures from a binary value to a string.

Value	Description
Hexadecimal	Converts an 8-bit binary value into a hexadecimal string of two characters.

EANCOMAUTACK/configuration/characterSetEncoding EDITPA Variable

Default: ASCII

The *EANCOMAUTACK/configuration/characterSetEncoding* variable identifies the type of character set in which the EANCOM structure is encoded when security algorithms are applied to the generated the digital signature.

Value	Description
ASCII	American Standard Code for Information Interchange (7-bit ASCII standard).
ISO8859-1	ISO/IEC 8859 series of the 8-bit ASCII standard.
ISO8859-15	ISO/IEC 8859 series of the 8-bit ASCII standard.
Cp850	Encoding schema defined by the code page.
Cp500	Encoding schema defined by the EBCDIC code page.

Note:

Ensure that you perform the following configurations

- To generate or process the AUTACK message, configure the Sign/Verify certificate in the partner profile. Also, ensure that you install the EANCOM 1 AUTACK document type to generate the AUTACK message. For more information, see [“Defining TN EDI Document Types” on page 34](#).
- Set `EDIDisableCONTRLVersion = true` to enable the recognition of the AUTACK message in Trading Networks.
- If you use the EDI Batch attribute for an EANCOM document with `autoGenerateAUTACK = on`, then you should also set the extended criteria, EDI Outbound FA to `true` in the corresponding processing rule of the EDI Batch attribute. For more information, see [“Defining Processing Rules to Batch EDI Documents” on page 293](#).

4 Defining Trading Partner Information When Using TRADACOMS

- Overview 98
- Defining Trading Partner Profiles When Using TRADACOMS 99
- Defining EDI Trading Partner Agreements When Using TRADACOMS 101

Overview

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

For more information about how Module for EDI processes inbound documents, see *webMethods Module for EDI Concepts Guide*.

Partner Information You Need to Define When Using TRADACOMS

The following table lists the information you must define for sender/receiver pairs:

For...	Define...
Transmission sender/receiver pairs	Trading partner profiles for each transmission sender and receiver. For more information, see “Defining Trading Partner Profiles When Using TRADACOMS” on page 99 .
Tailoring how Module for EDI 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 When Using TRADACOMS” on page 101.</p>
How Module for EDI validates inbound control numbers	<ul style="list-style-type: none"> ■ Whether to validate control numbers. ■ Actions that Module for EDI takes when encountering an invalid control number. ■ Control number validation settings (that is, maximum, minimum, increment, and window). <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p>Note: The term <i>control number</i> is used in the non-TRADACOMS EDI standards to refer to a number in an EDI document header. This number is used to validate and order the documents exchanged between trading partners. A control number in a non-TRADACOMS document is equivalent to the transmission reference number in the STX and BAT segments of TRADACOMS documents.</p> </div>

For...	Define...
	<p>Regardless of whether your EDI standard includes control numbers or transmission reference numbers, you define them the same way in Trading Networks; the only difference is in the terminology. For simplicity, Trading Networks and Module for EDI use the term <i>control number</i> to mean either control number or transmission reference number.</p>
	<p>For more information, see “Defining Control Number Information for Trading Partners” on page 113.</p>

For more information about how Module for EDI works with Trading Networks, see *webMethods Module for EDI Concepts Guide*. For more information about profiles and TPAs and how Trading Networks uses them, see the *Trading Networks Concepts Guide* for your release.

Defining Trading Partner Profiles When Using TRADACOMS

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 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.

Module for EDI 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 When Using TRADACOMS”](#) on page 99.

You create profiles using My webMethods. For steps to create profiles, see the *webMethods Trading Networks Administrator’s Guide* for your release.

External ID Types When Using 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 Module for EDI 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 can 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, the following four possibilities can result, depending on which fields have values:

Code present?	Name present?	Result
No	No	Unknown trading partner
Yes	No	Uses Code to look up trading partner
No	Yes	Uses Name to look up trading partner
Yes	Yes	Uses 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 =ABC -and- Name field is empty	The external ID type Tradacoms Code defined as ABC.
Code field is empty -and- Name =ABC	The external ID type Tradacoms Name defined as ABC.
Code field=ABC -and- Name field=XYZ	Both of the following: <ul style="list-style-type: none"> ■ The external ID type Tradacoms Code defined as ABC and ■ The external ID type Tradacoms Name defined as XYZ

Note:
To use only **Tradacoms Code** to look up partners when both **Code** and **Name** have values, see [“Optionally Configuring the External ID Types When Using TRADACOMS”](#) on page 100.

Optionally Configuring the External ID Types When Using TRADACOMS

Perform the following procedure to configure the **Tradacoms Code** and **Tradacoms Name** external ID types. When you configure these external ID types, Module for EDI 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 when using TRADACOMS

1. In My webMethods: **Administration > Integration > B2B Settings > Configure Properties**.
2. Add the following properties as required:

Property	Description
<code>wm.edi.tradacoms.CodeExternalIDType=value</code>	Overrides the default Tradacoms Code external ID.
<code>wm.edi.tradacoms.NameExternalIDType=value</code>	Overrides the default Tradacoms Name external ID.
<code>wm.edi.tradacoms.CodeOnly=true</code>	When both the Code and Name fields have values, overrides Trading Networks from using them both to look up a trading partner. Instead, Trading Networks uses only Code to look up a trading partner.

Note:

If `CodeOnly=true` is set and the **Code** field is blank, Trading Networks looks up a trading partner as described in the table in [“External ID Types When Using TRADACOMS” on page 99](#).

3. Click **Save**.

Defining EDI Trading Partner Agreements When Using TRADACOMS

An EDI trading partner agreement (EDITPA) is a set of variables that you provide to tailor how Module for EDI exchanges documents between two trading partners. Module for EDI 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 Module for EDI uses only when processing documents that match the specified sender/receiver pair.
- A *default EDITPA* has a sender and receiver set to "unknown" and contains variables used by all trading partners when partner-specific information is not available.

During processing, Module for EDI 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, Module for EDI uses the value from the default EDITPA.

Modifying the Default EDITPA When Using TRADACOMS

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 Integration Server, Module for EDI 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:

Do not disable the default EDITPA. If you do not specify a value for a TPA parameter, Module for EDI uses the value from the default EDITPA.

You modify the default EDITPA using My webMethods. For instructions, see the *webMethods Trading Networks Administrator's Guide* for your release.

When you modify the default EDITPA, the following settings (visible on the Administration > Integration > B2B > Trading Partner Agreements page) must be maintained.

For this 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 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 “TRADACOM Variables” on page 103 .

Defining a Partner-Specific EDITPA When Using TRADACOMS

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, Module for EDI functions as if the partner-specific EDITPA does not exist. That is, Module for EDI uses the values in the default EDITPA.

You create a partner-specific EDITPA using My webMethods. 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, see the *webMethods Trading Networks Administrator's Guide* for your release.

When creating a partner-specific EDITPA, specify the following on the Administration > Integration > B2B > Trading Partner Agreements page:

For this 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	<p>wm.b2b.editn.TPA:EDITPA</p> <p>To set the values for the variables in wm.b2b.editn.TPA:EDITPA, see the description of the variables in “TRADACOM Variables” on page 103.</p>
Initialization Service	<p>A service that you created to define the default values for the wm.b2b.editn.TPA:EDITPA IS document type. (Or, leave this field blank.)</p> <p>It is recommended that you do not use the initialization service (wm.b2b.editn.TPA:initService) that is provided with Module for EDI. The wm.b2b.editn.TPA:initService service is used to populate the values of the default EDITPA.</p>

wm.b2b.editn.TPA:EDITPA IS Document Type When Using TRADACOMS

The descriptions and default values for the variables of the wm.b2b.editn.TPA:EDITPA IS document type are listed below. If no value is specified in the partner-specific EDITPA, nor the default EDITPA, Module for EDI uses a default value. The wm.b2b.editn.TPA:initService initialization service defines these default values in the default EDITPA when it is executed.

Use the variable descriptions to help you determine the appropriate value to set in the EDITPA. When creating partner-specific EDITPAs, remember to only specify values for the variables that you want to override in the default EDITPA. Leave the values blank in the variables for which the default value applies.

Tip:

It is helpful to have an understanding about how Module for EDI processes EDI documents to understand how Module for EDI uses the variables in the EDITPA. For more information, see *webMethods Module for EDI Concepts Guide*.

TRADACOM Variables

TRADACOMS/splitOption EDITPA Variable

Default: File

This variable indicates how you want Module for EDI to split a transmission segment within an EDI document. The 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 Module for EDI 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 111](#).

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

Value	Description
Transmission	Creates only the Transmission document.
Batch	Creates the Transmission document and a Batch document for each batch segment in the transmission segment.
File	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, My webMethods may not be able to display all of the content parts. In this case, you may want to consider using an alternative storage option.

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.
N Messages	Instead of storing all detail messages in a single File document type, Trading Networks creates a separate File document type for each detail

Value	Description
	message. For more information, see “Storage Options for File Document Types” on page 109.

Note:

When you use the Monitoring > Integration > B2B > Transaction page 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 Module for EDI to validate and track control numbers in the transmission headers of inbound EDI documents.

Value	Description
true	Validates and tracks control numbers in the transmission headers of inbound EDI documents.
false	Does not 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 Module for EDI to validate and track control numbers in the batch headers of inbound EDI documents.

Value	Description
true	Validates and tracks control numbers in the batch headers of inbound EDI documents.
false	Does not validate or track control numbers in the batch headers of inbound EDI documents.

TRADACOMS/ControlNumberManagement/validateInboundFileControlNumbers EDITPA Variable

Default: false

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

Value	Description
true	Validates and tracks control numbers in the file headers of inbound EDI documents.
false	Does not validate or track control numbers in the file headers of inbound EDI documents.

TRADACOMS/ControlNumberManagement/duplicateControlNumberAction EDITPA Variable
Default: Error & Continue

This variable indicates the action you want Module for EDI 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 Module for EDI Can Take for Invalid Control Numbers”](#) on page 234.

Value	Description
Error & Continue	Module for EDI logs the error and then continues normal processing of the EDI document that contains the invalid control number.
ProcessNormally	Module for EDI logs a warning and then continues normal processing of the EDI document that contains the invalid control number.
Reject	Module for EDI logs the error and then sends the document to Trading Networks without splitting it. (Typically, Module for EDI splits an inbound EDI based on the EDITPA <i>splitOption</i> variable and sends the split documents to Trading Networks for processing.)

Additionally, Module for EDI 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 236.

If desired, you can force a duplicate document to be processed later. For more information, see [“Reprocessing EDI Documents with Invalid Control Numbers”](#) on page 238.

TRADACOMS/ControlNumberManagement/outOfSequenceControlNumberAction EDITPA Variable

Default: Error & Continue

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

For more information about each of the following actions, see [“Actions Module for EDI Can Take for Invalid Control Numbers”](#) on page 234.

Value	Description
Error & Continue	Logs the error; then continues normal processing of the EDI document that contains the invalid control number.
ProcessNormally	Logs a warning and continues normal processing of the EDI document that contains the invalid control number.
Reject	<p>Logs the error and sends the document to Trading Networks without splitting it. Typically, Module for EDI splits an inbound EDI based on the EDITPA <i>splitOption</i> variable and sends the split documents to Trading Networks for processing.</p> <p>Additionally, the 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 236.</p> <p>You can later force processing of the out-of-sequence document if you want. For more information, see “Reprocessing Documents with Out-of-Sequence Control Numbers” on page 240.</p>

***TRADACOMS/enforceApplicationReference* EDITPA Variable**

Default: false

This variable is used to validate that the contents of the APRF field in the STX segment match the file type of the messages in the transmission.

Value	Description
true	Validates that the APRF field contents match the file type of the messages.
false	Does not perform APRF validation.

***TRADACOMS/OutBound/createReconciliationMessage* EDITPA Variable**

Default: false

This variable indicates whether Module for EDI creates a reconciliation (RSGRSG) message in outbound batched EDI documents.

Value	Description
true	Creates an RSGRSG message at the end of the transmission for the batched TRADACOMS document.
false	Does not create an RSGRSG message.

***TRADACOMS/OutBound/recipientTransmissionReference* EDITPA Variable**

Default: false

(optional) This variable indicates whether Module for EDI returns the receiver's transmission reference in outbound EDI documents.

Value	Description
true	Returns the receiver's transmission reference in outbound EDI documents.
false	Does not return the receiver's transmission reference.

***TRADACOMS/OutBound/applicationReference* EDITPA Variable**

(optional) This variable returns 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* EDITPA Variable**

Default: false

(optional) This variable indicates whether Module for EDI returns 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.

Value	Description
true	Returns the value of the APRF field, which is the file type specified in the transmission, for example, ORDHDR.
false	Uses the value specified for the <i>TRADACOMS/Outbound/applicationReference</i> variable.

***TRADACOMS/persistMultipleDocEnvelope* EDITPA Variable**

Default: true

This variable indicates whether Module for EDI saves the original EDI document in the Trading Networks database. The original EDI document that Trading Networks receives typically contains multiple transmission segments. The module only uses the *persistMultipleDocEnvelope* variable from the default EDITPA.

Note: Module for EDI 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. You can control whether the Transmission, Batch, and File documents are saved to the Trading Networks database via a processing rule.

Value	Description
true	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>Note: This behavior occurs only if the original EDI document contains multiple transmission segments.</p>
false	Does not 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.

TRADACOMS/publishBatchFailEvent EDITPA Variable

Default: false

This variable indicates whether Module for EDI publishes an IS document when it cannot 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` service. For more information about this IS document type, see *webMethods Module for EDI Built-In Services Reference*. For more information about how this EDITPA variable is used during batching and how to handle failure to publish an IS document, see [“Updating the Task Status and Publishing Documents for Failed Tasks” on page 283](#).

Value	Description
true	Publishes an IS document when it encounters errors including an EDI document into the final batch EDI document.
false	Does not publish an IS document when it encounters errors including an EDI document into the final batch EDI document.

Storage Options for File Document Types

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)

For example:

Message contents	The default storage method creates...
STX Header1Detail1 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 more information, see *webMethods Module for EDI 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), My webMethods may not be able to 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.

For example:

Message contents	The default storage method creates...	The alternative 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	Three file document types (one per detail message), each with six content parts as follows: Document type 1: STX Header1 Detail1 VAT1 Trailer1 END

Message contents	The default storage method creates...	The alternative storage method creates...
	END	<p>Document type 2:</p> <pre>STX Header1 Detail2 VAT1 Trailer1 END</pre> <p>Document type 3:</p> <pre>STX Header1 Detail3 VAT1 Trailer1 END</pre>

To select which storage option to use, set the *TRADACOMS/storageOption* variable in your EDITPA, as described in the [“TRADACOM Variables” on page 103](#).

Errors that Prevent the Splitting of Transmissions

The following categories of errors will prevent the *TRADACOMS/splitOption* EDITPA variable from splitting a TRADACOMS transmission. For information about setting the *TRADACOMS/splitOption* variable, see [“TRADACOM Variables” on page 103](#).

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

- When ANAA is specified in the STX segment and:
 - There is no RSGRSG
 - The control number in the RSGRSG does not match the control number in STX
 - The Sender Reference in RSGRSG does not match the Sender Reference in STX
- When ANAA is not specified in STX segment and:

- 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

5 Defining Control Number Information for Trading Partners

■ Overview	114
■ Managing Control Number Information	114
■ Turning Inbound Control Number Validation On and Off	121
■ Defining Control Number Information for Processing Documents	123
■ Defining Actions Module for EDI Can Take for Invalid Control Numbers	124
■ Searching for Existing Control Number Settings	127

Overview

A *control number* is a number in the header of a non-TRADACOMS EDI document that is used for validation and for ordering documents exchanged between trading partners.

For inbound EDI documents, Module for EDI can perform control number validation, which determines whether control numbers are valid or invalid and detects duplicate or out-of-sequence control numbers. Also, inbound control number validation determines whether the control numbers are in order, and therefore whether the EDI documents arrived in order. However, inbound control number validation does not guarantee that EDI documents are processed in the same order in which they are received.

This chapter discusses actions that are taken to define and manage control number information. For information about control number validation, see [“Optional Inbound Processing When Using Trading Networks”](#) on page 229.

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 the same way in Trading Networks. The only difference is in the terminology. For simplicity, Trading Networks and Module for EDI use the term *control number* to mean either control number or transmission reference number.

Managing Control Number Information

Module for EDI maintains information about control numbers in the EDIControlNumber table, which is a Module for EDI-specific table in the Trading Networks database. The module maintains control number information for unique combinations of:

- Sender/receiver
- EDI standard and version
- Production mode (Production, Test, or Custom). For VDA, the Production mode is always set to Custom.
- Type
 - For ANSI X12 or UN/EDIFACT (and all supported sub-standards), type is one of the following:
 - Envelope, if the control number is to be used for an interchange header
 - Group, if the control number is to be used for a group header
 - For TRADACOMS, type is one of the following:
 - Envelope, if the control number is to be used for a TRADACOMS transmission header
 - Batch, if the control number is to be used for a TRADACOMS batch header, a TRADACOMS file header (for example, ORDHDR, ACKHDR, and INVHDR), or a

message type for a TRADACOMS file type, (for example, ORDHDR, ACKHDR, and INVHDR).

- For VDA, type is Envelope

When Module for EDI is processing an EDI document and requires control number information, the module uses the above information from the EDI document being processed to look up the control number information to use. For example, if the 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 following table lists the actions you can take to manage control number information, as well as where you can find more information about each action.

To...	See...
Define how Module for EDI creates the control number sequence	“Defining How Module for EDI Creates the Control Number Sequence” on page 115
Add entries to the EDIControlNumber table for unique combinations of sender/receiver/EDI standard/version/production mode/type	“Adding Entries to the EDIControlNumber Table” on page 116
Configuring control number settings	“Configuring Control Number Settings” on page 116
Updating control number settings	“Updating Control Number Settings” on page 121

Defining How Module for EDI Creates the Control Number Sequence

Use the EDIAllowCustomControlNumberSequence property to define how Module for EDI creates the control number sequence. For more information, see [“Defining Module for EDI Properties” on page 24](#).

Adding Entries to the EDIControlNumber Table

Entries can be added to the EDIControlNumber table for unique combinations of sender/receiver/EDI standard/version/production mode/type.

Module for EDI automatically adds entries to the EDIControlNumber table when it is validating inbound control numbers. In My webMethods, you can add a control number for a specific combination of sender/receiver/EDI standard/version/production mode/type, and then the module adds the control number to the EDIControlNumber table. Additionally, you can add entries that you want to use for outbound processing. For more information, see [“Configuring Control Number Settings” on page 116](#).

Configuring Control Number Settings

Module for EDI uses the control number maximum, minimum, increment, and window settings to determine the next control number to expect, whether a control number is valid or invalid, and whether an invalid control number is a duplicate or an out-of-sequence control number. Module for EDI maintains these settings in the EDIControlNumber table.

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

Setting	Description	Default
control number	The next control number to expect. You can set an initial value. Module for EDI updates this setting so it is always set to the next expected control number.	1
control number increment	The value you want Module for EDI to use to increment a control number when determining the next expected control number.	1
control number maximum	The highest number the control number can be.	99999999999999
control number minimum	The lowest number the control number can be.	0 (for VDA) 1 (for all other standards)
control number window	A number used to compute a range of numbers. Module for EDI uses this range of numbers to determine whether an invalid control number is a duplicate or an out-of-sequence control number. For more information, see “Invalid Control Numbers” on page 231 .	100

For example, Module for EDI receives an Interchange document that has an interchange header identifying sender A, receiver B, a production mode of Production, and a type of Envelope. The module locates the control number settings by locating the EDIControlNumber table entry for

sender A, receiver B, where production mode is Production, and type is Envelope. If the 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 module:

1. Adds an entry to the table
2. Sets the control number maximum, minimum, increment, and window to their defaults
3. Sets the next expected control number to the sum of the control number from the inbound EDI document plus the control number increment

Note:

It is possible for Module for EDI to encounter more than one sender/receiver pair that maps to the same Trading Networks internal ID. For example, when you have the same sender ID/receiver ID pair combination for two separate sender/receiver pairs, the two pairs map to the same Trading Networks internal ID. Because Module for EDI uses the same Trading Networks internal ID to identify senders and receivers in the EDIControlNumber table, both sender/receiver pairs correlate to the same row in the EDIControlNumber table. As a result, when Module for EDI receives a document from either sender/receiver pair, it will update the next expected control number in the same row of the EDIControlNumber table.

When your environment is set up so that Module for EDI maintains separate control number sequences for each sender/receiver pair, the module will indicate that the example's control numbers are invalid. This is because the module checks the control number sequences for two sender/receiver pairs using the same EDIControl number entry.

To identify sender and receiver pairs correctly in this scenario, set the EDIAllowCustomControlNumberSequence property to true. When EDIAllowCustomControlNumberSequence is set to true, Module for EDI looks up the next control number using both the sender/receiver ID and the sender/receiver qualifiers.

For more information about the EDIAllowCustomControlNumberSequence property, see [“Defining Module for EDI Properties” on page 24](#).

Configure control number settings using either the Control Numbers page or the Partner Profiles page in My webMethods. For more information about configuring control number settings using the Control Numbers page, see [“Configuring Control Number Settings Using the My webMethods Control Numbers Page” on page 117](#). For more information about configuring control number settings using the Partner Profiles page, see [“Configuring Control Number Settings Using the My webMethods Partner Profiles Page” on page 120](#).

Note:

You must have the Manage Control Numbers general functional permission to configure control number settings using the Control Number page.

Configuring Control Number Settings Using the My webMethods Control Numbers Page

Use this procedure to define control number settings using the My webMethods Control Numbers page.

➤ **To configure control number settings using the Control Numbers page**

1. In My webMethods: **Administration > Integration > B2B Settings > Control Numbers**.
2. Click **Add Control Number**.
3. If you have set the value of the EDIAllowCustomControlNumberSequence property to
 - True or strictCompatibility, select one of the following options in the Define By panel to define the control number:
 - **Partner name** (Default)
 - **Partner external ID**
 - False, define the control number for your partner using the corporation name of the partner.

For more information about the EDIAllowCustomControlNumberSequence property, see [“Defining Module for EDI Properties” on page 24](#).
4. In the Control Number Definition panel, specify information that describes when to use the control number as follows:

Field	Description
Sender	<p>The corporation name of the sender. Click Select to select the sender from the partner list. This field is available when:</p> <ul style="list-style-type: none"> ■ The value of the EDIAllowCustomControlNumberSequence property is false. ■ The value of the EDIAllowCustomControlNumberSequence property is true or strictCompatibility and you define the control number using Partner name.
Sender ID	<p>The Trading Networks ID for the sender. To determine the ID, you can run the wm.b2b.editn:ediPartnerIDToTNPartnerID service from Software AG Designer. Copy the resulting internal ID from the bottom of the Results tab and paste it into this field.</p> <p>This field is available when the value of the EDIAllowCustomControlNumberSequence property is true or strictCompatibility and you define the control number using Partner external ID.</p>
Sender Qualifier	<p>Leave this blank. This field is available when the value of the EDIAllowCustomControlNumberSequence property is true or</p>

Field	Description
	strictCompatibility and you define the control number using Partner external ID .
Receiver	<p>The corporation name of the receiver. Click Select to select the receiver from the partner list. This field is available when:</p> <ul style="list-style-type: none"> ■ The value of the EDIAllowCustomControlNumberSequence property is false. ■ The value of the EDIAllowCustomControlNumberSequence property is true or strictCompatibility and you define the control number using Partner name.
Receiver ID	<p>The Trading Networks ID for the receiver. To determine the ID, you can run the wm.b2b.editm:ediPartnerIDToTNPartnerID service from Designer. Copy the resulting internal ID from the bottom of the Results tab and paste it into this field.</p> <p>This field is available when the value of the EDIAllowCustomControlNumberSequence property is true or strictCompatibility and you define the control number using Partner external ID.</p>
Receiver Qualifier	Leave this blank. This field is available when the value of the EDIAllowCustomControlNumberSequence property is true or strictCompatibility and you define the control number using Partner external ID .
Production Mode	<p>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.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: For TRADACOMS and VDA, the only valid value is Production.</p> </div>
Standard	The EDI standard of the documents in which to use the control number. Select X12 , UNEDIFACT , EANCOM , UCS , VICS , VDA , 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.
Group Type	Whether the control number is to be used for an interchange (or transmission) header or a group (or batch) header.

Field	Description
	<ul style="list-style-type: none"> ■ Select Envelope if you are adding a control number for an interchange-level (or transmission-level) document. ■ Select Select if you are adding a control number for a group-level (or batch-level) document that has a specific group type, and then in the Select Group Type box, type one of the following: <ul style="list-style-type: none"> ■ For non-TRADACOMS documents: the name of the group (for example, PO or IN) ■ For TRADACOMS documents: the name of the batch or file type (for example, INVHDR or ORDHDR) ■ Select All if you are adding a control number that applies to all group-level documents.

5. In the Settings panel, specify information that describes the value to use for the control number as follows:

Field	Description
Duplicate Detection Window	A number that indicates a range of numbers that Module for EDI uses to determine whether a number is duplicate or out-of-sequence.
Next Control Number	The next control number that you expect for the interchange or group (or transmission or batch) to use.
Increment	The value that Module for EDI uses to increment the control number to determine the next expected control number.
Minimum	The lowest number that the control number can be.
Maximum	The highest number that the control number can be. If Module for EDI calculates the next expected control number and that number is greater than the control number maximum, Module for EDI sets the control number to the value you specify in the Minimum field.

Note:

When setting the values for **Maximum**, **Minimum**, and **Window**, the difference between **Maximum** and **Minimum** must be at least two times as large as the value you specify for **Window**.

6. Click **Save & Close**. Module for EDI adds the control number information you defined to the EDIControlNumber table.

Configuring Control Number Settings Using the My webMethods Partner Profiles Page

Use this procedure to configure control number settings using the My webMethods Partner Profile page.

➤ **To configure control number settings using the Partner Profiles page**

1. In My webMethods: **Administration > Integration > B2B > Partner Administration > Partner Profiles**.
2. On the Partner Profiles page, select a partner and click the **Edit** icon (✎).
3. On the Partner Details page, click the **Control Number** tab.

Note:

The **Control Number** tab will be available in the Partner Details page only if you have Edit EDI Control Number Sequences data permission and Manage Control Numbers general functional permission. For more information, see [“Assigning User-Specific Permissions in My webMethods” on page 22](#).

4. Click **Add Control Number** and continue with step 3 in [“Configuring Control Number Settings Using the My webMethods Control Numbers Page” on page 117](#).

Updating Control Number Settings

Use the following procedure to update any of the control number settings in the EDIControlNumber table.

➤ **To update control number settings**

1. In My webMethods: **Administration > B2B Settings > Control Numbers**
2. Search for the control number entry that you want to update. For information about searching the table, see [“Searching for Existing Control Number Settings Using My webMethods ” on page 127](#).
3. Click the **Edit** icon and update the control number settings, as desired.

Turning Inbound Control Number Validation On and Off

You can have the Module for EDI validate interchange control numbers, group control numbers, or both (or any combination of TRADACOMS transmission, batch, and file control numbers). By default, inbound control number validation is turned off.

Because you turn validation on or off using EDITPA variables, you can control whether Module for EDI validates control numbers for all sender/receiver pairs or for specific sender/receiver pairs.

Important: Module for EDI does not validate control numbers that contain non-numeric characters.

The following sections summarize the EDITPA variables used to turn validation on and off for each type of control number.

Activating Validation for ANSI X12 or UN/EDIFACT Documents

For ANSI X12 or UN/EDIFACT documents (and all supported sub-standards), use the EDITPA variables in the following table. For more information about these variables, see [“ControlNumberManagement Variables” on page 83](#).

Type of control number	How to turn validation on or off
Interchange	Set the <i>ControlNumberManagement/validateInboundEnvelopeControlNumbers</i> EDITPA variable as follows: <ul style="list-style-type: none"> ■ <code>true</code>—Turns on the interchange control number validation. ■ <code>false</code>—Turns off the interchange control number validation.
Non-standard	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 Module for EDI home page. For more information, see “Turning Inbound Control Number Validation On or Off When Using Non-Standard Processing” on page 350 .
Group	Set the <i>ControlNumberManagement/validateInboundGroupControlNumbers</i> EDITPA variable as follows: <ul style="list-style-type: none"> ■ <code>true</code>—Turns on the group control number validation. ■ <code>false</code>—Turns off the group control number validation.

Activating Validation for VDA Documents

For VDA documents, use the following EDITPA variables. For more information about these variables, see [“ControlNumberManagement Variables” on page 83](#).

Type of control number	How to turn validation on or off
Interchange	Set the <i>ControlNumberManagement/validateInboundEnvelopeControlNumbers</i> EDITPA variable as follows: <ul style="list-style-type: none"> ■ <code>true</code>—Turns interchange control number validation on. ■ <code>false</code>—Turns interchange control number validation off.

Type of control number	How to turn validation on or off
Structure	<p>Set the <i>ControlNumberManagement/VDA/validateStructure</i> EDITPA variable as follows:</p> <ul style="list-style-type: none"> ■ <code>true</code>—Turns structure control number validation on. ■ <code>false</code>—Turns structure control number validation off.

Activating Validation for TRADACOMS Documents

For TRADACOMS documents, use the EDITPA variables in the following table. For more information about these variables, see [“Defining Trading Partner Information When Using TRADACOMS” on page 97](#).

Type of control number	How to turn validation on or off
Transmission	<p>Set the <i>TRADACOMS/ControlNumberManagement/validateInboundTransmissionControlNumbers</i> EDITPA variable as follows:</p> <ul style="list-style-type: none"> ■ <code>true</code>—Turns transmission control number validation on. ■ <code>false</code>. Turns transmission control number validation off.
Batch	<p>Set the <i>TRADACOMS/ControlNumberManagement/validateInboundBatchControlNumbers</i> EDITPA variable as follows:</p> <ul style="list-style-type: none"> ■ <code>true</code>—Turns batch control number validation on. ■ <code>false</code>—Turns batch control number validation off.
File	<p>Set the <i>TRADACOMS/ControlNumberManagement/validateInboundFileControlNumbers</i> EDITPA variable as follows:</p> <ul style="list-style-type: none"> ■ <code>true</code>—Turns file control number validation on. ■ <code>false</code>—Turns file control number validation off.

Defining Control Number Information for Processing Documents

This section describes how to define control number information needed to process both inbound and outbound EDI documents.

Defining Control Number Information for Processing Inbound Documents

The following table describes the control number settings that you can define for inbound EDI documents, where to define them, and what information is associated with each setting.

Settings	Define in...	Setting is associated with...
Whether you want Module for EDI to validate inbound control numbers. For more information, see “Validating Inbound Control Numbers” on page 230 and “Turning Inbound Control Number Validation On and Off” on page 121 .	EDITPA	Sender/Receiver pair identified in the EDITPA
Actions you want Module for EDI to take when it encounters an invalid control number. For more information, see “Actions Module for EDI Can Take for Invalid Control Numbers” on page 234 .		
Settings used to track control numbers and determine whether an invalid control number is a duplicate or out of sequence. For more information, see “Managing Control Number Information” on page 114 .	EDIControlNumber table	Unique combination of: <ul style="list-style-type: none"> ■ Sender/Receiver ■ Production mode (for example, Testing or Production) ■ Type (Envelope or Group)
Initial value of the control number you expect to receive in an inbound document. For more information, see “Configuring Control Number Settings” on page 116 .		

Defining Control Number Information for Processing Outbound 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 [“Obtaining Control Numbers for Outbound Processing \(ANSI X12 and UN/EDIFACT\)” on page 263](#).

Defining Actions Module for EDI Can Take for Invalid Control Numbers

How you define the actions Module for EDI can take for invalid control numbers depends on the EDI standard you are using.

EDI Standard	Where to Find More Information
ANSI X12 or UN/EDIFACT	“Defining How to Handle Invalid Control Numbers for ANSI X12 or UN/EDIFACT Documents” on page 125
VDA	“Defining How to Handle Invalid Control Numbers for VDA Documents” on page 126
TRADACOMS	“Defining How to Handle Invalid Control Numbers for TRADACOMS Documents” on page 126

For a description of the specific actions Module for EDI can take when it encounters an invalid control number, see [“Actions Module for EDI Can Take for Invalid Control Numbers” on page 234](#).

Defining How to Handle Invalid Control Numbers for ANSI X12 or UN/EDIFACT Documents

The following table describes how to define the action you want Module for EDI 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 <i>ControlNumberManagement/duplicateControlNumberAction</i> EDITPA variable. For details, see “ControlNumberManagement/duplicateControlNumberAction EDITPA Variable” in “ControlNumberManagement Variables” on page 83 .
Out-of-sequence control number	Use the <i>ControlNumberManagement/outOfSequenceControlNumberAction</i> EDITPA variable. For details, see “ControlNumberManagement/outOfSequenceControlNumberAction EDITPA Variable” in “ControlNumberManagement Variables” on page 83 .
Non-standard	Note: 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 Module for EDI home page. For more information, see “Defining Actions for Invalid Control Numbers” on page 351 . For more information about the difference between standard and non-standard processing, see <i>webMethods Module for EDI Concepts Guide</i> .

Defining How to Handle Invalid Control Numbers for VDA Documents

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

Type of invalid control number	How to define the action
Duplicate control number	Use the <i>ControlNumberManagement/duplicateControlNumberAction</i> EDITPA variable. For details, see " <i>ControlNumberManagement/duplicateControlNumberAction</i> EDITPA Variable" in " ControlNumberManagement Variables " on page 83.
Out-of-sequence control number	Use the <i>ControlNumberManagement/outOfSequenceControlNumberAction</i> EDITPA variable. For details, see " <i>ControlNumberManagement/outOfSequenceControlNumberAction</i> EDITPA Variable" in " ControlNumberManagement Variables " on page 83.
Invalid structure	Use the <i>ControlNumberManagement/VDA/validateStructure</i> EDITPA variable. All control number fields are mandatory and are used to check the completeness of a message. For details, see " <i>ControlNumberManagement/VDA/validateStructure</i> EDITPA Variable" in " ControlNumberManagement Variables " on page 83.

Defining How to Handle Invalid Control Numbers for TRADACOMS Documents

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

Type of invalid control number	How to define the action
Duplicate control number	Use the <i>TRADACOMS/ControlNumberManagement/duplicateControlNumberAction</i> EDITPA variable. For details, see " <i>TRADACOMS/ControlNumberManagement/duplicateControlNumberAction</i> EDITPA Variable" in " TRADACOM Variables " on page 103.
Out-of-sequence control number	Use the <i>TRADACOMS/ControlNumberManagement/outOfSequenceControlNumberAction</i> EDITPA variable. For details, see " <i>TRADACOMS/ControlNumberManagement/outOfSequenceControlNumberAction</i> EDITPA Variable" in " TRADACOM Variables " on page 103.

Searching for Existing Control Number Settings

Note: This section describes how to search for existing control number settings using My webMethods home page.

Searching for Existing Control Number Settings Using My webMethods

When you have Manage Control Numbers general functional permission, you can search for existing control number settings to view and/or update using My webMethods.

➤ To search for existing control number settings using My webMethods

1. In My webMethods, navigate to the Control Numbers page by way of either of the following:

- Using B2B settings, go to **Administration > Integration > B2B Settings > Control Numbers**.
- Using the Partner Profiles page:
 1. Go to **Administration > Integration > B2B > Partner Administration > Partner Profiles**.
 2. Select **Edit** next to the partner profile for which you want to search.
 3. On the Partner Details page, click the **Control Number** tab.

Note:

The **Control Number** tab will be available in the Partner Details page only if you have Edit EDI Control Number Sequences data permission and Manage Control Numbers general functional permission. For more information, see [“Assigning User-Specific Permissions in My webMethods” on page 22](#).

2. Perform either a basic search if you know the control number, or an advanced search if you do not know the control number or if you want to display a list of control numbers that meet certain criteria.

- To perform a basic search:
 1. On the **Keyword** tab, enter the search keyword in the **Keywords** field.
 2. Click **Search**. Module for EDI executes the search in the Trading Networks database and displays the search results in the Control Numbers panel.
 3. If you want to save the search criteria, click **Save**, type a name and description for the search, and then click **Save**.
- To perform an advanced search:

1. On the **Advanced** tab, select a search field, an appropriate operator, and a value as follows:

For this field...	Specify this operator...	And this value...
Production Mode	Equals	The production mode associated with the interchange (or transmission) or group (or batch) header for which to use the control number. Select CUSTOM , PRODUCTION , or TEST .
	Not Equals	
Group Type	Equals	If you want to search for control numbers for: <ul style="list-style-type: none"> ■ Interchange or TRADACOMS transmission headers, type ENVELOPE. ■ Group or batch headers for all group or TRADACOMS batch types, type ALL. ■ Group or TRADACOMS batch headers of a specific type, type the group type (for example, PO) or the batch TRADACOMS type BATCH. ■ TRADACOMS transmissions, type the TRADACOMS file type.
	Not Equals	
	Contains	
	Begins With	
Receiver	Equals	The corporation name of the receiver.
	Not Equals	To locate and select a partner from a list, click Select Partner .
	Contains	
	Begins With	
Receiver External ID	Equals	The receiver ID for the interchange or group.
	Not Equals	If you are searching for information about a control number that was added by the Module for EDI batching feature and for which neither the sender nor the receiver has a Trading Networks profile, specify the ID of the receiver in this field and then click the  icon to add another search criteria row for Receiver Qualifier .
	Contains	
	Begins With	
Is Blank		
Receiver Qualifier	Is Not Blank	If you are searching for information about a control number that was added by the Module for EDI batching feature for a receiver that does not have a Trading Networks profile, specify the EDI ID
	Equals	
	Is Blank	

For this field...	Specify this operator...	And this value...
	Not Equals Contains Begins With	qualifier associated with the ID you specified for Receiver External ID .
Sender	Equals Not Equals Contains Begins With	<p>The corporation name of the sender.</p> <p>To locate and select a partner from a list, click Select Partner.</p>
Sender External ID	Equals Not Equals Contains Begins With	<p>The sender ID for the interchange or group (or transmission or batch).</p> <p>If you use the Module for EDI batching feature, Module for EDI might encounter a sender/receiver pair for which no Trading Networks profile exists for the sender, receiver, or both; therefore, no Trading Networks ID exists. If neither the sender or receiver has a Trading Networks ID, Module for EDI saves the ID and EDI qualifier for the sender and receiver from the EDI document in the EDIControlNumber table.</p> <p>To search for control numbers for a sender/receiver pair that does not have a Trading Networks profile, specify the ID of the sender in this field, click on  to add another search criteria row for Sender Qualifier, and specify the EDI ID qualifier in this row.</p>
Sender Qualifier	Is Blank Is Not Blank Equals Not Equals Contains Begins With	If you are searching for information about a control number that was added by the Module for EDI batching feature for a sender that does not have a Trading Networks profile, specify the EDI ID qualifier associated with the ID you specified for Sender External ID .
Standard	Equals Not Equals	The EDI standard of the documents that use the control number.

For this field...	Specify this operator...	And this value...
Version	Equals Not Equals Contains Begins With	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.

2. If you want to add another search criteria row, click **Add** and specify additional search criteria.
3. Click **Search** to execute the search in the Trading Networks database. The system displays the search results in the Control Numbers panel.
4. If you want to save the search criteria, click **Save**, type a name and description for the search, and then click **Save** again.

6 Creating Clients that Send EDI Documents to Integration Server

■ Overview	132
■ Defining the Content Type to Use	132
■ Before Creating the Service to Form an EDI Document	132
■ Creating the Service the Client Invokes to Handle EDI Documents	133
■ Sending EDI Documents to Integration Server via HTTP	136
■ Sending EDI Documents to Integration Server via FTP	138
■ Sending EDI Documents to Integration Server via File Polling	139

Overview

To send an EDI document to Integration Server for processing, you create a client to Integration Server. The client you create must:

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

For more information about creating clients, see the chapter about creating client code in the *webMethods Service Development Help* for your release.

For more information about using clients for inbound EDI document processing, see the *webMethods Module for EDI Concepts Guide*.

Defining the Content Type to Use

The client you create should send the EDI document to Integration Server using the application/EDIstream content type. When you use application/EDIstream, Integration Server passes the document to the Module for EDI content handler as an InputStream. The Module for EDI content handler forms the pipeline with the *edidata* parameter and assigns this parameter as the pointer to InputStream. Then the content handler invokes the service that the client specifies.

Note: For backward compatibility, Module for EDI also has content handlers to accept documents with the content types application/EDI, application/X12, and application/UNEDIFACT. With these content types, the Module for EDI 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 application/EDIstream content type because it conserves system memory.

Before Creating the Service to Form an EDI Document

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

- The flat file schema that defines the structure of the EDI document that you are forming.

Module for EDI uses the schema to create the EDI document from an IData object. For instructions, see [“Creating Flat File Schemas for EDI Documents” on page 358](#).

- 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 Software AG Designer 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 Service Development Help* for your release.

Creating the Service the Client Invokes to Handle EDI Documents

You create a service that the client invokes to process EDI documents. How the client identifies this service depends on whether you are sending the EDI document to Integration Server via HTTP, FTP, or File Polling. For more information, see “Identifying the Service to Invoke” in “Sending EDI Documents to Integration Server via HTTP” on page 136, “Sending EDI Documents to Integration Server via FTP” on page 138, and “Sending EDI Documents to Integration Server via File Polling” on page 139.

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 Center Samples area on the Empower Product Support website at <https://empower.softwareag.com>.

Important:

Delete the WmEDIsamples package before going into production.

Logic to Include in the Service to Form an EDI Document

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

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

```

1 => wm.b2b.edi:convertToValues
2 🏠 BRANCH on '/errors' (if erros, perform error handling)
  └─ Snull: SEQUENCE
    3 📌 MAP (Map data from IData object for internal documetns to IData objects for EDI document)
    4 => wm.b2b.edi:convertToString (Convert the IData object for EDI document to a String)
    5 => wm.b2b.edi:convertToValues (optional :validating the edi document)
    🏠 BRANCH on '/errors' (if errors,perform error handling)
      └─ Snull: SEQUENCE
        6 => wm.b2b.edi.util:addGroupEnvelope (add group envelope according to ANSI X12 UCS or VICS)
        7 => wm.b2b.edi.util:addICEEnvelope (add interchange envelope according to ANSI X12 UCS or VICS)
        8 => CustomCode:sendEDIIdocument
      Sdefault: SEQUENCE (handle validation errors)
    Sdefault: SEQUENCE (Handle errors)

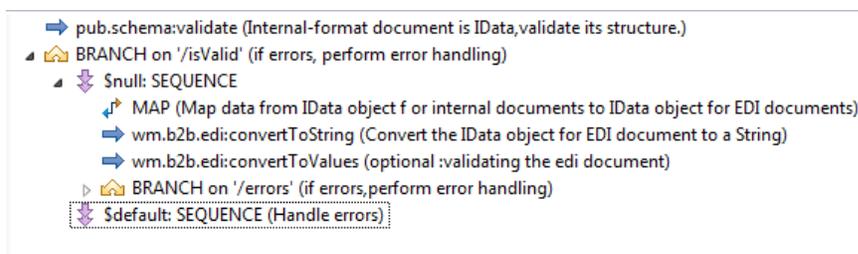
```

Flow operation	Description
1	<p>Invoke the <code>wm.b2b.edi:convertToValues</code> service to convert the incoming internal-format document, which is either a <code>String</code> or an <code>InputStream</code>, into an <code>IData</code> object. If you want, you can set the input parameters of the <code>convertToValues</code> service to have it validate the structure of the internal-format document. The inputs to the <code>convertToValues</code> service include the internal-format document and the flat file schema that defines the structure of the internal-format document.</p> <p>For backward compatibility, you can use an IS document type to define the structure of the 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 <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 362.</p>
2	<p>Add your own logic to handle errors that might result from executing the <code>convertToValues</code> service, for example, validation errors.</p>
3	<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.</p>
4	<p>Invoke the <code>wm.b2b.edi:convertToString</code> service to convert the EDI document from an <code>IData</code> object to <code>String</code> format. 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 <code>values</code> input parameter 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 IS 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 type of encoding the <code>convertToString</code> service uses to write data to the output file. When no value is provided for the <code>encoding</code> input parameter, the <code>convertToString</code> service uses the <code>EDIencoding</code> property located in the <code>IntegrationServer_directory\instances\instance_name\packages\WmEDI\config\properties.cnf</code> file. When the <code>EDIencoding</code> property is not specified, UTF-8 is used. <p>For more information about the <code>EDIencoding</code> property, see “Defining Module for EDI Properties” on page 24.</p>

Flow operation	Description
5	<p>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:</p> <ul style="list-style-type: none"> ■ The EDI document. The <i>string</i> output parameter of the <code>convertToString</code> service contains the EDI document. Map this to the <i>edidata</i> input parameter of the <code>convertToValues</code> service. ■ The flat file schema that defines the structure for the EDI document.
6	<p>If needed, add the group envelope to the EDI document by invoking either the <code>wm.b2b.edi.util:addICEEnvelopeEDIFACT</code> service if you are creating a UN/EDIFACT EDI document or the <code>wm.b2b.edi.util:addGroupEnvelope</code> service if you are creating another kind of EDI document.</p> <p>For more information about adding UN/EDIFACT envelopes, see “Adding UN/EDIFACT Envelopes” on page 362.</p>
7	<p>Add the interchange envelope to the EDI document by invoking either the <code>wm.b2b.edi.util:addICEEnvelopeEDIFACT</code> service if you are creating a UN/EDIFACT EDI document or the <code>wm.b2b.edi.util:addICEEnvelope</code> service if you are creating another kind of EDI document. For more information about adding UN/EDIFACT envelopes, see “Adding UN/EDIFACT Envelopes” on page 362.</p>
8	<p>Add your own logic or invoke a service that you create to send the EDI document outbound.</p>

Validating the Input Internal-Format Document When It Is an IData Object

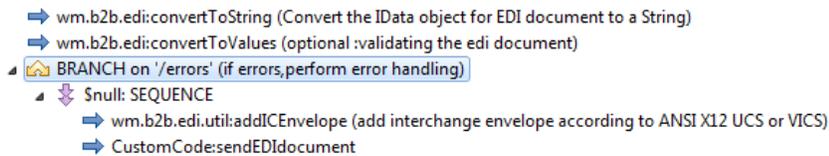
The logic in the sample code for forming an EDI document from an internal-format document assumes the internal-format document is passed to your service as a `String` or `InputStream`. In this situation, you can have the `wm.b2b.edi:convertToValues` service validate the document. If your client passes 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 in the following diagram:



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* for your release.

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` service or the `wm.b2b.edi.util:addICEnvelope` service, as shown in the following diagram:



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

Validating the EDI Document Envelope

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

Sending EDI Documents to Integration Server via HTTP

You can create a client that uses HTTP or HTTPS to post an EDI document to Integration Server. The HTTP client must meet the following requirements :

- The client must be able to send the EDI document as a String of data to Integration Server using the HTTP POST method.
- The client must be able to set the value of the content type (for example, `application/EDIStream`) 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 logic to:

- Submit a POST request to Integration Server.
- Address the request to the URL of the service that is to process the EDI document, for example, `http://rubicon:5555/invoke/myEDIServices.v1/processEDIDoc`.
- Set the content type to `application/EDIstream` in the HTTP request header.
- Put the EDI document to be processed 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 like you might do with an XML document.

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

The following example describes the values that you would set when using 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* for your release.

Set this parameter...	Of data type...	To...
<i>url</i>	String	The URL of the service that you want to invoke to process the EDI document. Example: <code>http://rubicon:5555/invoke/myEDIServices.v1/processEDIDoc</code> In this example, the <code>processEDIDoc</code> service located in the <code>myEDIServices.v1</code> folder on the "rubicon" server with port number "5555" would be invoked.
<i>method</i>	String	post
<i>loadAs</i>	String	The data type of the input data source. Specify one of the following: <ul style="list-style-type: none"> ■ <code>bytes</code>. The document data source is a <code>byte[]</code>. ■ <code>stream</code>. The document data source is an <code>InputStream</code>.
<i>data/string</i>	String	The EDI document that you want to post.
<i>headers</i>	Document	An <code>IData</code> object that contains the following:

Set this parameter...	Of data type...	To...
	Parameter	Value
	<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.

Sending EDI Documents to Integration Server via FTP

You can create a client that sends an EDI document to 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 logic to:

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

For 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 send a file to a service in the `myEDIServices.v1` folder, you use `\ns\myEDIServices\v1\ ServiceName` as the path to that service, not `Integration Server_directory\instances\ $instance_name\ packages\ myEDIServices\v1\ ServiceName`.

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

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

For 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 Parameters 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 send 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* for your release.

Set this parameter...	Of data type...	To...
<i>serverhost</i>	String	The name of the machine running Integration Server.
<i>serverport</i>	String	The port on which 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	<code>put</code>
<i>dirpath</i>	String	The path representing the service to invoke, for example, <code>\ns\myEDIServices\v1\processEDIDoc</code> .
<i>localfile</i>	String	The name of the source file containing the EDI document, for example, <code>x12_850</code> .
<i>remotefile</i>	String	The name to assign the file on Integration Server and the content type. Use the following format: <i>filename;content type:content sub-type</i> For example, <pre>x12_850;application:EDIStream</pre>
<i>secure</i>	Document	Indicates whether the FTP session is with a secure FTP server. The <i>auth</i> parameter specifies the kind of authentication mechanism to use (SSL, TLS, or TLS-P), and the <i>securedata</i> parameter 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 Integration Server via File Polling

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

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

When files such as EDI documents are placed in this directory (for example, sent via FTP into the directory), Integration Server uses the content type you specify. Because the content type is an EDI content type, when Integration Server receives the EDI document in the monitored directory, it passes the document to the appropriate Module for EDI 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

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 the...
Monitoring Directory	Directory on Integration Server that the listener monitors for incoming EDI documents.
Content type	Content type that you want Integration Server to use for documents that are placed in the Monitoring Directory . Specify a content type that the Module for EDI recognizes. For more information about these content types, see “Defining the Content Type to Use” on page 132 .
Processing Service	Service that is to process the EDI document.

For general steps to configure a File Polling listener port, see the *webMethods Integration Server Administrator’s Guide* for your release.

Identifying the Service to Invoke

When using File Polling, while adding the File Polling listening port, identify the service with which you want to process the EDI document in the **Processing Service** field.

7 Creating Clients that Send EDI Documents to Trading Networks

■ Overview	142
■ Content Type to Use	142
■ Service the Client Invokes	143
■ Sending EDI Documents to Trading Networks via HTTP/S	143
■ Sending EDI Documents to Trading Networks via FTP	144
■ Sending EDI Documents to Trading Networks via File Polling	145
■ Sending EDI Documents to Trading Networks via My webMethods	146
■ Using a Service to Send Multiple EDI Documents to Trading Networks	147

Overview

If you want to process your EDI documents through webMethods Trading Networks (Trading Networks), create a client to Integration Server that submits the documents to Trading Networks. For more information about creating clients, see the chapter about creating client code in the *webMethods Service Development Help* for your release. For more information about using clients for inbound EDI document processing when you are using Trading Networks with Module for EDI, see the *webMethods Module for EDI Concepts Guide*.

The client you create must:

- Send the document to Integration Server using a content type that Module for EDI handles. For more information, see [“Content Type to Use” on page 142](#).
- Invoke the appropriate service to handle the document.
- Use HTTP, HTTPS, FTP, File Polling, or EDIINT to transport the document to Integration Server. For more information, see one of the following sections in this chapter:
 - [“Sending EDI Documents to Trading Networks via HTTP/S” on page 143](#)
 - [“Sending EDI Documents to Trading Networks via FTP” on page 144](#)
 - [“Sending EDI Documents to Trading Networks via File Polling” on page 145](#)

For more information about using EDIINT, see the *webMethods Module for EDIINT Installation and User’s Guide*.

Content Type to Use

The content type your client should use to send the EDI documents to Trading Networks depends on the type of data you are sending and/or the type of transport you want to use. You can have the client send EDI documents either with or without mainframe data using HTTP, HTTPS, FTP, or File Polling.

Content Type for EDI Documents that Do Not Contain Mainframe Data

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

Note: For backward compatibility, Module for EDI also has content handlers to accept documents with the content types application/EDI, application/X12, and application/UNEDIFACT. With these content types, the Module for EDI 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 application/EDIstream content type because it conserves system memory.

Content Types for EDI Documents that Contain 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 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 remove the...
application/x-wmediwrap80	81st character from every record in the document
application/x-wmediwrap132	133nd character from every record in the document
application/x-wmedisemiwrap80	81st character from a record only when a segment is longer than 80 characters
application/x-wmedisemiwrap132	132nd character from a record only when a segment is longer than 133 characters

When you use one of the application/x-wmedi... content types, Integration Server passes the document to the Module for EDI 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 `edidata` parameter and assigns this parameter as the pointer to `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, it should invoke the `wm.tn:receive` service to send the document directly to Trading Networks.

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

Sending EDI Documents to Trading Networks via HTTP/S

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

The requirements and logic for an HTTP client to send EDI documents to Trading Networks are almost the same as the requirements and logic described in [“Sending EDI Documents to Integration Server via HTTP” on page 136](#). The one difference is the service that the client invokes. To create a client that sends 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 Parameters 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* for your release.

Set this parameter...	Of data type...	To...
<code>url</code>	String	The URL for the <code>wm.tn:receive</code> service, for example, <code>http://rubicon:5555/invoke/wm.tn/receive</code> . In this example, the <code>wm.tn:receive</code> service on the "rubicon" server with port number "5555" would be invoked.
<code>method</code>	String	<code>post</code>
<code>loadAs</code>	String	The data type of the input data source. Specify one of the following: <ul style="list-style-type: none"> ■ <code>bytes</code>. The document data source is a <code>byte[]</code> ■ <code>stream</code>. 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 headers that you want to set. Specify a key for each header field that you want to set. The key's name represents the name of the header field, for example, <code>Content-type</code> . The key's value represents the value of that header field, 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 Trading Networks via FTP

You can create a client that sends an EDI document to 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 Integration Server via FTP” on page 138](#). 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 Parameters for the pub.client:ftp Service

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

Set this parameter...	Of data type...	To...
<i>serverhost</i>	String	The name of the machine running Integration Server.
<i>serverport</i>	String	The port on which 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 <code>wm.tn:receive</code> service, that is the service to invoke: <code>\ns\wm\tn\receive</code>
<i>localfile</i>	String	The name of the source file containing the EDI document, for example, <code>x12_850</code>
<i>remotefile</i>	String	The name to assign the file on 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>
<i>secure</i>	Document	Indicate whether the FTP session is with a secure FTP server. The <i>auth</i> parameter specifies the kind of authentication mechanism to use (SSL, TLS, or TLS-P), and the <i>securedata</i> parameter 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 Trading Networks via File Polling

You can use File Polling to send 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 Integration Server via File Polling” on page 139](#), but with 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 142](#).

Sending EDI Documents to Trading Networks via My webMethods

You can send an EDI document to Trading Networks using My webMethods if you have Submit Documents to TN general functional permission. For more information, see [“Assigning User-Specific Permissions in My webMethods” on page 22](#).

➤ To send an EDI document to Trading Networks via My webMethods

1. In My webMethods: **Administration > Integration > B2B Settings > Submit Documents**
2. For **Document Format**, select **EDI**.

Note:

This selection is available only if the WmEDIforTN package is installed.

3. Select the submission method by doing one of the following:
 - To upload the document, click **Upload file** and then click **Browse** to locate and select the document for uploading.
 - To copy the contents of a document, click **Copy text** and then paste the text into the Document Content box.
4. From the **PRT Ignore Document** list, select one of the following:
 - **True**, to not route the document to the webMethods Process Runtime module. Use this setting if the WmPRT package is not installed.
 - **False**, to route the document to the webMethods Process Runtime module. Use this setting if the WmPRT package is installed.

Module for EDI saves the setting you select in the pipeline `prtIgnoreDocument` parameter in the submitted document.

5. Click **Submit**.

Using a Service to Send Multiple EDI Documents to Trading Networks

If you create a client that obtains multiple EDI documents (for example, reading them from file) and you want to loop and send each EDI document to Trading Networks in one invocation of your client, you must be careful to cleanup the pipeline between sending each document. Use either the `wm.tn:receive` service or `wm.tn.doc.xml:routeXml` service to send the document. Then, after each invocation of either of these services, drop the `editn_env` parameter from the pipeline.

8 Handling Large Documents When Using Trading Networks

■ Overview	150
■ Configuring Module for EDI for Large Document Handling	150
■ Converting Large Documents to IData Objects Iteratively	152
■ Troubleshooting Memory Constraint Problems	152

Overview

A large EDI document is defined as either an EDI document that contains many interchanges, many groups, and many transactions, or as a transaction that contains many line items. Typically, when using webMethods Module for EDI (Module for EDI) with webMethods Trading Networks (Trading Networks), Module for EDI uses the Trading Networks large document settings.

Tip:

To simplify large document handling for all of your documents, you might choose to assign identical values to some or all of the Module for EDI and Trading Networks large document settings.

However, some services provided with the WmEDI package process large documents differently. When these services are invoked, Module for EDI uses the Module for EDI large document settings described in [“Configuring Module for EDI for Large Document Handling” on page 150](#).

The following services use the Module for EDI large document settings. You might invoke these services from a Trading Networks processing rule.

- For ANSI X12 and UN/EDIFACT documents:
 - `wm.b2b.edi:envelopeProcess`
 - `wm.b2b.edi:convertToValues`
 - `wm.b2b.edi.util:generateFA`
- For TRADACOMS and VDA documents:
 - `wm.b2b.edi.tradacoms:convertToValues`

Configuring Module for EDI for Large Document Handling

Use the following procedure to configure Module for EDI to use large document handling. The procedure updates the configuration in both the memory, so that the changes take effect immediately, and in the `WmEDI/config/properties.cnf` file.

➤ **To configure Module for EDI for large document handling**

1. Make sure the `EDIBigDocThreshold` property is set for large documents, as described in [“Defining Module for EDI Properties” on page 24](#).
2. Update Trading Networks large document settings:
 - a. From Integration Server Administrator, select **Settings > Extended**.
 - b. Click **Edit Extended Settings**, and then add or update the following Integration Server properties:

Property	Description
<code>watt.server.tspace.location</code>	Specifies the absolute directory path of the hard disk drive space in which Integration Server is to temporarily store large documents rather than keep them in memory.
<code>watt.server.tspace.max</code>	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 <code>watt.server.tspace.location</code> property. The default value is 52,428,800 bytes (50 MB).
<code>watt.server.tspace.timeToLive</code>	Specifies the amount of time in milliseconds that Integration Server temporarily stores documents in Tspace, rather than keeping them in memory. Setting this property prevents a document from being deleted as soon as it is created, thus prevents any resulting exceptions that occur while Integration Server tries to read back from the document. The default value is 0.
	Note: When batching EDI documents, you must set this property so that Integration Server can access the file from the temporary directory and store it in the database.

For more information about Integration Server properties, see the *webMethods Integration Server Administrator's Guide* for your release.

3. Click **Save Changes**.

For more information about configuring Trading Networks for large document handling, see the *webMethods Trading Networks Administrator's Guide* for your release.

Determining When an EDI Document Is Large

Module for EDI checks the configuration file during the following actions to determine if the EDI document being processed is large:

- 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 unparsed. These unparsed sections of the document are known as *undefined segments*. The `envelopeProcess` service uses the `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 `_RID_ element`. 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 (*EDIValues*) after the last identified segment.

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

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

Converting Large Documents to IData Objects Iteratively

When you are converting a large EDI document to an `IData` object, you should process the document iteratively, segment by segment. To do so, set the `iterator` input parameter of the `convertToValues` service to `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 178](#).

Troubleshooting Memory Constraint Problems

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

- Configure Module for EDI to handle large documents differently. The `EDIBigDocThreshold` EDI configuration property defines a threshold size at which Module for EDI considers a document to be large. When Module for EDI processes large documents, rather than keeping them in memory, Module for EDI temporarily saves the large documents to local hard disk drive space (known as `Tspace`). The built-in services provided with Module for EDI automatically recognize when a document is in `Tspace` and perform their processing accordingly.

For more information about configuring the `EDIBigDocThreshold` property for large document handling, see [“Defining Module for EDI Properties” on page 24](#).

- Create clients that send documents to Integration Server as `InputStreams`.
- Convert documents to `IData` objects iteratively by using the `iterator` parameter in the `wm.b2b.edi:convertToValues` service to process a document a segment or section at a time.

9 Generating Acknowledgments

■ Overview	154
■ Generating Functional Acknowledgments	154
■ Before You Can Generate a Functional Acknowledgment	156
■ Determining How Module for EDI Reports FA Status	156
■ When to Invoke the generateFA Service	163
■ Output from the generateFA Service	166
■ Tracking Late Functional Acknowledgments	167
■ Generating Interchange Acknowledgments	169

Overview

Module for EDI can generate functional acknowledgments (FAs) and interchange acknowledgments (TA1s) both manually and automatically. The following table lists what to use to generate acknowledgments and where to find more information.

To...	Use the...	As described in...
Manually generate FAs	wm.b2b.edi.util:generateFA service	“Generating Functional Acknowledgments” on page 154
Manually generate TA1s	wm.b2b.edi:generateX12TA1 service	“Generating Interchange Acknowledgments” on page 169
Automatically generate FAs	FAGeneration/autoGenerateFA EDITPA Variable	“FAGeneration Variables” on page 75
Automatically generate TA1s	X12TA1Generation/autoGenerateTA1 EDITPA Variable	“X12TA1Generation Variables” on page 93

Generating Functional Acknowledgments

A *functional acknowledgment* is a transaction set sent by the receiver of an EDI transmission to the sender, acknowledging that the message has been received and its syntax is acceptable. Functional acknowledgments do not indicate that the document has been processed by the receiver.

Note:

Functional acknowledgments (FAs) are not applicable to the TRADACOMS and VDA standards.

Module for EDI provides the wm.b2b.edi.util:generateFA service to generate functional acknowledgments (FAs). To generate an FA manually, you invoke the generateFA service from the service that you create 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 the FA as output.

The following table shows the type of FA that Module for EDI generates based on the EDI standard of the inbound document.

EDI Standard	Type of FA the Module Generates	Description
ANSI X12 UCS VICS	ANSI X12 997	The module generates an FA for each group in the inbound document.
UN/EDIFACT EANCOM	UN/EDIFACT CONTRL	The module generates an FA for each interchange in the inbound document.

EDI Standard	Type of FA the Module Generates	Description
--------------	---------------------------------	-------------

ODETTE

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

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 when the ANSI X12 997 standard is used. Refer to this table to see the related FA status that is reported when the UN/EDIFACT CONTRL standard is used.

The value that the `generateFA` service reports for the FA status depends on whether a child transaction is allowed in its envelope and the values you specify for the `syntaxErrorStatus`, `logicalErrorStatus`, and `childTransactionRejectedStatus` input parameters. For more information, see [“Determining How Module for EDI Reports FA Status” on page 156](#).

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.

For more information about...	See...
How to have Module for EDI automatically generate FAs when you are using Trading Networks	“Automatically Generating Functional Acknowledgments” on page 246
How to generate a report for FA reconciliation	“Reconciling Functional Acknowledgments” on page 303
The <code>wm.b2b.edi.util:generateFA</code> service	<i>webMethods Module for EDI Built-In Services Reference</i>

Before You Can Generate a Functional Acknowledgment

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

- Ensure that both the flat file schema for the FA and the flat file schema that defines the structure of the EDI document for which you are generating an FA are created. For more information about creating flat file schemas, see [“Defining TN EDI Document Types” on page 34](#).

The `generateFA` service uses the flat file schema to validate the EDI document.

- When generating FAs at the segment and element levels, configure the maximum number of errors to report per FA transaction. For more information, see [“Configuring the Maximum Number of Transaction Errors” on page 156](#).

Configuring the Maximum Number of Transaction Errors

When you want to generate FAs at the element or segment 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. In My webMethods: **Administration > Integration > B2B Settings > Configure Properties**.
2. In the Module for EDI Configuration Properties section, set the `EDIMaxFATransactionErrors` property as follows:

Specify this value...	To indicate...
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.

For more information about this property, see [“Defining Module for EDI Properties” on page 24](#).

3. Click **Save**.

Determining How Module for EDI Reports FA Status

The following table lists the information that the module uses to determine the FA status for a transaction, group, or UN/EDIFACT interchange of an EDI document when executing the `wm.b2b.edi.util:generateFA` service.

Payload Section**Transaction****Information Used to Determine FA Status**

- Whether the transaction is allowed in the group in which it resides. For more information, see [“Determining If the Child Transaction is Allowed in Its Envelope”](#) on page 158.
- Whether the transaction is an FA (997 or CONTRL). For more information, see [“Determining If a Transaction Is an FA or If a Group Contains an FA”](#) on page 158.
- The value of *syntaxErrorStatus*. For more information, see [“syntaxErrorStatus Parameter”](#) on page 158.
- The value of *logicalErrorStatus*. For more information, see [“logicalErrorStatus Parameter”](#) on page 159.

Group**Note:**

The generateFA service will report FA status for UN/EDIFACT groups if the UN/EDIFACT document contains groups.

- Whether the group is "FA" or "CONTRL". For more information, see [“Determining If a Transaction Is an FA or If a Group Contains an FA”](#) on page 158.
- The value of *syntaxErrorStatus*. For more information, see [“syntaxErrorStatus Parameter”](#) on page 158.
- The value of *logicalErrorStatus*. For more information, see [“logicalErrorStatus Parameter”](#) on page 159.
- The value of *childTransactionRejectedStatus*. For more information, see [“childTransactionRejectedStatus Parameter”](#) on page 160.

Interchange (UN/EDIFACT only)

- The value of *syntaxErrorStatus*. For more information, see [“syntaxErrorStatus Parameter”](#) on page 158.
- The value of *logicalErrorStatus*. For more information, see [“logicalErrorStatus Parameter”](#) on page 159.
- The value of *childTransactionRejectedStatus*. For more information, see [“childTransactionRejectedStatus Parameter”](#) on page 160.

Determining If the Child Transaction is Allowed in Its Envelope

When setting the FA status for a child transaction, the `generateFA` service first determines whether the child transaction is allowed in its group (and, therefore, its envelope). 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 values of the `syntaxErrorStatus`, `logicalErrorStatus`, and `childTransactionRejectedStatus` parameters for the child transaction.
- Not allowed, the `generateFA` service sets the FA status to "Not Allowed" and does not determine the rest of the statuses.

Determining If a Transaction Is an FA or If a Group Contains an FA

The `generateFA` service reports the FA status as "FA" when either a transaction is an FA, or a group contains an FA, as follows:

- A transaction is an FA when it is either an ANSI X12 997 or UN/EDIFACT CONTRL
- A group is an FA when it is either an ANSI X12 "FA" or a UN/EDIFACT "CONTRL" group

syntaxErrorStatus Parameter

The `syntaxErrorStatus` input parameter to the `generateFA` service indicates how you want the service to report the syntax error status. `syntaxErrorStatus` indicates whether syntax errors exist 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 values of the `syntaxErrorStatus` input parameter and their meanings:

Set *syntaxErrorStatus* to... When you...

Rejected	Want the <code>generateFA</code> service to reject the element (for example, the transaction) because of syntax errors. Reports the syntax error status as: <ul style="list-style-type: none">■ "Accepted" if there are no syntax errors■ "Rejected" if there are syntax errors
----------	---

Set *syntaxErrorStatus* to... When you...

Accepted, But Errors Were Noted
 Want to know whether there were syntax errors, but do not want to reject the element (for example, transaction) because of syntax errors.

Reports the syntax error status as:

- "Accepted" if there are no syntax errors
- "Accepted, But Errors Were Noted" if there are syntax errors

Accepted

Do not want to check for syntax errors.

Always reports the syntax error status as "Accepted" regardless of whether there are any syntax errors.

***logicalErrorStatus* Parameter**

The *logicalErrorStatus* input parameter 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 values of the *logicalErrorStatus* input parameter and their meanings:

Set *logicalErrorStatus* to... To have the *generateFA* service...

Rejected

Report the logical error status as:

- "Accepted" when there are no logical errors
- "Rejected" when there are logical errors

Use this setting when you want to reject the element (for example, transaction) because of logical errors.

Accepted, But Errors Were Noted

Report the logical error status as:

- "Accepted" if there are no logical errors
- "Accepted, But Errors Were Noted" when there are logical errors

Use this setting when you want to know whether there were logical errors but do not want to reject the element (for example, transaction) because of logical errors.

Set *logicalErrorStatus* to... To have the generateFA service...

Accepted	Always reports the logical error status as "Accepted" regardless of whether there are any logical errors.
	Use this setting when you do not want to check for logical errors.

***childTransactionRejectedStatus* Parameter**

The *childTransactionRejectedStatus* input parameter 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"

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

When the FA status of...	And the <i>childTransactionRejectedStatus</i> is set to...	The generateFA service sets the child transaction rejected status to...
Every child transaction is Accepted	Rejected	Accepted
	Partially Accepted	Accepted
	Accepted, But Errors Were Noted	Accepted
At least one child transaction is Accepted, But Errors Were Noted -AND- no child transactions are Accepted	Rejected	Rejected
	Partially Accepted	Accepted, But Errors Were Noted
	Accepted, But Errors Were Noted	Accepted, But Errors Were Noted
At least one child transaction is "Accepted, But Errors Were Noted -AND- at least one child transaction is Accepted"	Rejected	Rejected
	Partially Accepted	Partially Accepted
	Accepted, But Errors Were Noted	Accepted, But Errors Were Noted

When the FA status of...	And the childTransactionRejectedStatus is set to...	The generateFA service sets the child transaction rejected status to...
Every child transaction is Rejected	Rejected	Rejected
	Partially Accepted	Rejected
	Accepted, But Errors Were Noted	Rejected

How Module for EDI 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 determines 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 CONTRL). For more information about transactions that are not allowed, see [“Determining If a Transaction Is an FA or If a Group Contains an FA” on page 158](#).
- Syntax error status. For more information about how the module sets the syntax error status, see [“syntaxErrorStatus Parameter” on page 158](#).
- Logical error status. For more information about how the module sets the logical error status, see [“logicalErrorStatus Parameter” on page 159](#).

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.

Module for EDI 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	No	Yes	Any syntax error status	Any logical error status
FA	Yes	Yes	Any syntax error status	Any logical error status
Rejected	Yes	No	At least one status is Rejected	

Module for EDI 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	Yes	No	Statuses are either Accepted, But Errors Were Noted OR Accepted, AND at least one status is Accepted, But Errors Were Noted	
Accepted	Yes	No	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". For more information, see [“Determining If a Transaction Is an FA or If a Group Contains an FA” on page 158](#).
- Syntax error status. For more information about how the module sets the syntax error status, see [“syntaxErrorStatus Parameter” on page 158](#).
- Logical error status. For more information about how the module sets the logical error status, see [“logicalErrorStatus Parameter” on page 159](#).
- Child transaction rejected status. For more information about how the module sets the child transaction rejected error status, see [“childTransactionRejectedStatus Parameter” on page 160](#).

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.

Module for EDI reports this FA status for a group...	When the group has the following statuses...			
	Is group either "FA" or "CONTRL"?	Syntax Error Status	Logical Error Status	Child Transaction Rejected Status
FA	Yes	Any syntax error status	Any logical error status	Any child transaction rejected status
Rejected	No	At least one status is Rejected		
Partially Accepted	No	Accepted, But Errors Were Noted OR Accepted	Accepted, But Errors Were Noted OR Accepted	Partially Accepted
Accepted, But Errors Were Noted	No	Accepted, But Errors Were Noted OR Accepted, AND at least one status is Accepted, But Errors Were Noted		

Module for EDI reports this FA status for a group...	When the group has the following statuses...			
	Is group either "FA" or "CONTRL"?	Syntax Error Status	Logical Error Status	Child Transaction Rejected Status
Accepted	No	Accepted	Accepted	Accepted

UN/EDIFACT Interchange FA Status

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

- Syntax error status. For more information about how the module sets the syntax error status, see “[logicalErrorStatus Parameter](#)” on page 159.
- Logical error status. For more information about how the module sets the logical error status, see “[logicalErrorStatus Parameter](#)” on page 159.
- Child transaction rejected status. For more information about how the module sets the child transaction rejected error status, see “[childTransactionRejectedStatus Parameter](#)” on page 160.

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.

Module for EDI 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 OR Accepted	Accepted, But Errors Were Noted OR Accepted	Partially Accepted
Accepted, But Errors Were Noted	Statuses are either Accepted, But Errors Were Noted OR Accepted, AND 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

When to Invoke the generateFA Service

You invoke the `wm.b2b.edi.util.generateFA` service from the service you created to process an inbound EDI document. At what time during processing you wish to invoke the generateFA service depends on whether the service you created processes the EDI document all at once or iteratively, segment

by segment. For more information about these two options, see [“Determining Which Process Method to Use” on page 173](#).

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

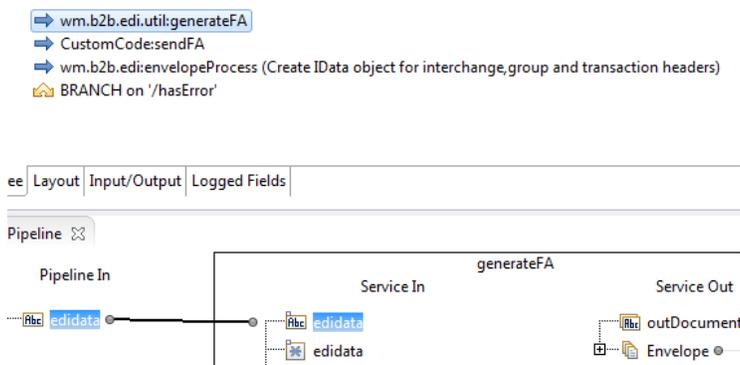
Using a Service that Processes Documents at One Time

When you use a service that processes the entire 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 more information about processing the entire document at one time, see [“Processing the Entire Document at One Time” on page 174](#).

Invoking `generateFA` Before Invoking `envelopeProcess`

When you invoke the `generateFA` service before the `envelopeProcess` service, the service uses the unparsed EDI document as input. The unparsed document is in the pipeline in the `edidata` parameter. The `edidata` parameter is placed in the pipeline by an EDI content handler. For more information about the `edidata` parameter, see [“Inputs to Your Service” on page 172](#).

The following diagram shows an updated version of the service illustrated in [“Processing the Entire Document at One Time” on page 174](#), which includes invocation of the `generateFA` service. Note that in the **Pipeline** section of the screen, the `edidata` parameter in **Pipeline In** is mapped to the `edidata` in **Service In**.



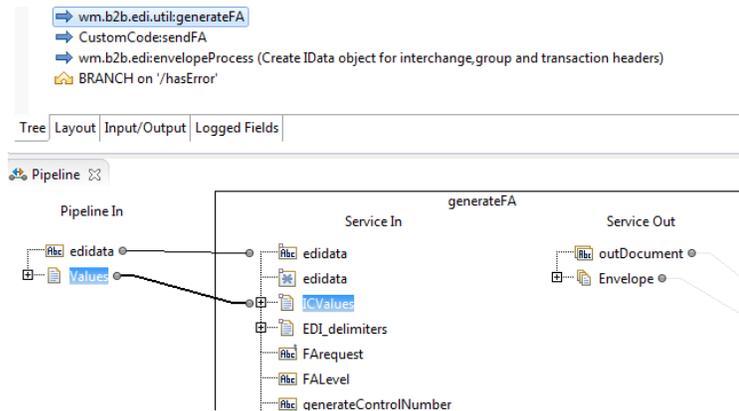
After invoking the `generateFA` service to generate the FA, add logic or invoke a service that you create to deliver the FA to the sender of the original EDI document.

Invoking `generateFA` Immediately after Invoking `envelopeProcess`

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

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, that is, the error array must be included in the results.

The following diagram shows an updated version of the service that was illustrated in [“Processing the Entire Document at One Time” on page 174](#) which includes the invocation of the generateFA service. In the **Pipeline** section of the screen, the *Values* parameter in **Pipeline In** is the output from the envelopeProcess service. The *ICValues* parameter in **Service In** is the input parameter to the generateFA service that accepts a parsed EDI document. In the **Pipeline** section of the screen, map the *Values* parameter to the *ICValues* parameter, as shown.



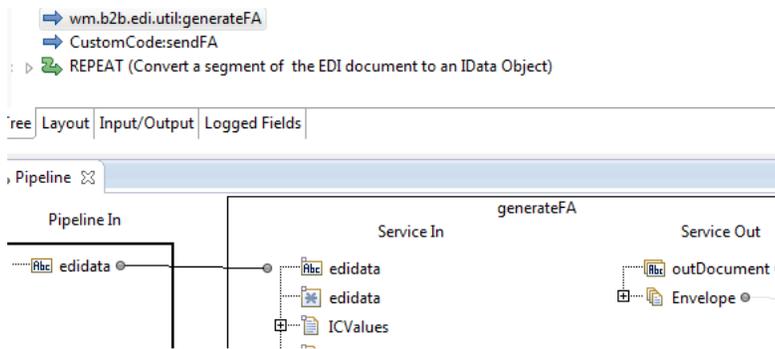
After invoking the generateFA service to generate the FA, add logic or invoke a service that you create to deliver the FA to the sender of the original EDI document.

Using a Service that Processes Documents Iteratively

When you use a service that processes the EDI document iteratively, segment by segment, invoke the generateFA service before performing other processing. For more information about applying logic to a service that processes documents iteratively, see [“Processing the Document Iteratively, Segment by Segment” on page 178](#).

When you invoke the generateFA service, the service uses the unparsed EDI document as input. The unparsed document is in the pipeline in the *edidata* parameter. The *edidata* parameter is placed in the pipeline by an EDI content handler. For more information about the *edidata* parameter, see [“Inputs to Your Service” on page 172](#).

The following diagram shows an updated version of the service illustrated in [“Processing the Document Iteratively, Segment by Segment” on page 178](#), which includes invocation of the generateFA service. Note that in the **Pipeline** section of the screen, the *edidata* parameter in **Pipeline In** is mapped to the *edidata* parameter in **Service In**.



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.

Output from the `generateFA` Service

The output from the `wm.b2b.edi.util:generateFA` service is the FA itself. The following table summarizes the information in the generated FA, including items that Module for EDI sets in the:

- BizDocEnvelope that the 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).

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
	EDI FA Outbound custom attribute	true
Interchange header	sender	The receiver from the inbound document's interchange header
	receiver	The sender from the inbound document's interchange header

Item	Setting	Value
	control number	The value is based on the <i>FAGeneration/generateControlNumber</i> EDITPA variable
Group	sender	The receiver from the inbound document's group header
	receiver	The sender from the inbound document's group header
	control number	The value is based on the <i>FAGeneration/generateControlNumber</i> EDITPA variable
Transaction header	control number	001

Because the `generateFA` service does not specify what to do with the FA that it has created, 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 codes listed in the table of possible FA statuses in [“Generating Functional Acknowledgments” on page 154](#).

Tracking Late Functional Acknowledgments

Tracking late functional acknowledgments based on sender, receiver, and document type of the transactions can help you analyze the documents that are acknowledged late by your partners.

Module for EDI supports the tracking of late FAs for a partner pair or at a group level of a partner pair.

- For partner pair level tracking, the module supports tracking late functional acknowledgments of EANCOM, ODETTE, UCS, UNEDIFACT, VICS, and X12 standard documents.
- For group level tracking for a partner pair, the module supports tracking of group level late functional acknowledgments of X12 standard documents and X12 sub-standard documents.

Before Tracking Late FAs

Before tracking late FAs, you must define the *LateFA* variables in the partner-specific EDITPAs in the Trading Partner Agreement Details screen, as follows:

- In the EDITPA where your partner is the receiver, specify the following EDITPA variables:

- *LateFA/LateFATime* variable

Specify the time in minutes by which Trading Networks must receive the FA for a partner pair.

- *LateFA/LateFAForX12Group* variables:

- *LateFA/LateFAForX12Group/groupType* variable

Specify the groups used for tracking late FAs.

- *LateFA/LateFAForX12Group/LateFATime* variable

Specify the time in minutes by which Trading Networks must receive the FA.

For more information about these variables, see [“Defining Trading Partner Information” on page 53](#).

- In the EDITPA where you are the receiver, set the *FAReconciliation* EDITPA variable to true so that Module for EDI can map the late FAs to the original document and track the late FAs. If the value of the *FAReconciliation* EDITPA variable is false, the module cannot track the late FAs.

Tracking Late FAs

Once you have edited the *LateFA* variables in the partner-specific EDITPAs, you can track Late FAs using the following procedure.

> To track late FAs

1. In My webMethods: **Monitoring > Integration > B2B > Transactions**.
2. Select the transaction for which you want to track late FAs.
3. In the Transaction Details section of the Transactions page, view the status of the **Late FA** attribute for the transaction you selected. If the **Late FA** attribute is set to Y, then the corresponding document has received a late functional acknowledgment.

For more information about managing and tracking your documents, see the *webMethods Trading Networks User’s Guide* for your release.

Example

In this example, consider that partner A is the sender and partner B is the receiver. Partner A wants to track the late FAs for PO and IN groups of X12 standard documents.

Partner A must:

1. Set the *LateFATime* variable for both the *LateFA/LateFATime* and the *LateFA/LateFAForX12Group* partner-specific EDITPA variables

2. Set the *FAReconciliation* EDITPA variable to true in the EDITPA where partner A is the receiver, so that the module can map the late FAs to the original document and track the late FAs

Partner A sets the *LateFA/LateFATime* and the *LateFA/LateFAForX12Group* partner-specific EDITPA variables as follows:

For the *LateFA/LateFATime* EDITPA variable:

LateFATime Value

1

For the *LateFA/LateFAForX12Group* EDITPA variable:

<u>Group Type</u>	<u>LateFATime Value</u>
PO	2
IN	3

When partner A sends X12 documents to partner B, the module sets the Late FA attribute of the original document in Trading Networks to Y when any of the following conditions occur:

- Partner A receives the functional acknowledgment for the PO group more than two minutes later.
- Partner A receives the functional acknowledgment for the IN group more than three minutes later.
- Partner A receives the functional acknowledgments for any other group more than one minute later.

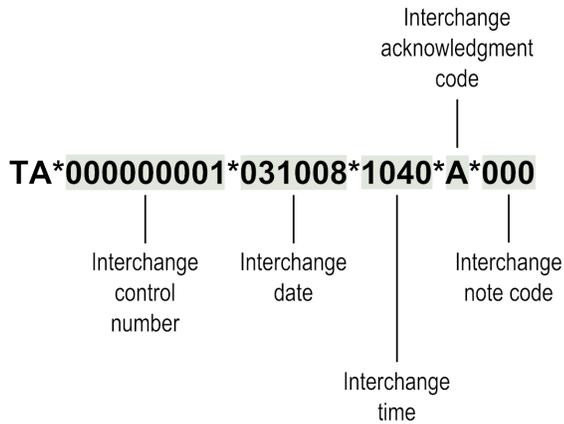
Module for EDI sets the Late FA attribute of the original document in Trading Networks to N when the FA is not late.

Generating Interchange Acknowledgments

When an X12 interchange contains a control header and trailer envelope that surround one or more functional groups, an *Interchange Acknowledgment* (TA1) notifies the sender whether the receiver successfully processed the interchange header and trailer.

The TA1 does not report the status of the functional groups and transaction sets within the interchange envelope.

The format of a TA1 looks as follows:



Field	Description
<i>Interchange control number</i>	Uniquely identifies the interchange. The sender assigns the interchange control number, which, along with the sender ID, uniquely identifies the interchange data to the recipient.
<i>Interchange date</i>	Indicates the date, in YYMMDD format, when the interchange was prepared. For example, if the interchange was prepared on January 2, 2014, the interchange date would be 140102.
<i>Interchange time</i>	Indicates the time, in 24-hour format (HHMM), when the interchange was prepared.
<i>Interchange acknowledgment code</i>	Indicates the Interchange Acknowledgment Code, as agreed upon by trading partners. Valid values are: <ul style="list-style-type: none"> ■ A—Accepted ■ R—Rejected ■ E—Accepted, but the file contains errors and must be resubmitted.
<i>TA1 code</i>	A three-digit number that corresponds to a TA1 code. For more information about TA1 codes, see “TA1 Status and TA1 Code” on page 195 .

Module for EDI provides a built-in service, `wm.b2b.edi:generateX12TA1`, to generate a TA1 acknowledgment. While generating the TA1 acknowledgment, this service provides the option to include a functional acknowledgment, as well. To generate a TA1 with or without an FA, you invoke the `generateX12TA1` service from the service that you create to process an EDI document.

For more information about the `wm.b2b.edi:generateX12TA1` service, see *webMethods Module for EDI Built-In Services Reference*.

10 Before Creating the Service to Process EDI Documents

■ Overview	172
■ Inputs to Your Service	172
■ Logic to Include in the Service to Process EDI Documents	172
■ Samples	182

Overview

The WmEDI package of Module for EDI provides built-in services that support functions required by the EDI standard. When you create the service that processes your EDI documents, you add logic to invoke the Module for EDI built-in services. This chapter contains information about how to create the service to process an EDI document. For information about how to process an inbound EDI document, see the *webMethods Module for EDI Concepts Guide*.

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

- The flat file schema that defines the structure of the EDI document to process. Module for EDI uses the flat file schema for parsing, converting, and validating an inbound EDI document.
- 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 (for example, a back-end system). Module for EDI provides a service that you can have your service invoke to create your internal-format document based on the flat file schema. Use Software AG Designer to create the flat file schema. For more information, see *Flat File Schema Developer's Guide*.

Inputs to Your Service

The service you create should accept the *edidata* parameter that the Module for EDI content handler placed in the pipeline as input. The data type of the *edidata* parameter 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 [“Defining the Content Type to Use” on page 132](#).

Logic to Include in the Service to Process EDI Documents

You can perform any processing on the EDI document that you want. For example, you can:

- Map information from the EDI document into one or more internal-format documents. The internal-format is the format that an internal application (for example, 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 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 generate and send functional acknowledgments (FAs) for EDI documents. For more information, see [“Generating Acknowledgments” on page 153](#).

Determining Which Process Method to Use

There are two basic ways to process documents:

- Process the entire document at one time. With this method, you invoke services that consume the entire EDI document at one time and then you can process it. Use this method 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). For more information about handling large documents, see [“Handling Large Documents When Using Trading Networks” on page 149](#).

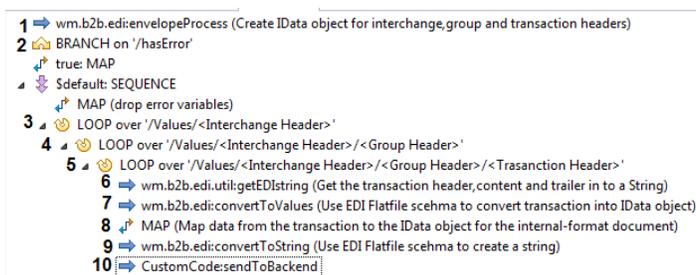
The table below lists the advantages and disadvantages of each method:

Processing Method	Advantages	Disadvantages
Processing the entire document at one time	<ul style="list-style-type: none"> ■ Easier to implement because most of the processing and set up you need to do is provided with Module for EDI. ■ You can use built-in services to perform envelope validation and compliance checks. For more information about envelope validation and compliance checks, see “Performing Envelope Validation and Compliance Checks” on page 177. 	<ul style="list-style-type: none"> ■ Requires more memory because an IData object is created for the entire document.
Processing the document segment by segment	<ul style="list-style-type: none"> ■ Requires less memory because as the segments are being processed, an IData object is created for only the segment or group of segments that represent repeating content (for example, line items). Then, when working on subsequent segments, the memory is reused. 	<ul style="list-style-type: none"> ■ If you want to perform envelope validation and compliance checks, you must add your own logic to do so. ■ You likely will need to modify the flat file schema for the EDI document you are processing.

Processing Method	Advantages	Disadvantages
		<ul style="list-style-type: none"> You might need to create additional flat file schemas for segments within the EDI document you are processing.

Processing the Entire Document at One Time

The following diagram shows sample code that includes the basic logic you would apply to process an entire inbound EDI document at one time. This processing shows how to map information from the EDI document to an internal-format document and then send the document to an internal application, for example, a back-end system. For information about all of the built-in services that the sample uses, see *webMethods Module for EDI Built-In Services Reference*.



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 <i>IData</i> object named <i>Values</i>. That is:</p> <ul style="list-style-type: none"> For an ANSI X12 document, it creates an <i>IData</i> object of the ISA/IEA, GS/GE and ST/SE headers/trailers. For a UN/EDIFACT document, it creates an <i>IData</i> object of the UNB/UNZ, UNG/UNE, and UNH/UNT headers/trailers. |
|---|---|

After the header segments are converted to the *Values* *IData* object, your service can access and act on data elements within the header segments of the EDI document.

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 more information, see [“Handling Large Documents When Using Trading Networks”](#) on page 149.

Flow Operation	Description
	<ul style="list-style-type: none"> ■ For documents not considered large, the service leaves the transaction set contents in String format. The service assigns the transaction set body to the <i>unDefData</i> parameter 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 parameter 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 element <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/UNB/UNG/UNH/_RID_</i>. For more information about how Module for EDI handles large documents, see “Handling Large Documents When Using Trading Networks” on page 149. <p>Additionally, the <i>envelopeProcess</i> 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 parameter for the <i>envelopeProcess</i> service. For more information about the compliance check, see “Performing Envelope Validation and Compliance Checks” on page 177.</p>
2	Add your own logic to handle errors that might result from executing the <i>wm.b2b.edi:envelopeProcess</i> 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 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/UNB</i>.
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 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 slightly different. For sample code, see the <i>sampleServices:UNEDIFACTToValues</i> in the <i>WmEDIsamples</i> package, which is located under Code Samples in the Technical community area of the Empower Product Support website at https://</p>

Flow Operation	Description
	empower.softwareag.com . The samples in this folder have been certified by Software AG.
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.

Flow Operation	Description
6	<p>Invoke the <code>wm.b2b.edi.util:getEDLstring</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 <code>wm.b2b.edi:convertToValues</code> service. 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 trailers. If you input data without the header and trailer, the <code>convertToValues</code> service will return errors.</p>
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:getEDLstring</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 to the internal-format document.</p> <p>Now that the content of the transaction set 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.</p>

Flow Operation	Description
9	<p>Invoke the <code>wm.b2b.edi:convertToString</code> service to convert the internal-format document from an <code>IData</code> object to <code>String</code> 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, the <code>convertToString</code> service can accept 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, for example, a back-end system.</p>

Performing Envelope Validation and Compliance Checks

While processing an envelope, the `wm.b2b.edi:envelopeProcess` service optionally can validate the envelope against the predefined flat file schema for non-TRADACOMS documents. The service also can perform compliance checks against the interchanges.

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

Set this input parameter to true...	To have the <code>wm.b2b.edi:envelopeProcess</code> service...
<i>validate</i>	<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 <code>errorArray</code> output parameter.</p> <p>The interchange envelopes are validated against the EDI flat file schema for the EDI document. Module for EDI 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>
<i>complianceCheck</i>	<p>Check for matching interchange control numbers, matching group control numbers, matching transaction control numbers, segment counts, transaction counts, and group counts. After the first error is detected, the error service stops executing.</p>

If you set both the `complianceCheck` and `validate` parameters to true, the `envelopeProcess` service performs the interchange envelope validation first, followed by 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 either a single segment or a group of segments at one time. To process segment by segment, when you invoke the `wm.b2b.edi:convertToValues` service to convert data from String to an IData object, set the `iterator` input parameter to `true`. This 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

The following procedure describes the high-level steps required to customize the EDI flat file schema for iterative processing using Designer. For more detailed information about managing flat file schemas, see the *Flat File Schema Developer's Guide*.

Step	Description
------	-------------

- | | |
|---|---|
| 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 <code>convertToValues</code> service to return segments. |
|---|---|

When you execute the `convertToValues` service with the `iterator` input parameter set to `true`, the `convertToValues` service uses the flat file structure to determine the number of segments to process into an IData object. It starts processing at a top-level element and processes all of its children into an IData object in one invocation. For example, if you want 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
```

Step Description

```
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 want 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	.
.	.
.	.
.	.

- 4 During execution of 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 in an element named `_RID_` if the document is considered large. For more information about large documents, see [“Handling Large Documents When Using Trading Networks”](#) on page 149.

```
Sample Returned IData object
EDIValues
ST
  FA1
  FA2
unDefData
TDS
```

Step	Description
------	-------------

5	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 section describes the sample code for processing an inbound EDI document iteratively. The steps for processing show how to map information from the line item segments of an ANSI X12 810 document into an internal-format document, and then send the document to an internal application, for example, a back-end system. For more information about all of the built-in services that the sample uses, see *webMethods Module for EDI Built-In Services Reference*.

```

1 REPEAT ( Convert a segment of the EDIData to IData object)
2   wm.b2b.edi:convertToValues ( Convert a segment of the EDIData to IData object)
3   BRANCH on '/EDIVValues/ISA' (Perform logic under this branch if segment is ISA envelope)
4   BRANCH on '/EDIVValues/GS' (Perform logic under this branch if segment is GS)
5   BRANCH on '/EDIVValues/ST' (Perform logic under this branch if segment is a transaction set)
6     $null: SEQUENCE (If not ST, then check next branch that is branch on /EDIVValues/GE)
7     $default: SEQUENCE (If it is ST, perform this sequence)
8     4 MAP (Save ediObject and EDIFFSchema used for outer Repeat)
9     MAP (map in EDIFFSchema used for inner Repeat)
10    REPEAT (Loop over the repeating segment)
11    6   wm.b2b.edi.util:EDIconcat (Prepare unparsed data for convertToValues)
12    7   wm.b2b.edi:convertToValues (Convert segment left unparsed on 1st pass to IData object)
13    8   MAP (Map data from line item to the IData object of the internal format document)
14    9   wm.b2b.edi:convertToString (use flatfile schema for internal format document to create a String)
15    10  CustomCode:sendToBackend
16    BRANCH on '/ediObject' (Leave the Repeat loop when ediObject is null)
17    $null: SEQUENCE
18    11 MAP (Restore values of ediObject and EDIFFSchema to those needed by Outer Repeat)
19  BRANCH on '/EDIVValues/GE' (Perform logic under this branch for GE)
20  BRANCH on '/EDIVValues/IEA' (Perform logic under this branch for IEA)
21  BRANCH on '/ediObject' (Leave the outer repeat when ediObject is null)

```

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 <i>iterator</i> input parameter as true to the <code>convertToValues</code> service, the output contains the object <i>ediObject</i> , which is used to keep track of the input data segments. When the entire EDI document has been processed, <i>EDIObject</i> becomes null.
2	In the outer REPEAT loop, invoke the <code>convertToValues</code> service to: <ul style="list-style-type: none"> Convert a segment or group of segments to an IData object Validate the EDI structure

Flow operation	Description
	<p>The inputs to the <code>convertToValues</code> service include the <code>String</code> or <code>InputStream</code> that represents the EDI document and the customized flat file schema for the EDI document. The <code>convertToValues</code> service uses the flat file schema to determine how many segments to read. It returns the object <code>ediObject</code>, which keeps track of its location in the EDI document. This object is input to subsequent invocations of the <code>convertToValues</code> 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 <code>convertToValues</code> service to return.</p> <p>For example, the first BRANCH under the call to the <code>convertToValues</code> service checks for an <code>ISA</code> element in the returned <code>IData</code> object, <code>EDIVValues</code>. If <code>EDIVValues/ISA</code> exists and is not null, the flow service executes the logic under the BRANCH on 'EDIVValues/ISA' flow operation.</p>
4	<p>Prepare to invoke the <code>convertToValues</code> 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 <code>ediObject</code> and <code>EDIFFSchema</code> that the <code>convertToValues</code> 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 <code>EDIFFSchema</code> parameter. Make sure the value of <code>ediObject</code> is null.
5	<p>Repeatedly invoke the <code>convertToValues</code> service in this inner REPEAT loop to process an unparsed section of the document.</p>
6	<p>Invoke the <code>wm.b2b.edi.util:EDlconcat</code> service to prepare the unparsed section for the call to the <code>convertToValues</code> service.</p> <p>The <code>EDlconcat</code> service automatically gets its input from either the <code>unDefData</code> or <code>_RID_</code> elements that the first pass of the <code>convertToValues</code> service left in the pipeline for the unparsed sections. You can use the <code>EDlconcat</code> service to add a header, a trailer, or both a header and trailer to the unparsed data, if needed.</p>
7	<p>In the inner REPEAT loop, invoke the <code>convertToValues</code> service to:</p> <ul style="list-style-type: none"> ■ Convert the unparsed section to an <code>IData</code> object ■ Validate the EDI structure <p>Pass the output from the <code>EDlconcat</code> service (<code>output</code> parameter) to the <code>edidata</code> input parameter of the <code>convertToValues</code> service. In a previous flow operation, you set</p>

Flow operation	Description
	the value of the <i>EDIFFSchema</i> input parameter to the fully-qualified name of the flat file schema to use for this unparsed section of the EDI document.
8	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.
9	Invoke the <code>convertToString</code> service to convert the internal-format document from an <code>IData</code> object to <code>String</code> format. 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, the <code>convertToString</code> service can accept an <code>IS</code> document type to define the structure of the internal-format document.
10	Add your own logic or invoke a service that you create to send the internal-format document to your internal application, for example, a back-end system.
11	Restore the values of the <i>ediObject</i> and <i>EDIFFSchema</i> parameters, which you saved in a previous flow operation, in preparation for invoking the <code>convertToValues</code> service in the outer REPEAT loop.

Note: If you want to perform the interchange (or transmission) envelope validation and the compliance check, you need to add logic to your service to do so. For more information, see [“Performing Envelope Validation and Compliance Checks” on page 177](#).

Samples

For samples, see the flow services in the `sampleServices` folder of the `WmEDIsamples` package, which is located under Code Samples in the Technical community section of the Empower Product Support website at <https://empower.softwareag.com>. The samples in this folder have been certified by Software AG.

- 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 iteratively, segment by segment.

Additionally, the Tutorial.EDItoXML:processEDI850_4010 service shows a complete sample that converts an ANSI X12 850 document to an IData object and maps the data from the EDI document to an XML document.

Important:

Delete the WmEDIsamples package before going into production.

11 Preparing to Process Inbound Non-TRADACOMS Documents Using Trading Networks

■ Overview	186
■ Preparing to Receive EDI Documents	187
■ Specifying EDITPA Variables that Affect Inbound Processing	188
■ Trading Networks Attributes and EDI Documents	189
■ Defining Processing Rules to Process Inbound EDI Documents	198
■ Coding Services to Process Transaction and Group Documents	202

Overview

This chapter describes how to set up basic inbound processing when using Trading Networks for all EDI standards except TRADACOMS.

For more information about...	See...
Inbound processing using TRADACOMS	“Processing Inbound TRADACOMS Documents Using Trading Networks” on page 211
Optional inbound processing	“Optional Inbound Processing When Using Trading Networks” on page 229
Processing EDI documents using Module for EDI decoupled from Trading Networks	“Using Module for EDI Decoupled from Trading Networks” on page 355

When webMethods Trading Networks (Trading Networks) receives an EDI document, it passes it to an EDI recognizer, which performs the initial processing. The EDI recognizer is installed in Trading Networks when you install webMethods Module for EDI (Module for EDI). You can use EDITPA variables to tailor how the EDI recognizer processes an inbound EDI document. For more information, see [“Specifying EDITPA Variables that Affect Inbound Processing” on page 188](#).

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 setting of the *splitOption* EDITPA variable. The EDI recognizer can split the original EDI document into an Interchange, Group, and/or Transaction document. After creating the document(s), the EDI recognizer sends the document(s) to Trading Networks for normal processing.

You define the processing that Trading Networks performs against the Interchange, Group, and Transaction documents by defining processing rules. Processing rules consist of:

- Criteria that Trading Networks uses to select the appropriate processing rule for a 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 that Trading Networks selects the correct processing rule for your document. For example, you can set up criteria so that Trading Networks selects a processing rule only if it is from a specific sender/receiver pair and it is in a specific processing mode, such as testing or production. For more information about the document attributes that are available, see [“Trading Networks Attributes and EDI Documents” on page 189](#).
- Actions that Trading Networks is to take against the document. 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 198](#).

For more information about processing EDI documents and business processes, including an illustration of how EDI documents are passed to a business process, see *webMethods Module for EDI Concepts Guide*.

For more information about Trading Networks document attributes and processing rules, see the *Trading Networks Concepts Guide* and the *webMethods Trading Networks Administrator's Guide* for your release.

Preparing to Receive EDI Documents

The following steps must be performed before you receive EDI documents.

- | Step | Description |
|------|---|
| 1 | <p>Define the TN document types for the EDI documents that you want to process. For instructions, see “Defining TN EDI Document Types” on page 34.</p> <p>When you define the TN document types for EDI documents, Module for EDI creates the flat file schemas that define the structure of the EDI documents. Module for EDI uses the flat file schemas for parsing, converting, and validating the structure of inbound EDI documents.</p> |
| 2 | <p>Optional. Create the flat file schema that defines the structure of an internal-format document. Use Designer to create the flat file schema. For more information about creating a flat file schema, see the <i>Flat File Schema Developer's Guide</i>.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note:
You use the flat file schema created in this step if the service that you create to process an Interchange, Group, or Transaction document creates an internal-format document. Module for EDI provides a service that you can invoke to create the internal-format document based on the flat file schema.</p> </div> |
| 3 | <p>Define profiles for the senders and receivers in the inbound EDI document. For more information about how to create profiles, see “Defining Trading Partner Profiles” on page 55 and the <i>webMethods Trading Networks Administrator's Guide</i> for your release.</p> |
| 4 | <p>Define the default EDITPA and optionally, the partner-specific EDITPAs for the sender/receiver pairs in the EDI document. For more information about creating EDITPAs, see “Defining EDI Trading Partner Agreements” on page 59. For information about the EDITPA variables that affect inbound processing, see “Specifying EDITPA Variables that Affect Inbound Processing” on page 188.</p> |
| 5 | <p>Non-standard When using non-standard processing, you must also define the interchange sender/receiver pair information. For instructions, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 335 in “Using Module for EDI Decoupled from Trading Networks” on page 355. For more information about the difference between standard and non-standard processing, see <i>webMethods Module for EDI Concepts Guide</i>.</p> |

Specifying EDITPA Variables that Affect Inbound Processing

To tailor how you want Module for EDI to perform inbound processing, you must edit the EDITPA variables. When using EDI standards other than TRADACOMS, the module processes an inbound EDI document one interchange segment at a time. For each interchange segment in an inbound EDI document, the module obtains the EDITPA values to use for the interchange sender/receiver pair. The module uses these 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 62.

Note: **Non-standard** When using non-standard processing, Module for EDI uses EDITPAs for group-level sender/receiver pairs and the additional settings that you define from the module home page. For more information, see “[Variables that Affect Inbound Processing](#)” on page 352. For more information about the difference between standard and non-standard processing, see *webMethods Module for EDI Concepts Guide*.

Validating the EDIFACT Document Schema

Module for EDI determines the syntax version number, the syntax release number, and the directory version number of the incoming EDIFACT document and validates the envelope using the most specific schema available. The name of the most specific schema contains the syntax version number, the syntax release number, and the directory version number. The name of the least specific schema contains only the syntax version number.

The module searches for the appropriate flat file schema to use for the document validation of each incoming EDIFACT document in the following search order priority:

1. EDIFFSchema.*Standard.envelope:VSyntaxVersionNumber_DirectoryVersionNumber_SyntaxReleaseNumber* schema
2. EDIFFSchema.*Standard.envelope:VSyntaxVersionNumber_DirectoryVersionNumber* schema
3. EDIFFSchema.*Standard.envelope:VSyntaxVersionNumber* schema
4. `wm.b2b.editn.rec:EDIHEADERS` (default document type)

Module for EDI uses the most specific schema that exists. If the module does not find any of the listed schemas, the module uses the default `wm.b2b.editn.rec:EDIHEADERS` document type. Only the `wm.b2b.editn.rec:EDIHEADERS` document type is available when you install the module. You must install the more specific EDIFACT schemas.

Validating VDA Message Structure

Module for EDI can validate the message structure of a VDA document. The VDA schema includes a trailer record that contains dedicated fields for each type of record in the schema. These dedicated fields should contain the number of occurrences of the corresponding record type. The module validates that the value of the dedicated field matches the number of occurrences of the

corresponding record type. All control number fields are mandatory, and the module uses each field to check the completeness of a message.

You activate message structure validation for inbound VDA documents by setting the `ControlNumberManagement/VDA/validateStructure` EDITPA variable to true. For more information about this variable, see [“wm.b2b.editn.TPA:EDITPA IS Document Type” on page 62](#).

Trading Networks Attributes and EDI Documents

Trading Networks attributes specify the content from a document to use for later processing. For example, you can define the processing rule criteria to select which processing rule is executed based on the value of an attribute.

Trading Networks supports two types of attributes:

- System attributes, which Trading Networks defines.
- Custom attributes, which are additional attributes that are added to Trading Networks. Module for EDI provides some custom attributes. Additionally, you can write code to define your own custom attributes and assign them values.

At run time, Module for EDI uses information from the EDI document being processed to set values for some of the system attributes and custom attributes.

You can use 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.

System Attributes that Module for EDI Sets

Module for EDI sets the **DocumentID**, **GroupID**, and **ConversationID** system attributes using information from the EDI document. Module for EDI uses different values based on whether it is assigning the attribute to an Interchange, Group, or Transaction document. The sections below describe how the module sets each attribute for each type of document (interchange, group, and transaction).

The examples provided are based on the following sample:

Control numbers in ANSI X12 ISA, GS, and ST headers

```
ISA*00*  *00*      *01*123456789  *ZZ*987654  *030423*1810*U*00200*000005334*
GS*PO*186704136*138183702*030423*1810*1*X*003040\
ST*850*0003
```

DocumentID

The **DocumentID** system attribute is an identifier of the document. Module for EDI sets the **DocumentID** as follows:

Type of Document	Value Used for DocumentID
Interchange	The interchange control number. For example, the sample's DocumentID would be set to 000005334, which is the value of ISA13.
Group	The group control number. For example, the sample's DocumentID would be set to 1, which is the value of GS06.
Transaction	The transaction control number. For example, the sample's DocumentID would be set to 0001, which is the value of ST02. For VDA documents, Module for EDI sets the DocumentID to the value of the new control number field.

GroupID

Module for EDI hierarchically assigns the **GroupID** system attribute so that by viewing the **GroupID**, you can determine the envelope to which an element originally belonged. The module sets the **GroupID** as follows:

Type of Document	Value Used for GroupID
Interchange	The interchange control number. For example, the sample's GroupID would be set to 000005334, which is the value of ISA13.
Group	The interchange control number, so that you can determine to which interchange the Group document belongs. For example, the sample's GroupID would be set to 000005334, which is the value of ISA13.
Transaction	The group control number, so that you can determine to which group the Transaction document belongs. For example, the sample's 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 use the same **ConversationID**.

Trading Networks can extract **ConversationIDs** from EDI Envelope, Group, and Transaction documents and use them to pass documents to webMethods Process Engine after Trading Networks performs the actions identified by a processing rule. Process Engine is the facility of Integration Server that executes and manages business processes. For more information, see the *webMethods Trading Networks Administrator's Guide* for your release.

For more information about assigning conversation IDs to EDI documents and including EDI documents in business processes, see ["Including Documents in a Business Process"](#) on page 323.

Enabling Extraction of ConversationIDs from an EDI Document

By default, Trading Networks does not extract the **ConversationID** from EDI documents. Use the following procedure to enable the extraction of **ConversationIDs** from EDI documents so that Trading Networks can pass them to Process Engine.

➤ To enable extraction of ConversationIDs from EDI documents

1. Obtain a license and install Process Engine if you haven't already done so.
2. For Transaction document types, enable Module for EDI to assign **ConversationIDs** by creating an instance ID query that specifies the value to use for the conversation ID.

Note: By default, Module for EDI assigns the conversation IDs in Interchange and Group documents. Because these documents have a conversation ID, they are passed to Process Engine.

3. Using Software AG Designer, add a package dependency from WmTN to WmPRT to ensure that the conversation IDs are extracted correctly. For details, see the *webMethods Service Development Help* for your release.
4. Configure a PRT database connection. Follow the installation instructions for Process Engine to configure a data source.
5. Reload the WmTN package.

Disabling 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 function to extract **ConversationIDs** from EDI documents using the following procedure.

1. Do one of the following:
 - For Envelope and Group document types, set the property `disableCIDSupport=true` in the configuration file, `Integration Server_directory\instances\$instance_name\packages\WmEDI\config\properties.cnf` file.
 - For Transaction document types, use Designer to remove the dependency in the WmTN package on the WmPRT package.
2. Reload the WmTN package.

Custom Attributes that Module for EDI Sets

When you install Module for EDI, the following custom attributes are added to Trading Networks. You can use these attributes as criteria when designing processing rules.

Custom Attribute	Where to Find More Information
EDI Batch	“Delivering the Batch EDI Document” on page 293
EDI Group Type	“EDI Group Type” on page 192
EDI Outbound FA	“EDI Outbound FA” on page 192
EDI Processing Mode	“EDI Processing Mode” on page 192
EDI Status	“EDI Status” on page 193
EDI Version	“EDI Version” on page 193
Envelope CntrlNum Status and Group CntrlNum Status	“Envelope CntrlNum Status and Group CntrlNum Status” on page 193
EDI FA Status	“EDI FA Status” on page 194
Late FA	“Tracking Late Functional Acknowledgments” on page 167
TA1 Status and TA1 Code	“TA1 Status and TA1 Code” on page 195

For information about custom attributes that Module for EDI sets when using TRADACOMS, see [“Custom Attributes that Module for EDI Sets When Using TRADACOMS” on page 215](#).

EDI Group Type

For Group documents, Module for EDI sets this attribute to the value of the functional identifier (for example, PO, or RC) from the group header.

EDI Outbound FA

Module for EDI sets this attribute to indicate whether the document it generated is an outbound FA. The value of the **EDI Outbound FA** custom attribute is set to:

- true for an outbound FA that the module generated
- false for all other documents

EDI Processing Mode

Module for EDI sets this attribute to `Testing`, `Production`, or `Custom` based on the value of the `EDITPA processingMode` variable. For more information, see [“processingMode Variable” on page 66](#).

EDI Status

Module for EDI sets this attribute based on the outcome of optional inbound processing, that is, inbound control number validation and automatic FA generation. The module only performs this optional inbound processing when the EDITPA indicates to do so. For more information, see [“Optional Inbound Processing When Using Trading Networks” on page 229](#).

If the EDITPA indicates to do so, the module first performs inbound control number validation. If the module determines that a document contains:

- A duplicate control number, it sets this attribute to `Duplicate Control Number`.
- An out-of-sequence control number, it sets this attribute to `Out of Sequence Control Number`.

Regardless of whether control number validation was performed, Module for EDI will generate the FA if the EDITPA indicates to do so. If there is an unacceptable FA status, the module sets the **EDI Status** attribute based on the FA status as follows:

- Generate FA–Not Allowed
- Generate FA–Rejected
- Generate FA–Partially Accepted
- Generate FA–Accepted, But Errors Were Noted

If the control numbers are valid *and* the FA status is acceptable, Module for EDI sets the **EDI Status** attribute to `Processed`. Additionally, if the module does not validate the control number nor generate the FAs, it sets the EDI Status attribute to `Processed`.

EDI Version

Module for EDI sets this attribute to the version of the EDI standard that the document uses (for example, 4010).

Envelope CntrlNum Status and Group CntrlNum Status

The **Envelope CntrlNum Status** and **Group CntrlNum Status** attributes indicate the control number validation status of documents, thus enabling child documents to know the validation status of their parent documents.

When validating document control numbers, Module for EDI sets this attribute to `Valid`, `Duplicate`, `Out of Sequence`, or `Not Validated`.

- For Interchange documents, the module only sets the **Envelope CntrlNum Status** attribute.
- For Group documents, the module sets the following attributes:
 - **Envelope CntrlNum Status** to indicate the status of the associated Interchange document
 - **Group CntrlNum Status** to indicate the status of the associated Group document
- For Transaction documents, the module sets the following attributes:

- **Envelope CntrlNum Status** to indicate the status of the associated Interchange document
- **Group CntrlNum Status** to indicate the status of the associated Group document

Note:

For UN/EDIFACT Transaction documents, Module for EDI may also set this attribute to Not Present.

EDI FA Status

The **EDI FA Status** attribute indicates the status of a functional acknowledgment (FA) that Module for EDI has returned to the sender of an inbound document.

- For inbound ANSI X12 Group documents and UN/EDIFACT Interchange documents, this attribute may return the following values:

Value	Description
Not Acknowledged	Module for EDI has not yet received or sent an FA to acknowledge this document
Duplicate	Module for EDI has one or more other documents recorded in the EDITRACKING table that match the FA for this document.
Accepted	Module for EDI received a single FA that matches this document and the FA has either: <ul style="list-style-type: none"> ■ "A" (Accept) status on the confirmed level (ANSI X12) ■ "7" status on the confirmed level (UN/EDIFACT)
Accepted w/Errors	Module for EDI received a single FA that matches this document and the FA has an "E" status (Errors) on the confirmed level.
Accepted - Partial	Module for EDI received a single FA that matches this document and the FA has a "P" status (Partially Accept) on the confirmed level. This value is used only for Group documents.
Rejected	Module for EDI received a single FA that matches this document and the FA has either: <ul style="list-style-type: none"> ■ "R" (Reject) status on the confirmed level (ANSI X12) ■ "4" status on the confirmed level (UN/EDIFACT)
FA Error	Module for EDI encountered other errors, such as "unknown" or "invalid FA status."
Disabled	The <i>FAReconciliation</i> EDITPA variable is set to false, which disables FA reconciliation and reporting. For more information, see "FAReconciliation Variable" on page 68 .

- For functional acknowledgments (for example, ANSI X12 997s and UN/EDIFACT CONTRL documents), this attribute may return the following values:

Value	Description
Matched	A matching document exists in the EDITRACKING table.
Unmatched	No matching document exists in the EDITRACKING table.
Duplicated	Module for EDI has one or more other documents recorded in the EDITRACKING table that match the FA for this document.
Disabled	The <i>FAReconciliation</i> EDITPA variable is set to <code>false</code> , which disables FA reconciliation and reporting. For more information, see “FAReconciliation Variable” on page 68 .
DUP_FA	Module for EDI received more than one FA that matches this document.

Note:

Be sure to turn on automatic FA generation, as described in [“Turning Automatic FA Generation On and Off” on page 247](#). Otherwise, the **FA Status** attribute will always return `Not Acknowledged` or `Disabled`.

TA1 Status and TA1 Code

When Module for EDI receives an inbound EDI document containing a TA1 segment, the module uses Trading Networks to recognize, route, and persist the document. During recognition, the TA1 segment is processed and the TA1 data are set as the bizdoc attributes **TA1 Status** and **TA1 Code**.

The **TA1 Status** attribute indicates the status of an interchange acknowledgment (TA1) that Module for EDI in an inbound document.

TA1 Status	Description
A	The TA1 segment was accepted.
R	The TA1 segment was rejected.
E	The TA1 segment has an error.

When the **TA1 Status** is `A`, the **TA1 Code** is `000`.

When the **TA1 Status** is either `R` or `E`, the **TA1 Code** attribute indicates the type of error that the module encountered. The following table lists the error codes and their meanings.

TA1 Code Value	Description
001	The interchange control number in the header and trailer do not match. The acknowledgment uses the value in the header.

TA1 Code Value	Description
002	The standard noted in the Control Standards Identifier is not supported.
003	The version of the controls is not supported.
004	The segment terminator is not valid.
005	The interchange ID qualifier for sender is not valid.
006	The interchange ID for sender is not valid.
007	The interchange ID qualifier for recipient is not valid.
008	The interchange ID for recipient is not valid.
009	The interchange receiver ID is unknown.
010	The Authorization Information Qualifier value is not valid.
011	The Authorization Information value is not valid.
012	The Security Information Qualifier is not valid.
013	The Security Information value is not valid.
014	The Interchange Date is not valid.
015	The Interchange Time is not valid.
016	The Interchange Standards ID is not valid.
017	The Interchange Version ID is not valid.
018	The Interchange Control Number is not valid.
019	The Acknowledgment Request value is not valid.
020	The Test Indicator is not valid.
021	The Number of Included Group value is not valid.
022	The control structure is not valid.
023	Improper end of file.
024	The Interchange content is not valid.
025	Interchange Control Number is a duplicate.
026	The Data Element Separator is not valid.
027	The Component Element Separator is not valid.
028	The delivery date in the Deferred Delivery Request is not valid.

TA1 Code Value	Description
029	The delivery time in the Deferred Delivery Request is not valid.
030	The delivery time code in the Deferred Delivery Request is not valid.
031	The grade of service code is not valid.

Managing Custom Attributes for EDI Documents

When you want to create and use custom attributes, you must perform the following tasks:

Task	Where to Find More Information
Define the custom attributes using My webMethods.	The <i>webMethods Trading Networks Administrator's Guide</i> for your release.
Define the TN document types for the types of EDI documents with which you want to associate the custom attributes.	“Defining TN EDI Document Types” on page 34.
Create a service that sets the attribute values at run time.	“Setting the Attribute Values at Run Time” on page 197.
Define a processing rule that uses the custom attributes.	“Defining a Processing Rule that Uses Your Custom Attribute as Criteria” on page 198.

Setting the Attribute Values at Run Time

You must create a service to set the attribute values at run time. To execute this service, include it as a parameter of the **Execute a Service** processing action in a Trading Networks processing rule. Note that your EDI document type or processing rule must use the **Save Document to Database** preprocessing 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 copy of the document that is saved in the Trading Networks database.

Example

Let's say you are working with an EDI ANSI X12 850 document, and you 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 service that extracts the value for the PurchaseOrderNumber attribute from the 850 EDI document, and assign this service as input to the **Execute a Service** processing action in the processing rule. Your service should do the following:
 - Invoke `wm.tn.doc:setAttribute` to set the value of the attribute in the BizDocEnvelope
 - Invoke `wm.tn.doc:updateAttributes` 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 [“Managing Custom Attributes for EDI Documents” on page 197](#).
- Create a processing rule to process the document, considering the following:
 - Do not use content-based processing.
 - Leave the **Extended Criteria** tab of the Processing Rules screen blank.
 - In the **Execute a Service** processing rule, invoke the service that sets the values of the custom attributes, as described in [“Setting the Attribute Values at Run Time” on page 197](#).
 - Ensure that the service invokes the `wm.tn.reroute` service so that Trading Networks selects another processing rule for the document. (The next processing rule uses your custom attributes as criteria.)
- Create a processing rule that specifies the custom attributes as criteria in the **Extended Criteria** tab.

Note:

When ordering 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. This ensures that 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

You can specify how Module for EDI processes Interchange, Group, and Transaction documents that have been split from the original inbound EDI document by using processing rules. Your processing rules specify the preprocessing and processing actions that you want Trading Networks to perform against the document, as well as the selection criteria that Trading Networks uses to determine which processing rule to apply.

The processing rules that you define are based on the *splitOption* variable that you set 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 (for example, ANSI X12 850, ANSI X12 810, UN/EDIFACT INVOIC, or UN/EDIFACT ORDERS)

Define processing rules in My webMethods. For more information, see the *webMethods Trading Networks Administrator's Guide* for your release.

Specifying Processing Rule Criteria

When you create a processing rule, you specify the criteria that Trading Networks uses to select the appropriate processing rule for an Interchange, Group, or Transaction document. The following table lists the criteria you can specify.

Use this criterion...	To have Trading Networks select a processing rule based on...
Sender	<p>The sender of the document</p> <p>The <i>GSRouting/routingMode</i> and <i>GSRouting/senderQualifier</i> EDITPA variables can affect the sender that Module for EDI uses for an interchange, group, and transaction. For more information, see “GSRouting Variables” on page 63.</p>
Receiver	<p>The receiver of the document</p> <p>The <i>GSRouting/routingMode</i> and <i>GSRouting/receiverQualifier</i> EDITPA variables can affect the receiver that Module for EDI uses for an interchange, group, and transaction. For more information, see “GSRouting Variables” on page 63.</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. The following are example settings:</p>

Use this criterion...	To have Trading Networks select a processing rule based on...	
	To select a processing rule for document type...	Set Document Type to...
	ANSI X12 interchange	X12 Envelope
	ANSI X12 group	X12 Group
	ANSI X12 850 transaction, version 4010	X12 4010 850
	UN/EDIFACT interchange	UNEDIFACT Envelope
	UN/EDIFACT group	UNEDIFACT Group
	UN/EDIFACT ORDERS transaction, version 99A	UNEDIFACT 99A ORDERS
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 that the EDI Processing Mode attribute must be <i>Production</i> to select the processing rule. For more information about the custom attributes that Module for EDI sets, see “Custom Attributes that Module for EDI Sets” on page 192. For information about how to set your own custom attributes, see “Managing Custom Attributes for EDI Documents” on page 197.</p>	

Note: Module for EDI does not set the **User Status** system attribute, so you should not use the **User Status** criterion.

Specifying Preprocessing Actions

Default preprocessing actions are defined in TN document types. However, you can override the settings in the processing rule. For more information about how the preprocessing actions apply to EDI documents, see the section about preprocessing actions in *webMethods Module for EDI Concepts Guide*.

The following table lists the preprocessing actions that you can set in a processing rule and the default settings for each action in the TN document types for EDI documents.

Preprocessing 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

Preprocessing action	Description	Default setting in TN document type for the EDI document
Check for Duplicate Document	Determines if Trading Networks has already saved 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 preprocessing 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** preprocessing 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 transactions from an inbound EDI document, for example, to transform that information into a format that is readable by an internal (back-end) system, you will primarily use the **Execute a Service** action. You must create the service that the **Execute a Service** processing action invokes.

In this scenario, the *splitOption* EDITPA variable must be set to Transaction or Group.

- To process the Transaction and/or Group documents from the split, you would assign the appropriate processing rules in the **Execute a Service** processing action. For more information about how to create the services, see [“Coding Services to Process Transaction and Group Documents” on page 202](#).
- For the Interchange document, you could define a processing rule with no processing actions selected. This causes Trading Networks to ignore the document because there is no need to process it. Alternatively, you could define processing rules that 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 in Trading Networks to simply deliver it to a destination without processing the file, you will primarily use the **Deliver Document By** action, with the *splitOption* EDITPA variable set to Interchange. In this scenario, the only document sent to Trading Networks is the Interchange document. For more information about delivering documents, see [“Forming EDI Documents to Send Outbound When Using Trading Networks” on page 257](#).

Coding Services to Process Transaction and Group Documents

The logic that you design in your processing service (set in the **Execute a Service** action) depends on the type of document that Module for EDI generates when splitting the document from the original EDI document. This document type is specified in the EDITPA *splitOption* variable.

The following table explains the types of documents that the module creates depending on the value of the *splitOption* variable.

If the <i>splitOption</i> is...	Module for EDI creates...
Transaction	<p>Transaction, and Group and Interchange documents from the original EDI document. When the split document is of type:</p> <ul style="list-style-type: none"> ■ Transaction. Create a service that processes a single transaction set from the document. For example, create a document in a format used by an internal application based on the transaction set information. For information about how to create this service, see “Logic to Process a File Document” on page 224. ■ Group and interchange. Do not create a service because all processing for EDI documents is handled when processing Transaction documents. You could also set up processing rules specifically for group and Interchange documents that have no processing actions selected. Or, set up your processing rules so that Interchange documents fall through to the Default rule. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: If you want to generate functional acknowledgments (FAs), see “Automatically Generating Functional Acknowledgments” on page 246.</p> </div>
Group	<p>Group and Interchange documents from the original EDI document. When the split document is of type:</p> <ul style="list-style-type: none"> ■ Group. Create a service that processes each transaction within the group. For information about how to create this service, see “Logic to Map a File Document to an Internal-Format Document” on page 226. ■ Interchange. Do not create a service. Processing for the EDI document is handled when processing the Group document. You could also set up a processing rule specifically for the Interchange document that has no processing actions defined or set up your processing rules so that Interchange document falls through to the Default rule. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: If you want to generate functional acknowledgments (FAs), see “Automatically Generating Functional Acknowledgments” on page 246.</p> </div>

If the *splitOption* is... Module for EDI creates...

Interchange Interchange documents from the original EDI document. You would set the *splitOption* to Interchange if you want to simply deliver the EDI document to its receiver.

For more information about how to deliver a document, see [“Delivering the Outbound EDI Document” on page 264](#).

The services you create can use information that is in the pipeline. For information about using data in the pipeline when your service is invoked, see [“Information in the Pipeline that Your Service Can Access” on page 203](#).

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 accessible to your service from the pipeline:

- 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 more information about the BizDocEnvelope, see the *webMethods Trading Networks Built-In Services Reference* for your release. The following variables are contained in the BizDocEnvelope:

Variable	Description
<i>DocumentID</i>	The EDI control number from the interchange, group, or transaction header, as described in “DocumentID” on page 214 .
<i>Attributes</i>	The custom attributes that were set for the document. For information about the custom attributes that Module for EDI sets, see “Custom Attributes that Module for EDI Sets When Using TRADACOMS” on page 215 .
<i>Errors</i>	The errors that Trading Networks and Module for EDI 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 document and generates this summary. 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 can also perform a query within Trading Networks that checks the status of each document using the Trading Networks internal ID for each document.

The following table 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 transactions sets that it contains.																																
	<table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Interchange document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type used for the Interchange document.</td> </tr> <tr> <td><i>docTypeName</i></td> <td>String Name of the TN document type used for the Interchange document, for example, X12 Envelope.</td> </tr> <tr> <td><i>group Documents</i></td> <td>Document List (optional) Summary of all the Group documents in the interchange.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Group document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type.</td> </tr> <tr> <td><i>docType Name</i></td> <td>String Name of the TN document type used, for example, X12 Group.</td> </tr> <tr> <td><i>transaction Documents</i></td> <td>Document List (optional) Summary of all the Transaction documents in the group.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Transaction document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type used.</td> </tr> <tr> <td><i>docType Name</i></td> <td>String Name of the TN document type used, for example, X12 4010 850.</td> </tr> </tbody> </table> </td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Key	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, for example, X12 Envelope.	<i>group Documents</i>	Document List (optional) Summary of all the Group documents in the interchange.		<table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Group document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type.</td> </tr> <tr> <td><i>docType Name</i></td> <td>String Name of the TN document type used, for example, X12 Group.</td> </tr> <tr> <td><i>transaction Documents</i></td> <td>Document List (optional) Summary of all the Transaction documents in the group.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Transaction document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type used.</td> </tr> <tr> <td><i>docType Name</i></td> <td>String Name of the TN document type used, for example, X12 4010 850.</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Key	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.	<i>docType Name</i>	String Name of the TN document type used, for example, X12 Group.	<i>transaction Documents</i>	Document List (optional) Summary of all the Transaction documents in the group.		<table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Transaction document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type used.</td> </tr> <tr> <td><i>docType Name</i></td> <td>String Name of the TN document type used, for example, X12 4010 850.</td> </tr> </tbody> </table>	Key	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.	<i>docType Name</i>	String Name of the TN document type used, for example, X12 4010 850.
Key	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, for example, X12 Envelope.																																
<i>group Documents</i>	Document List (optional) Summary of all the Group documents in the interchange.																																
	<table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Group document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type.</td> </tr> <tr> <td><i>docType Name</i></td> <td>String Name of the TN document type used, for example, X12 Group.</td> </tr> <tr> <td><i>transaction Documents</i></td> <td>Document List (optional) Summary of all the Transaction documents in the group.</td> </tr> <tr> <td></td> <td> <table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Transaction document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type used.</td> </tr> <tr> <td><i>docType Name</i></td> <td>String Name of the TN document type used, for example, X12 4010 850.</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Key	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.	<i>docType Name</i>	String Name of the TN document type used, for example, X12 Group.	<i>transaction Documents</i>	Document List (optional) Summary of all the Transaction documents in the group.		<table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Transaction document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type used.</td> </tr> <tr> <td><i>docType Name</i></td> <td>String Name of the TN document type used, for example, X12 4010 850.</td> </tr> </tbody> </table>	Key	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.	<i>docType Name</i>	String Name of the TN document type used, for example, X12 4010 850.												
Key	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.																																
<i>docType Name</i>	String Name of the TN document type used, for example, X12 Group.																																
<i>transaction Documents</i>	Document List (optional) Summary of all the Transaction documents in the group.																																
	<table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>String Internal ID that Trading Networks generated for the Transaction document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>String Internal ID of the TN document type used.</td> </tr> <tr> <td><i>docType Name</i></td> <td>String Name of the TN document type used, for example, X12 4010 850.</td> </tr> </tbody> </table>	Key	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.	<i>docType Name</i>	String Name of the TN document type used, for example, X12 4010 850.																								
Key	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.																																
<i>docType Name</i>	String Name of the TN document type used, for example, X12 4010 850.																																

Logic to Process a Transaction Document

The following sample code shows the logic you might want to apply when creating a service that will process a Transaction document that contains a single transaction set from an inbound EDI document. The processing in this example shows how to map information from the transaction set to an internal-format and send the document to an internal application.

```

1 → wm.b2b.edi:getTspace
2 → wm.b2b.edi:convertToValues
  ↳ BRANCH on '/isValid'
    ↳ true: SEQUENCE
      3 → MAP (Map the data from the EDI transaction set into a document in internal)
      4 → wm.b2b.edi:convertToString (Uses flat file schema to convert internal format document to string.)
    ↳ BRANCH on '/bizdoc/Attributes/EDI Processing Mode'
      6 → Production: SEQUENCE
        ↳ wm.sendToBackend
      7 → $default: SEQUENCE
        ↳ pub.flow.debugLog

```

Flow operation	Description
1	<p>Invoke the <code>wm.b2b.edi:getTspace</code> service to retrieve the content of the Transaction document from the BizDocEnvelope (in the <code>bizdoc</code> variable) and use it to populate the <code>edidata</code> variable.</p> <p>You can use the <code>getTspace</code> service regardless of whether the document is considered large. For more information about large document handling, see “Handling Large Documents When Using Trading Networks” on page 149.</p> <p>For more information about the <code>getTspace</code> service, see <i>webMethods Module for EDI Built-In Services Reference</i>.</p>
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 object into an IData object. Validate the structure of the EDI transaction. <p>The inputs to the <code>convertToValues</code> service include the flat file schema for the EDI transaction. The <code>convertToValues</code> service uses the flat file schema to 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 <i>webMethods Module for EDI Built-In Services Reference</i>.</p>
3	<p>Map the data from the EDI transaction set into a document in internal format.</p> <p>Now that the transaction set 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.</p>

Flow operation	Description
4	<p>Invoke the <code>wm.b2b.edi:convertToString</code> service to convert the document in internal format from an <code>IData</code> object into a <code>String</code>.</p> <p>The input to the <code>convertToString</code> service includes the <code>IData</code> object, which contains the data and the flat file schema for the internal-format document. <code>convertToString</code> uses the flat file schema to determine how to form the internal-format document. Alternatively, <code>convertToString</code> can use an IS document type to define the structure of the internal-format document.</p> <p>For more information about <code>convertToString</code>, see <i>webMethods Module for EDI 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. Module for EDI sets the value of the EDI Processing Mode custom attribute based on the setting of the <code>EDITPA processingMode</code> variable. For more information, see “processingMode Variable” on page 66.</p>
6	<p>If the value of the EDI Processing Mode custom attribute is:</p> <ul style="list-style-type: none"> <li data-bbox="302 947 1380 1020">■ <code>Production</code>—Send the internal-format document to the internal (back-end) system, by adding your own logic or invoking a service that you created. <li data-bbox="302 1041 1380 1146">■ A value other than <code>Production</code> (for example, <code>Testing</code>)—Invoke <code>wm.pub.flow:debugLog</code> to log a message to the server log. For example, the message may be, “Testing document was received but not sent to back-end system.”

Logic to Process a Group Document when *splitOption* is Group

When the *splitOption* EDITPA variable is `Group`, Module for EDI does not create Transaction documents. If you want to process individual transactions, you can do so when you process the Group document. The following sample code illustrates the processing of individual transactions within the group.

```

1 → wm.b2b.editn:getTspace (get the Group document content from the bizdoc to edidata)
2 → wm.b2b.edi:envelopeProcess (convert the interchange and Group envelope to an IData object)
├─ BRANCH on '/hasError'
│   └─ true: SEQUENCE (If errors occur when converting envelopes to IData objects, record error to server log)
│       └─ pub.flow:debugLog (log the error to the server log)
├─ $default: SEQUENCE (envelope conversion to IData object was successful)
│   └─ MAP (drop error variables)
│       └─ LOOP over '/Values/ISA'
│           └─ LOOP over '/Values/ISA/GS'
│               └─ LOOP over '/Values/ISA/GS/ST'
│                   └─ 5 → wm.b2b.edi.util:getEDIStrng (get the transaction, including headers, into a String)
│                       └─ 6 → wm.b2b.edi:convertToValues (convert the transaction String to an IData object)
│                           └─ BRANCH on '/isValid'
│                               └─ true: SEQUENCE
│                                   └─ 7 → MAP (map data from the transaction to the IData object for the internal-format document)
│                                       └─ 8 → wm.b2b.edi:convertToString (Uses flat file schema for internal-format document to create to String)
└─ 9 → BRANCH on '/bizdoc/Attributes/EDI Processing Mode'
    └─ Production: SEQUENCE
        └─ 10 → wm:sendToBackend
            └─ $default: SEQUENCE
                └─ 11 → pub.flow:debugLog

```

Flow operation	Description
1	<p>Invoke the <code>wm.b2b.editn:getTspace</code> service to retrieve the content of the Group document from the BizDocEnvelope (in the <code>bizdoc</code> variable) to populate the <code>edidata</code> variable. You can use the <code>getTspace</code> service regardless of whether the document is considered large.</p> <p>For more information about <code>getTspace</code>, see <i>webMethods Module for EDI 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 IData object named <i>Values</i>. The contents of the transaction sets remain unparsed.</p> <ul style="list-style-type: none"> For ANSI X12 documents, Module for EDI creates an IData object of the ISA/IEA, GS/GE, and ST/SE headers/trailers. For UN/EDIFACT documents, Module for EDI creates an IData 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 module already performs this validation during initial processing of the EDI document.</p> <p>For more information about <code>envelopeProcess</code>, see <i>webMethods Module for EDI 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 <i>Values</i> IData object:</p>

Flow operation	Description
	<ul style="list-style-type: none"> ■ For ANSI X12 documents, transaction set headers are in <i>Values/ISA/GS/ST</i> (as shown in the sample code). ■ For UN/EDIFACT documents, transaction set headers are in <i>Values/UNB/UNG/UNH</i>.

The remaining steps specify processing to perform for the content of each transaction.

Flow operation	Description
5	<p>Invoke the <code>wm.b2b.edi.util:getEDString</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 <code>String</code> or <code>InputStream</code> object. For more information about the <code>wm.b2b.edi.util:getEDString</code> service, see <i>webMethods Module for EDI Built-In Services Reference</i>.</p> <p>You must 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 <code>convertToValues</code> 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, <code>convertToValues</code> returns 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 a <code>String</code> or <code>InputStream</code> object to an <code>IData</code> object. ■ Validate the structure of the EDI transaction. <p>The inputs to the <code>convertToValues</code> service include the flat file schema for the EDI transaction. <code>convertToValues</code> uses the flat file schema to determine how to parse the transaction set into an <code>IData</code> object and validate its structure.</p> <p>For more information about <code>convertToValues</code>, see <i>webMethods Module for EDI Built-In Services Reference</i>.</p>
7	<p>Map the data from the EDI transaction set into the internal format.</p> <p>Now that the contents of the transaction set are 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 <code>MAP</code> flow operation, or create a separate service to perform the mapping.</p>
8	<p>Invoke the <code>wm.b2b.edi:convertToString</code> service to convert the internal-format document from an <code>IData</code> object to a <code>String</code>.</p>

Flow operation	Description
	<p>The inputs to the <code>convertToString</code> service include the <code>IData</code> object, which contains the data and the flat file schema for the document in internal format. <code>convertToString</code> uses the flat file schema to determine how to form the internal-format document. <code>convertToString</code> can also use an IS document type to define the structure of the internal-format document.</p> <p>For more information about <code>convertToString</code>, see <i>webMethods Module for EDI 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. Module for EDI sets the value of the EDI Processing Mode custom attribute based on the setting of the <code>EDITPA processingMode</code> variable. For more information, see “processingMode Variable” on page 66.</p>
10	<p>If the value of the EDI Processing Mode custom attribute is:</p> <ul style="list-style-type: none"> <li data-bbox="415 856 1481 926">■ Production—Send the document in internal format to the back-end system by adding your own logic or invoking a service you created. <li data-bbox="415 947 1481 1052">■ A value other than <code>Production</code> (for example, <code>Testing</code>) —Invoke <code>wm.pub.flow:debugLog</code> to log a message to the server log. For example, the message may be, “Testing document was received but not sent to back-end system.”

12 Processing Inbound TRADACOMS Documents Using Trading Networks

■ Overview	212
■ Before Setting Up Processing of Inbound EDI Documents When Using TRADACOMS .	213
■ Specifying EDITPA Variables that Affect Inbound Processing When Using TRADACOMS	213
■ Trading Networks Attributes and EDI Documents When Using TRADACOMS	214
■ Defining Processing Rules to Process Inbound EDI Documents When Using TRADACOMS	218
■ Coding Services to Process File and Batch Documents When Using TRADACOMS ...	221

Overview

This chapter specifically addresses processing inbound TRADACOMS documents using Trading Networks. For information about processing inbound documents using any other EDI standard, see [“Preparing to Process Inbound Non-TRADACOMS Documents Using Trading Networks” on page 185](#).

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 in Trading Networks when you install webMethods Module for EDI (Module for EDI). You can use EDITPA variables to tailor how the EDI recognizer processes an inbound EDI document. For more information, see [“Specifying EDITPA Variables that Affect Inbound Processing When Using TRADACOMS” on page 213](#).

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 setting of the *TRADACOMS/splitOption* variable 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 to Trading Networks for normal processing.

You define the processing that Trading Networks performs against transmission and batch documents, and files by defining processing rules. Processing rules consist of:

- Criteria that Trading Networks uses to select the appropriate processing rule for a 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 that Trading Networks selects the correct processing rule for your document. For example, you can set up criteria so that Trading Networks selects a processing rule only if it is from specific senders. For more information about the document attributes that are available, see [“Trading Networks Attributes and EDI Documents When Using TRADACOMS” on page 214](#).
- Actions that Trading Networks is to take against the document. 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 When Using TRADACOMS” on page 218](#).

Note:

This chapter describes how to set up basic inbound processing when using Trading Networks. For information about setting up to validate transmission and/or batch control numbers, see [“Optional Inbound Processing When Using Trading Networks” on page 229](#).

For more information about processing EDI documents and business processes, including an illustration of how EDI documents are passed to a business process, see *webMethods Module for EDI Concepts Guide*.

For more information about Trading Networks document attributes and processing rules, see the *Trading Networks Concepts Guide* and the *webMethods Trading Networks Administrator’s Guide* for your release.

Before Setting Up Processing of Inbound EDI Documents When Using TRADACOMS

When using the TRADACOMS standard, the following tasks must be performed before setting up processing for inbound of EDI documents.

- | Task | Description |
|------|--|
| 1 | <p>Define the TN document types for the EDI documents that you want to process. For instructions, see “Defining TN EDI Document Types” on page 34.</p> <p>When you define the TN document types for EDI documents, the flat file schemas that define the structure of the EDI documents to process are created. Module for EDI uses the flat file schemas for parsing, converting, and validating the structure of an inbound EDI document.</p> |
| 2 | <p>Optional. Create the flat file schema that defines the structure of an internal-format document. Use Designer to create the flat file schema. For more information about creating a flat file schema, see <i>Flat File Schema Developer’s Guide</i>.</p> <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <p>Note:
You use the flat file schema created in this step if the service that you create to process a file, or a transmission or batch document might create an internal-format document. Module for EDI provides a service that you can invoke to create the internal-format document based on the flat file schema.</p> </div> |
| 3 | <p>Define profiles for the senders and receivers in the inbound EDI document. For more information about creating profiles, see “Defining Trading Partner Profiles When Using TRADACOMS” on page 99 and the <i>webMethods Trading Networks Administrator’s Guide</i> for your release.</p> |
| 4 | <p>Define the default EDITPA and optionally, the partner-specific EDITPAs for the sender/receiver pairs in the EDI document. For more information about creating EDITPAs, see “Defining EDI Trading Partner Agreements When Using TRADACOMS” on page 101. For information about the EDITPA variables that affect inbound processing, see “Specifying EDITPA Variables that Affect Inbound Processing When Using TRADACOMS” on page 213.</p> |

Specifying EDITPA Variables that Affect Inbound Processing When Using TRADACOMS

To tailor how you want Module for EDI to perform inbound processing, you must edit the EDITPA variables. When using TRADACOMS, Module for EDI processes inbound EDI documents one transmission segment at a time. For each transmission segment in an inbound EDI document, the module obtains the EDITPA values to use for transmission sender/receiver pair. The module uses these values when processing all documents (transmission, batch, and file) for the transmission segment.

For a complete list and description of the TRADACOMS EDITPA variables, see [“TRADACOM Variables” on page 103](#).

Trading Networks Attributes and EDI Documents When Using TRADACOMS

Trading Networks attributes specify the content from a document to use for later processing. For example, you can define the processing rule criteria to select which processing rule is executed based on the value of an attribute.

Trading Networks supports two types of attributes:

- System attributes, which are defined by Trading Networks.
- Custom attributes, which are additional attributes that are added to Trading Networks. Module for EDI provides some custom attributes. Additionally, you can write code to define your own custom attributes and assign them values.

At run time, Module for EDI uses information from the EDI document being processed to set values for some of the system attributes and 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.

System Attributes that Module for EDI Sets When Using TRADACOMS

When using TRADACOMS, Module for EDI sets the **DocumentID** and **GroupID** system attributes using information from the EDI document. The 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 module sets each attribute for each type of document (transmission, batch, and file).

DocumentID

The **DocumentID** system attribute is an identifier of the document. Module for EDI sets the **DocumentID** as follows:

Type of Document	Value Used for DocumentID
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.

Type of Document	Value Used for DocumentID
File	The file generation number in the message header's FIL segment.

GroupID

Module for EDI hierarchically assigns the **GroupID** system attribute so that by viewing the **GroupID**, you can determine the envelope to which an element originally belonged. The module sets the **GroupID** as follows:

Type of Document	Value Used for GroupID
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 Module for EDI Sets When Using TRADACOMS

When you install Module for EDI, the following custom attributes are added to Trading Networks. You can use these attributes as criteria when designing processing rules. The following 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	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	
Transaction CntrlNum Status	<p>When validating document control numbers, Module for EDI sets this attribute to Valid, Duplicate, Out of Sequence, or Not Validated.</p> <ul style="list-style-type: none"> ■ For transmission documents, the module only sets the attribute Envelope CntrlNum Status. ■ For batch documents, the module sets the following attributes: <ul style="list-style-type: none"> ■ Envelope CntrlNum Status to indicate the status of the associated transmission document. ■ Group CntrlNum Status To indicate the status of the associated batch document. ■ For file documents, the 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 module might also set this attribute to Not Present. ■ Transaction CntrlNum Status. Indicates the status of the associated transaction document.

Managing Your Own Custom Attributes for EDI Documents When Using TRADACOMS

When you are using TRADACOMS and you want to create and use custom attributes of your own, you must perform the following steps:

Step	Description	Where to Find More Information
1	Define the custom attributes in My webMethods.	The <i>webMethods Trading Networks Administrator's Guide</i> for your release.
2	Define the TN document type for the type of EDI document with which you want to associate the custom attribute.	“Defining TN EDI Document Types” on page 34.

Step	Description	Where to Find More Information
3	Invoke the <code>wm.b2b.editn:addAttributeTypeToBizDoc</code> service to associate the custom document attribute that you created in the previous step with the TN document type you installed in the first step.	<i>webMethods Module for EDI Built-In Services Reference</i>
4	Define a processing rule that uses your custom attributes.	“Defining a Processing Rule that Uses Your Custom Attribute as Criteria When Using TRADACOMS” on page 218.

Setting Attribute Values at Run Time When Using TRADACOMS

As with other standards, you must create a service to set the attribute values at run time when using TRADACOMS. To execute this service, include it as a parameter of the **Execute a Service** processing action in a Trading Networks processing rule. Note that your EDI document type or processing rule must use the **Save Document to Database** preprocessing 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 copy of the document that is saved in the Trading Networks database.

Example

Let's say you are working with the File document type, and you 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.editn:addAttributeTypeToBizDoc` service to associate the `PurchaseOrderNumber` attribute with the TN document type for the ORDERS document.
4. Create a service that extracts the value for the `PurchaseOrderNumber` attribute from the ORDERS document, and assign this service as input to the **Execute a Service** processing action in the processing rule. Your service should do the following:
 - Invoke the `wm.tn.doc:setAttribute` service to set the value for the attribute in the `BizDocEnvelope`
 - Invoke `wm.tn.doc:updateAttributes` 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 When Using TRADACOMS

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 [“Managing Your Own Custom Attributes for EDI Documents When Using TRADACOMS”](#) on page 216.
- Create a processing rule to process the document, considering the following:
 - Do not use content-based processing.
 - Leave the **Extended Criteria** tab of the Processing Rules screen blank.
 - In the **Execute a Service** processing rule, invoke a service that sets the values of the custom attributes, as described in [“Setting Attribute Values at Run Time When Using TRADACOMS”](#) on page 217.
 - Ensure that the service invokes the `wm.tn.reroute` service so that Trading Networks selects another processing rule for the document. (The next processing rule uses your custom attributes as criteria.)
- Create a processing rule that uses the custom attributes as criteria in 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. This ensures that 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 When Using TRADACOMS

When using TRADACOMS, you can specify how Module for EDI processes transmission, batch, and file documents that have been split from the original inbound EDI document by using processing rules. Your processing rules specify the preprocessing and processing actions that you want Trading Networks to perform against the document, as well as the selection criteria that Trading Networks uses to determine which processing rule to apply.

The processing rules you need to define are based on the `TRADACOMS/splitOption` variable that you set in the EDITPA.

If you set <code>TRADACOMS/splitOption</code> to...	Then define processing rules for...
Transmission	Transmission document
Batch	■ Transmission document

If you set <i>TRADACOMS/splitOption</i> to...	Then define processing rules for...
	<ul style="list-style-type: none"> ■ Batch documents
File	<ul style="list-style-type: none"> ■ Transmission document ■ Batch documents ■ Each type of File document (for example, ORDHDR, INVHDR, etc.)

Define processing rules using My webMethods. For more information, see the *webMethods Trading Networks Administrator's Guide* for your release.

Specifying Processing Rule Criteria When Using TRADACOMS

When you create a processing rule, you specify the criteria that Trading Networks uses to select the appropriate processing rule for a transmission, batch, or file document. The following table 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								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="border-bottom: 1px solid black;">To have Trading Networks select a processing rule for...</th> <th style="border-bottom: 1px solid black;">Set the Document Type criterion to this TN document type...</th> </tr> </thead> <tbody> <tr> <td>Transmission document</td> <td>TRADACOMS Transmission</td> </tr> <tr> <td>Batch document</td> <td>TRADACOMS Batch</td> </tr> <tr> <td>File document</td> <td>A particular TRADACOMS File type and version, such as TRADACOMS 9 ORDHDR.</td> </tr> </tbody> </table>	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.
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	<p>The values of custom attributes</p> <p>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.</p>								

Use this criteria... **To have Trading Networks select a processing rule based on...**

For information about the custom attributes that Module for EDI sets, see [“Custom Attributes that Module for EDI Sets When Using TRADACOMS” on page 215](#). For information about how to set your own custom attributes, see [“Managing Your Own Custom Attributes for EDI Documents When Using TRADACOMS” on page 216](#).

Note: Module for EDI does not set the **User Status** system attribute, so do not use the **User Status** criterion.

Specifying Preprocessing Actions When Using TRADACOMS

Default preprocessing actions are defined in TN document types. However, you can override the settings in the processing rule. For more information about how the preprocessing actions apply to EDI documents, see the section about preprocessing actions in *webMethods Module for EDI Concepts Guide*.

The following table lists the preprocessing 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.

Preprocessing 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 preprocessing 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** preprocessing action because values for the **SignedBody** and **Signature** system attributes are not set for EDI documents.

Specifying Processing Actions When Using TRADACOMS

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 primarily use the **Execute a Service** action. You create the service that the **Execute a Service** processing action invokes.

In this scenario, the *TRADACOMS/splitOption* EDITPA variable must be set 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 When Using TRADACOMS” on page 221](#).
- There is no need to process the transmission document, so you can define a processing rule that has no processing actions selected, which in effect causes Trading Networks to ignore the document. Alternatively, you can 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 primarily use the **Deliver Document By** action, with the *TRADACOMS/splitOption* EDITPA variable set to Transmission. In this scenario, the only document sent to Trading Networks is the transmission document. For more information about delivering documents, see [“Forming EDI Documents to Send Outbound When Using Trading Networks” on page 257](#).

Coding Services to Process File and Batch Documents When Using TRADACOMS

The logic you design in your processing service (set in the **Execute a Service** action) depends on the types of documents that Module for EDI is splitting from the original EDI document. This information is based on what is specified in the *TRADACOMS/splitOption* EDITPA variable.

If *TRADACOMS/splitOption* Module for EDI creates... is...

- | | |
|------|--|
| File | File, batch, and transmission documents from the original EDI document. When the split document is of type: <ul style="list-style-type: none"> ■ File. 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, such as a back-end system. For information about how to create this service, see “Logic to Process a File Document” on page 224. |
|------|--|

If *TRADACOMS/splitOption* Module for EDI creates...
is...

- **Batch and transmission.** Do not create a service. All processing for these EDI documents is handled 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.

Batch

Batch and transmission documents from the original EDI document. When the split document is of type:

- **Batch.** 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 226.
- **Transmission.** Do not 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.

Transmission

Transmission documents from the original EDI document. You would set the *TRADACOMS/splitOption* to *Transmission* if you want to simply deliver the EDI document to its receiver.

For more information about how to deliver a document, see [“Delivering the Outbound EDI Document”](#) on page 264.

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 When Using TRADACOMS”](#) on page 222.

Information in the Pipeline that Your Service Can Access When Using TRADACOMS

When a transmission, batch, or file document is passed to the processing rule, the following information is accessible to your service from the pipeline:

- *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 more information about the BizDocEnvelope, see the *webMethods Trading Networks Built-In Services Reference* for your release.

The following table lists some of the variables within the BizDocEnvelope.

Variable	Description
<i>DocumentID</i>	The EDI transmission control number/file generation number from the transmission, batch, or file header as described in “DocumentID” on page 214 .
<i>Attributes</i>	The custom attributes that were set for the document. For information about the custom attributes that Module for EDI sets, see “Custom Attributes that Module for EDI Sets When Using TRADACOMS” on page 215 .
<i>Errors</i>	The errors that Trading Networks and Module for EDI 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 table shows the structure of the EDI document summary information in the *envelopeDocuments* variable.

Variable	Description								
<i>envelopeDocuments</i>	Document List Summary of the transmission document including the batch and file that it contains.								
	<table border="1"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>docID</i></td> <td>A String that contains the Internal ID that Trading Networks generated for the Transmission document.</td> </tr> <tr> <td><i>docTypeID</i></td> <td>A String that contains the internal ID of the TN document type used for the Transmission document.</td> </tr> <tr> <td><i>docTypeName</i></td> <td>A String that contains the name of the TN document type used for the Transmission document, for example, TRADACOMS Transmission.</td> </tr> </tbody> </table>	Key	Description	<i>docID</i>	A String that contains the Internal ID that Trading Networks generated for the Transmission document.	<i>docTypeID</i>	A String that contains the internal ID of the TN document type used for the Transmission document.	<i>docTypeName</i>	A String that contains the name of the TN document type used for the Transmission document, for example, TRADACOMS Transmission.
Key	Description								
<i>docID</i>	A String that contains the Internal ID that Trading Networks generated for the Transmission document.								
<i>docTypeID</i>	A String that contains the internal ID of the TN document type used for the Transmission document.								
<i>docTypeName</i>	A String that contains the name of the TN document type used for the Transmission document, for example, TRADACOMS Transmission.								

Variable	Description								
<i>groupDocuments</i>	(optional) A Document List that contains a summary of all the batch documents in the transmission.								
	<table><thead><tr><th>Key</th><th>Description</th></tr></thead><tbody><tr><td><i>docID</i></td><td>A String that contains the Internal ID that Trading Networks generated for the Batch document.</td></tr><tr><td><i>docTypeID</i></td><td>A String that contains the Internal ID of the TN document type used for the Batch document.</td></tr><tr><td><i>docTypeName</i></td><td>A String that contains the name of the TN document type used for the Batch document, for example, Tradacoms Batch.</td></tr></tbody></table>	Key	Description	<i>docID</i>	A String that contains the Internal ID that Trading Networks generated for the Batch document.	<i>docTypeID</i>	A String that contains the Internal ID of the TN document type used for the Batch document.	<i>docTypeName</i>	A String that contains the name of the TN document type used for the Batch document, for example, Tradacoms Batch.
Key	Description								
<i>docID</i>	A String that contains the Internal ID that Trading Networks generated for the Batch document.								
<i>docTypeID</i>	A String that contains the Internal ID of the TN document type used for the Batch document.								
<i>docTypeName</i>	A String that contains the name of the TN document type used for the Batch document, for example, Tradacoms Batch.								
<i>transactionDocuments</i>	(optional) A Document List that contains a summary of all the file documents in the batch.								
	<table><thead><tr><th>Key</th><th>Description</th></tr></thead><tbody><tr><td><i>docID</i></td><td>A String that contains the Internal ID that Trading Networks generated for the File document.</td></tr><tr><td><i>docTypeID</i></td><td>A String that contains the Internal ID of the TN document type used for the File document.</td></tr><tr><td><i>docTypeName</i></td><td>A String that contains the name of the TN document type used for the File document, for example, Tradacoms File.</td></tr></tbody></table>	Key	Description	<i>docID</i>	A String that contains the Internal ID that Trading Networks generated for the File document.	<i>docTypeID</i>	A String that contains the Internal ID of the TN document type used for the File document.	<i>docTypeName</i>	A String that contains the name of the TN document type used for the File document, for example, Tradacoms File.
Key	Description								
<i>docID</i>	A String that contains the Internal ID that Trading Networks generated for the File document.								
<i>docTypeID</i>	A String that contains the Internal ID of the TN document type used for the File document.								
<i>docTypeName</i>	A String that contains the name of the TN document type used for the File document, for example, Tradacoms File.								

Logic to Process a File Document

The following sample code 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 109](#).

```

1  => wm.b2b.edi.tradacoms.doc.isFileEnvelope
  └─ BRANCH on '/isFileEnvelope'
    └─ EXIT '$flow' and signal FAILURE
      └─ MAP
2  => wm.b2b.edi.tradacoms.doc.getFFSchemaNames
3  => wm.b2b.edi.tradacoms.doc.getDocumentPartInfo
4  => wm.b2b.edi.tradacoms.doc.getContentPart (Get Header Message Content Part)
5  => wm.b2b.edi.util.getContentPartDataAsStream
6  => wm.b2b.edi.util.convertToValues (Convert Header to IData)
7  └─ BRANCH on '/hasVAT' (Test to see if the file contains a VAT message)
    └─ true: SEQUENCE (File has a VAT Message)
      └─ wm.b2b.edi.tradacoms.doc.getContentPart (Get VAT content Part)
        └─ wm.b2b.edi.util.getContentPartDataAsStream
          └─ wm.b2b.edi.util.convertToValues (Convert VAT to IData)
    └─ $default: MAP (No VAT message)
8  => wm.b2b.edi.tradacoms.doc.getContentPart (Get VAT content Part)
  └─ wm.b2b.edi.util.getContentPartDataAsStream
    └─ wm.b2b.edi.util.convertToValues (Convert VAT to IData)
  └─ MAP
  └─ REPEAT (Loop over detail message)
9  └─ BRANCH on '/hasVAT' (Exit after looping over each detail message)
    └─ wm.b2b.edi.tradacoms.doc.getContentPart (Get Nth Detail Message)
      └─ wm.b2b.edi.util.getContentPartDataAsStream
        └─ wm.b2b.edi.util.convertToValues (Convert Nth Detail Message to Values)
          └─ pub.list.appendToDocumentList
10 └─ pub.math.addInts

```

Flow operation	Description
1	Invoke the EDI built-in service <code>wm.b2b.edi.tradacoms.doc.isFileEnvelope</code> to determine whether a BizDocEnvelope 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.getContentPart</code> service to return the content part object representing the header message.
5	Invoke the <code>wm.b2b.edi.util.getContentPartDataAsStream</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 (IData 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>getDocumentPartDataAsStream</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>getDocumentPartDataAsStream</code> , and <code>convertToValues</code> services, as you did for the header message.

Flow operation	Description
9	Parse the detail messages by executing a loop and invoking the <code>getContentPart</code> , <code>getDocumentPartDataAsStream</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 *webMethods Module for EDI 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, that is, a back-end system.

```

1 ➔ wm.b2b.edi.tradacoms.compose:startTradacomsTransmission
  ↳ MAP (Map Header Message)
2 ➔ wm.b2b.edi.tradacoms.compose:addToTradacomsTransmission
  ↳ MAP (Map Detail Message)
3 ➔ wm.b2b.edi.tradacoms.compose:addToTradacomsTransmission
  ↳ MAP (Map Trailer Message)
4 ➔ wm.b2b.edi.tradacoms.compose:addToTradacomsTransmission
5 ➔ wm.b2b.edi.tradacoms.compose:endTradacomsTransmission
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.

Flow operation	Description
6	Invoke the Integration Server <code>pub.io:streamToBytes</code> built-in service to convert the <code>InputStream</code> to bytes.
7	Invoke the Integration Server <code>pub.string:bytesToString</code> built-in service to convert the output of <code>pub.io:streamToBytes</code> to a string.

13 Optional Inbound Processing When Using Trading Networks

■ Overview	230
■ Validating Inbound Control Numbers	230
■ Defining Processing Rules to Handle Documents with Invalid Control Numbers	236
■ Reprocessing EDI Documents with Invalid Control Numbers	238
■ Automatically Generating Functional Acknowledgments	246
■ Automatically Generating Interchange Acknowledgments	253
■ Performing Optional Processing of Inbound EDI Documents	254

Overview

During inbound processing, you can have webMethods Module for EDI (Module for EDI) perform the following optional processing:

- Validate inbound interchange and/or group control numbers (or TRADACOMS transmission and/or batch control numbers). For more information, see [“Validating Inbound Control Numbers” on page 230](#) and [“Performing Control Number Validation” on page 233](#).
- Automatically generate functional acknowledgments (FAs). For more information, see [“Automatically Generating Functional Acknowledgments” on page 246](#).

Note:

Functional acknowledgments (FAs) are not applicable to the TRADACOMS and VDA standards.

- Automatically generate interchange acknowledgments (TA1). For more information, see [“Automatically Generating Interchange Acknowledgments” on page 253](#).

For information about defining control number information for your trading partners and setting up control number validation, including turning on and off control number validation, see [“Defining Control Number Information for Trading Partners” on page 113](#).

Validating Inbound Control Numbers

During inbound processing, Module for EDI 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. Validation determines whether the control numbers are in order, and therefore whether the EDI documents arrived in order.

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

For more information about the EDIControlNumber table, see [“Managing Control Number Information” on page 114](#).

Valid Control Numbers

Module for EDI considers the control number of an inbound EDI document to be valid when it matches the next expected control number in the EDIControlNumber table entry. How this process works depends on the EDI standard in use:

- **For VDA documents.** Unlike other EDI standards, the VDA message has two control number fields at the envelope level: the old control number and the new control number. (For this reason, no control number incrementing occurs.) If validation is turned on, Module for EDI extracts the old control number from the document and matches it against the one stored in the database. Then the module updates the EDIControlNumber table with the value of the

new control number field so that the next document's old control number field can be validated against the current document's new one.

For the ordering of VDA documents exchanged between trading partners, the first document must have 00000 as the old control number, and the new control number must be any positive number. For subsequent exchanges, the old control number of that message must match the new control number fields of the previous message.

If the new control number field for the inbound document is greater than the *control number maximum*, Module for EDI sets the value of the next expected control number as the *control number minimum*.

- **For all other standards.** Non-VDA documents contain a single control number, which Module for EDI uses to compare with the control number entry in the EDIControlNumber table. When Module for EDI determines a control number is valid, it calculates the next expected control number, saves it in the EDIControlNumber table, and continues processing the EDI document.

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

Invalid Control Numbers

Module for EDI considers the control number of an inbound EDI document to be invalid when it does not match the next expected control number in the EDIControlNumber table entry. Once a control number is determined to be invalid, the module determines whether the invalid control number is a duplicate control number or an out-of-sequence control number. Last, the module takes the action you define in [“Actions Module for EDI Can Take for Invalid Control Numbers”](#) on page 234.

Duplicate Control Numbers

A *duplicate control number* is a control number that Module for EDI believes has already been used and, therefore, might indicate a duplicate document. Generally, the module considers a control number to be a duplicate if the control number from the header is less than the next expected control number in the EDIControlNumber table (for example, the 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. Module for EDI 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 this example :

minimum	maximum	window
0	999999999	5

expected	duplicate control numbers
0	99999999, 99999998, 99999997, 99999996, 99999995
1	0, 99999999, 99999998, 99999997, 99999996
2	0, 1, 99999999, 99999998, 99999997
3	0, 1, 2, 99999999, 99999998
4	0, 1, 2, 3, 99999999
5	0, 1, 2, 3, 4

Note:

Because Module for EDI 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 set manually, the module might determine that a control number is:

- A duplicate, even though the control number was never received previously.
- Out-of-sequence, even though the control number has been received already.

You define the action that you want Module for EDI to take if a document with a duplicate or out-of-sequence control number arrives. For example, you can set the module to prevent processing of documents with duplicate or out-of-sequence control numbers. For more information, see [“Actions Module for EDI Can Take for Invalid Control Numbers”](#) on page 234. Later, you can force the processing of these documents. For more information, see [“Reprocessing EDI Documents with Invalid Control Numbers”](#) on page 238 and [“Reprocessing Documents with Out-of-Sequence Control Numbers”](#) on page 240.

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. A control number is also considered out of sequence if it is not numeric.

Generally, Module for EDI considers a control number to be out of sequence if it is greater than the expected control number. For example, the 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 maximum. Module for EDI 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 this example :

minimum	maximum	window
0	999	5

expected	out-of-sequence control numbers
995	996, 997, 998, 999, 0
996	997, 998, 999, 0, 1
997	998, 999, 0, 1, 2
998	999, 0, 1, 2, 3
999	0, 1, 2, 3, 4

Note: Module for EDI maintains only the next expected control number in the EDIControlNumber table, not a list of all previously used control numbers. Because the next expected control number

can be manually set, the module might determine that a control number is out of sequence even though a document with the control number has already been received.

Performing Control Number Validation

When Module for EDI receives an EDI document, the module performs the following steps:

Step	Description
1	<p>Module for EDI determines whether it is to validate the control number. To determine whether to validate control numbers in inbound EDI documents, the 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 121.</p> <p>If the module is to validate the control number, the module continues with the next step.</p>
2	<p>Using the sender/receiver, EDI standard/version, production mode, and type that are identified in the interchange or group header (or the transmission or batch header), the module locates the corresponding entry in the EDIControlNumber table.</p>
3	<p>Module for EDI 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 module maintains the next expected control number in the EDIControlNumber table. The control number is either valid or invalid, as follows:</p> <ul style="list-style-type: none"> ■ Valid control number. The control number value from the header matches the next expected control number in the EDIControlNumber table entry. <ul style="list-style-type: none"> If there is no EDIControlNumber entry for the sender/receiver, EDI standard/version, production mode, and type (for example, "Envelope" or group type) identified in the interchange or group header, the module: <ol style="list-style-type: none"> 1. Assumes the control number is valid. 2. Adds an entry to the EDIControlNumber entry for the sender/receiver, EDI standard/version, production mode, and type. In the EDIControlNumber entry, the module sets the control number maximum, minimum, increment, and window to their defaults and calculates the next expected control number and saves this number in the EDIControlNumber table entry. For more information, see “Validating Inbound Control Numbers” on page 230. 3. Sets the Trading Networks EDI Status custom attribute to <code>Processed</code>. ■ Invalid control number. The control number value from the header does not match the next expected control number in the EDIControlNumber table

Step **Description**

entry. When the control number is invalid, Module for EDI first determines whether the control number is a duplicate or out-of-sequence, and then takes the action you defined based on this information.

For more information about invalid control numbers, see [“Invalid Control Numbers” on page 231](#). For more information about the actions the module takes for invalid control numbers, see [“Actions Module for EDI Can Take for Invalid Control Numbers” on page 234](#).

Actions Module for EDI Can Take for Invalid Control Numbers

The actions that you can define for duplicate or out-of-sequence control numbers are:

Action	Where to Find More Information
Error & Continue	“Error & Continue” on page 234
Process Normally	“Process Normally” on page 235
Reject	“Reject” on page 235

For information about how to define the action you want Module for EDI to take, see [“Defining Actions Module for EDI Can Take for Invalid Control Numbers” on page 124](#).

Error & Continue

When you select `Error & Continue` and Module for EDI encounters an invalid control number, the module:

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 the documents that 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, UN/EDIFACT, or VDA: 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 [“Validating Inbound Control Numbers” on page 230](#).

4. Continues normal processing of the document.

Process Normally

When you select `Process Normally` and `Module` for EDI encounters an invalid control number, the module performs these steps:

1. Logs the warning to the Trading Networks activity log.

Note:

Because an error is not logged to the BizDocEnvelope, you cannot use the **Recognition Errors** criterion in a processing rule to select the documents that have errors.

2. Sets the next expected control number for the sender/receiver by incrementing the control number in the appropriate `EDIControlNumber` table entry. For more information, see [“Validating Inbound Control Numbers” on page 230](#).
3. Continues normal processing of the document.

Reject

When you select `Reject` and `Module` for EDI encounters an invalid control number, it does not perform normal processing on the document. Rather, the module:

1. Does not split the Interchange or Group document according to the setting of the *splitOption* EDITPA variable (or the Transmission or Batch document according to the setting of the *TRADACOMS/splitOption* EDITPA variable).

For ANSI X12, UN/EDIFACT, or VDA:

- If an interchange control number is invalid and the *splitOption* variable is set to `Group`, `Module` for EDI does not create Group documents.
- If an interchange control number is invalid and the *splitOption* variable is set to `Transaction`, `Module` for EDI does not create Group and Transaction documents.
- (ANSI X12 and UN/EDIFACT only) If a group control number is invalid and the *splitOption* variable is set to `Transaction`, `Module` for EDI does not create Group and Transaction documents for the group that has the invalid control number.

For more information about the *splitOption* variable, see [“splitOption Variable” on page 67](#).

For TRADACOMS:

- If a transmission control number is invalid and the *TRADACOMS/splitOption* variable is set to `Batch`, the `Module` for EDI does not create Batch documents.
- If a transmission control number is invalid and the *TRADACOMS/splitOption* variable is set to `File`, `Module` for EDI does not create Batch or File documents.
- If a batch control number is invalid and the *TRADACOMS/splitOption* variable is set to `File`, `Module` for EDI 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 [“TRADACOM Variables” on page 103](#).

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 **EDI Status** custom attribute 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.
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 **EDI Status** custom attribute 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 236](#).

Additionally, you can later force the processing of documents that contain invalid control numbers. For more information, see [“Reprocessing EDI Documents with Invalid Control Numbers” on page 238](#).

Note:

If the control number is invalid, Module for EDI does not increment the control number in the EDIControlNumber table.

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 [“Actions Module for EDI Can Take for Invalid Control Numbers” on page 234](#). For more information about the actions that Module for EDI takes when you set the action to **Reject**, see [“Reject” on page 235](#).

Module for EDI 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.

Setting the EDI Status Custom Attribute for Control Number Validation

Module for EDI sets the value of the **EDI Status** custom attribute for control number validation of inbound EDI documents.

Custom Attribute	Description
EDI Status	<p>Module for EDI sets this attribute to indicate whether a control number for a document is valid. If the module determines that a document contains:</p> <ul style="list-style-type: none"> ■ Valid control number, it sets this attribute to <code>Processed</code>. ■ 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>. <p>For more information about duplicate or out-of-sequence control numbers, see “Invalid Control Numbers” on page 231.</p>

For information about other Trading Networks document attributes that Module for EDI uses, see:

- For non-TRADACOMS users: [“Trading Networks Attributes and EDI Documents” on page 189](#).
- For TRADACOMS users: [“Trading Networks Attributes and EDI Documents When Using TRADACOMS” on page 214](#).

Defining the Processing Rule to Handle Documents with Invalid Control Numbers

In My webMethods, use the [Administration > Integration > B2B > Processing Rules](#) page to define a processing rule for handling EDI documents with invalid control numbers. When you define a processing rule, you define the criteria that you want Trading Networks to use to select the processing rule, the preprocessing actions, and the processing actions. For more information about creating processing rules, see the *webMethods Trading Networks Administrator’s Guide* for your release.

The following table provides details about how to define the processing rules.

Processing Rule Tab	Description						
Extended Criteria	<p>To select EDI documents that Module for EDI rejected because it detected an invalid control number and the action to take was set to <code>Reject</code>, 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> <p>To select documents with duplicate control numbers, specify:</p> <table border="1"> <thead> <tr> <th>Attribute</th> <th>Operator</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Attribute	Operator	Value			
Attribute	Operator	Value					

Processing Rule Tab	Description										
	<table border="1"> <tr> <td>EDI Status</td> <td>Equals</td> <td>Duplicate Control Number</td> </tr> </table> <p>To select documents with out-of-sequence control numbers, specify:</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>Out of Sequence Control Number</td> </tr> </tbody> </table>	EDI Status	Equals	Duplicate Control Number	Attribute	Operator	Value	EDI Status	Equals	Out of Sequence Control Number	
EDI Status	Equals	Duplicate Control Number									
Attribute	Operator	Value									
EDI Status	Equals	Out of Sequence Control Number									
	<p>Important: When specifying the value for extended criteria, be sure to use the exact combination of upper and lower case letters.</p>										
Pre-Processing	You should override the default preprocessing actions with the following:										
	<table border="1"> <thead> <tr> <th>Action</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Verify Digital Signature</td> <td>Do not verify digital signature</td> </tr> <tr> <td>Validate Structure</td> <td>Do not validate structure</td> </tr> <tr> <td>Check for Duplicate Document</td> <td>Do not use Trading Networks check for duplication</td> </tr> <tr> <td>Save Document to Database</td> <td>Save</td> </tr> </tbody> </table>	Action	Description	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
Action	Description										
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										
	<p>Note: If you do not select to save the content, Module for EDI 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 238.</p>										
Action	<p>Define the actions that you want to take to handle the EDI document containing the invalid control number. 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. 										

Reprocessing EDI Documents with Invalid Control Numbers

When 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 Module for EDI Can Take for Invalid Control Numbers](#)” on page 124. For more information about how Module for EDI processes documents when the action is Reject, see “[Reject](#)” on page 235.

The way you reprocess documents with invalid control numbers varies based on whether the invalid control number is a duplicate control number or an 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 Software AG Designer, invoke the `wm.b2b.editn.util.reprocess.listUnprocessDoc` service. When you invoke this service, specify `Duplicate` for the `type` input parameter. 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.listUnprocessDoc` service, use that internal ID to view the content of the document, as follows:
 - a. In My webMethods: **Monitoring > Integration > B2B > Transactions**
 - b. In the Transaction Search panel, select **Internal ID** from the search query field list, select the **Equals** operator, and paste the internal ID into the Value box.
 - c. Select **Search**.
 - d. Select a transaction row that the query returns and then select the **Content** tab to view contents of the document that had a duplicate control number.
3. If you want to reprocess a document, do one of the following:
 - In My webMethods, select **Reprocess**.
When you use this method, Module for EDI uses the default values for `generateFA` and `updateControlNumber`, described below.
 - In Designer, invoke the `wm.b2b.editn.util.reprocess.reprocessDocument` service and specify the following:

For this input parameter...	Specify...						
<i>internalID</i>	Trading Networks internal ID of the document you want to reprocess.						
<i>generateFA</i>	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: <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td><i>true</i></td> <td>Generate FAs. Specify <i>true</i> when reprocessing a document with a duplicate interchange control number.</td> </tr> <tr> <td><i>false</i></td> <td>Do not generate FAs. This is the default. Specify <i>false</i> when reprocessing a document with a duplicate group control number.</td> </tr> </tbody> </table>	Value	Meaning	<i>true</i>	Generate FAs. Specify <i>true</i> when reprocessing a document with a duplicate interchange control number.	<i>false</i>	Do not generate FAs. This is the default. Specify <i>false</i> when reprocessing a document with a duplicate group control number.
Value	Meaning						
<i>true</i>	Generate FAs. Specify <i>true</i> when reprocessing a document with a duplicate interchange control number.						
<i>false</i>	Do not generate FAs. This is the default. Specify <i>false</i> when reprocessing a document with a duplicate group control number.						
<i>updateControlNumber</i>	Whether you want to update the next expected control number in the EDIControlNumber table. Specify <i>false</i> (the default).						

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.

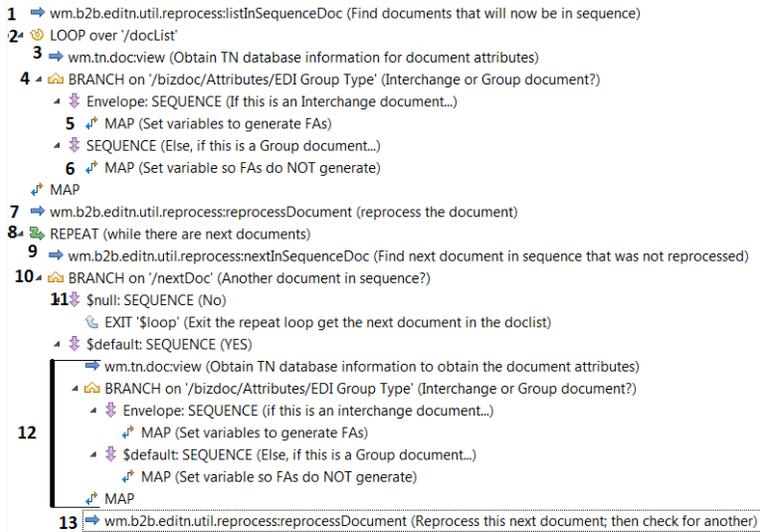
Out-of-sequence control numbers in EDI document might indicate that there were missing documents. Later, if the missing EDI documents arrive, you might want to force the processing of the out-of-sequence document. This time, because the missing EDI documents have arrived, the control number will no longer be out-of-sequence and Module for EDI 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 documents 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 240.

Rather than invoke the service you create manually, you should use Integration 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 *webMethods Integration Server Administrator's Guide* for your release.

Creating a Service to Reprocess Documents with Out-of-Sequence Control Numbers

The following diagram 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.



Flow operation	Description
----------------	-------------

1	Invoke the <code>wm.b2b.editm.util.reprocess:listInSequence</code> service to retrieve a list of EDI documents that had out-of-sequence control numbers, but now due to missing documents arriving, are in sequence.
---	--

For more information about the `listInSequence` service, see the *webMethods Module for EDI Built-In Services Reference*.

2	The <code>wm.b2b.editm.util.reprocess:listInSequence</code> service returns a list of Trading Networks internal IDs in the <code>docList</code> parameter. 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 Acknowledgments” on page 246 .
---	---

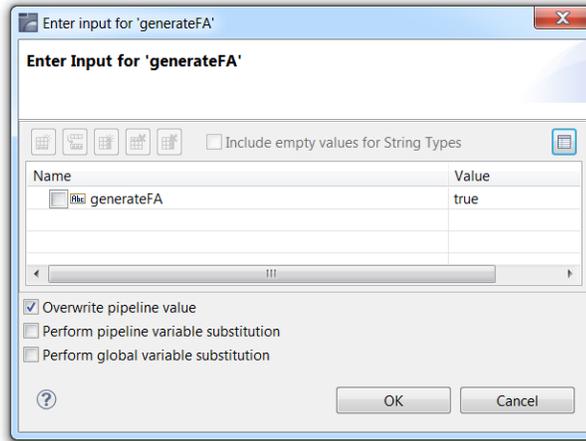
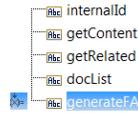
Invoke the `wm.tn.doc:view` service to retrieve information about the document. Specifically, the value of the **EDI Group Type** attribute is needed to determine whether the document is an Interchange or Group document.

When you invoke `wm.tn.doc:view`, set the following input parameters:

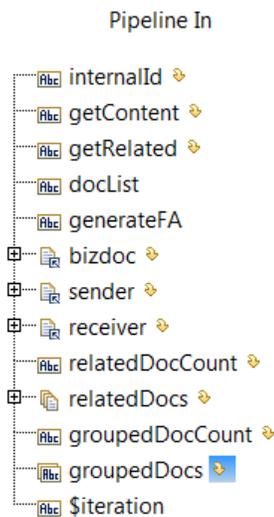
- Map the `docList` parameter under **Pipeline In** to `internalID` under **Service In**.
- Set the value of `getContent` under **Service In** to `false`.
- Set the value of `getRelated` under **Service In** to `false`.

Flow operation	Description
4	<p>The value of the EDI Group Type attribute is in the <i>bizdoc/Attributes/EDI Group Type</i> pipeline variable. To branch based on its value, set the Switch property to the following:</p> <pre data-bbox="386 380 1365 415">bizdoc/Attributes/EDI Group Type</pre> <ul style="list-style-type: none"> <li data-bbox="381 449 1338 512">■ When <i>bizdoc/Attributes/EDI Group Type</i> is <i>Envelope</i>, it is an Interchange document. <li data-bbox="381 541 1377 604">■ When <i>bizdoc/Attributes/EDI Group Type</i> is another value (for example, PO), it is a Group document.
5	<p>Add a MAP flow operation to set the value of the <i>generateFA</i> parameter, which is an input to the <i>wm.b2b.editn.util.reprocess:reprocessDocument</i> service.</p> <p>To do so, with the MAP flow operation selected on the Pipeline tab, add a <i>generateFA</i> parameter under Pipeline Out and set the value to one of the following:</p> <ul style="list-style-type: none"> <li data-bbox="381 863 1377 968">■ In the SEQUENCE for when the EDI Group Type attribute is <i>Envelope</i>, set the value to <i>true</i>, which indicates the <i>wm.b2b.editn.util.reprocess:reprocessDocument</i> service should generate the FAs. <li data-bbox="381 997 1377 1129">■ In the SEQUENCE for when the EDI Group Type attribute is another value, set the value to <i>false</i>, which indicates the <i>wm.b2b.editn.util.reprocess:reprocessDocument</i> service should not generate the FAs. <p>For example, the diagram shows the pipeline for the MAP when the EDI Group Type attribute is <i>Envelope</i> (that is, the <i>generateFA</i> parameter is set to <i>true</i>).</p>

Flow operation	Description
----------------	-------------



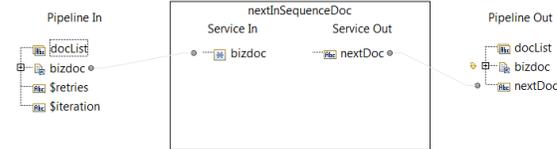
- 6 Use a MAP flow operation to clean up the pipeline by dropping all parameters related to invoking the `wm.tn.doc.view` service. Drop the following parameters: *internalID*, *getContent*, *getRelated*, *bizdoc*, *sender*, *receiver*, *relatedDocCount*, *relatedDocs*, *groupDocCount*, and *groupedDocs*.



- 7 Invoke the `wm.b2b.edi.tn.util.reprocess:reprocessDocument` service to reprocess an EDI document that is now in sequence.

Flow operation	Description
	<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>docList</i> parameter under Pipeline In to <i>internalID</i> under Service In. ■ If you added the <i>generateFA</i> parameter to the pipeline in the previous steps, the <i>generateFA</i> parameter exists under Pipeline In and is automatically mapped to the <i>generateFA</i> input parameter under Service In. If you are not using automatic FA generation, you can set the value of <i>generateFA</i> under Service In to <code>false</code>. ■ Set the value of the <i>updateControlNumber</i> parameter under Service In to <code>true</code>. This is necessary so the service updates the next expected control number in the EDIControlNumber table. <p>For more information about the <code>reprocessDocument</code> service, see <i>webMethods Module for EDI Built-In Services Reference</i>.</p>
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 and 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> <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 <code>-1</code>. ■ Set the Repeat On property to <code>SUCCESS</code>.
9	<p>Invoke the <code>wm.b2b.editm.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.editm.util.reprocess:reprocessDocument</code> service. The Pipeline In variable will automatically map to the Service In variable. You should drop the <i>bizdoc</i> parameter in Pipeline Out.</p>

Flow operation	Description
----------------	-------------



If the `nextInSequenceDoc` service locates a next document, it returns the Trading Networks internal ID of the document in the `nextDoc` parameter. Otherwise, the `nextDoc` parameter will be null.

For more information about the `nextInSequenceDoc` service, see *webMethods Module for EDI Built-In Services Reference*.

- | | |
|----|--|
| 10 | Use a BRANCH flow operation to branch based on the value of the <code>nextDoc</code> parameter, which is the result of the <code>nextInSequenceDoc</code> service. |
| 11 | If the value of <code>nextDoc</code> parameter is null, exit the REPEAT loop. |
| 12 | If the <code>nextDoc</code> parameter has a value, prepare to execute the <code>wm.b2b.editn.util.reprocess:reprocessDocument</code> service against this next document to reprocess it. |

The following flow operations are similar to steps 3 through 6 described above. Differences are noted below.

- INVOKE `wm.tn.doc:view`
 - When setting the inputs to this service, map the `nextDoc` parameter under **Pipeline In** to `internalID` under **Service In**.
- BRANCH on `bizdoc/Attributes/EDI Group Type`
- SEQUENCE for setting `generateFA` when EDI Group Type is Envelope
- SEQUENCE for setting `generateFA` when EDI Group Type has another value
- MAP to drop variables related to invoking `wm.tn.doc:view`

- | | |
|----|--|
| 13 | Invoke the <code>wm.b2b.editn.util.reprocess:reprocessDocument</code> service against the next document to reprocess it. |
|----|--|

To set the inputs for this service, with this flow operation selected, perform the following on the **Pipeline** tab:

- Map the `nextDoc` parameter under **Pipeline In** to `internalID` under **Service In**. You should drop the `nextDoc` parameter in **Pipeline In**.
- If you added the `generateFA` parameter to the pipeline in the previous steps, the `generateFA` parameter exists under **Pipeline In** and is automatically mapped to the input variable `generateFA` under **Service In**. If you are not

Flow operation	Description
	<p>using automatic FA generation, you can set the value of <i>generateFA</i> under Service In to <code>false</code>.</p> <ul style="list-style-type: none"> ■ Set the value of the <i>updateControlNumber</i> parameter 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.

Automatically Generating Functional Acknowledgments

Optionally, Module for EDI can automatically generate functional acknowledgments (FAs) for inbound documents during processing. After the FAs are generated, the module sends them to Trading Networks so they can be delivered to the sender of the inbound document. You must define a processing rule to deliver the FAs.

For more information about FAs, including the contents of a generated FA, see [“Generating Acknowledgments” on page 153](#). For more information about defining a processing rule to deliver FAs, see [“Defining a Processing Rule to Deliver FAs” on page 251](#).

Note:

Functional acknowledgments (FAs) are not applicable to the TRADACOMS and VDA standards.

The following table lists the tasks you must perform to set up automatic FA generation and includes a cross reference to the section in this guide that provides more information about each task.

Task	Where to Find More Information
Ensure you have installed the TN document types and flat file schemas for the FA (997 or CONTRL)	“Configuring webMethods Module for EDI” on page 21
Turn on automatic FA generation	“Turning Automatic FA Generation On and Off” on page 247
Set the level of detail to include in the generated FAs	“FAGeneration/FALevel EDITPA Variable” in “FAGeneration Variables” on page 75
Set how to generate control numbers for the FA	“FAGeneration/generateControlNumber EDITPA Variable” in “FAGeneration Variables” on page 75
Determine how Module for EDI sets the FA status for an interchange, group, or transaction within an EDI document	“Determining How Module for EDI Reports FA Status” on page 247
Define the action Module for EDI takes for an interchange, group, or transaction document based on its FA status	“Defining Actions Module for EDI Takes Based on FA Status” on page 249

Task	Where to Find More Information
Define one or more Trading Networks processing rules to perform normal processing on documents that have FA statuses defined as acceptable	<p>“Defining Processing Rules to Process Inbound EDI Documents” on page 198</p> <p>For more information about acceptable FA statuses, see “Defining FA Statuses as Acceptable and Unacceptable” on page 247.</p>
Define one or more Trading Networks processing rules to perform processing on documents that have FA statuses defined as unacceptable	<p>“Defining Processing Rules for Documents with Unacceptable FA Statuses” on page 252</p> <p>For more information about unacceptable FA statuses, see “Defining FA Statuses as Acceptable and Unacceptable” on page 247.</p>
Define a Trading Networks processing rule to deliver the generated FAs	“Defining a Processing Rule to Deliver FAs” on page 251

Turning Automatic FA Generation On and Off

You set the *FAGeneration/autoGenerateFA* EDITPA variable to indicate whether you want Module for EDI 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 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).

For information about setting the *FAGeneration/autoGenerateFA* EDITPA variable, see [“FAGeneration Variables”](#) on page 75.

Determining How Module for EDI Reports FA Status

Module for EDI uses the same information to determine the FA status for a transaction, group, or interchange of an inbound EDI document whether the FA is generated by the *generateFA* service or the *FAGeneration/autoGenerateFA* EDITPA variable. For more information about how the module determines FA status, see [“Determining How Module for EDI Reports FA Status”](#) on page 156.

Defining FA Statuses as Acceptable and Unacceptable

You use the *FAGeneration/processDocument* EDITPA variable to define FA statuses as acceptable or unacceptable. Module for EDI processes documents that have acceptable FA statuses normally and handles documents that have unacceptable FA statuses in a different manner. For more information about defining *FAGeneration/processDocument* EDITPA variable, see [“FAGeneration Variables”](#) on page 75.

The following table describes how Module for EDI processes a document based on the document's FA status and the setting of the *FAGeneration/processDocument* EDITPA variable.

When <i>processDocument</i> is set to...	and FA status of the document is...	Module for EDI...
All	Any of the following: <ul style="list-style-type: none"> ■ Not Allowed ■ Rejected ■ Partially Accepted ■ Accepted, But Errors Were Noted 	Processes the document with the acceptable FA status normally.
Only Accepted	Accepted	Processes the document with the acceptable FA status normally.
	Any of the following: <ul style="list-style-type: none"> ■ Not Allowed ■ Rejected ■ Partially Accepted ■ Accepted, But Errors Were Noted 	Handles the processing differently for these unacceptable FA statuses.
Not Rejected	Any of the following: <ul style="list-style-type: none"> ■ 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 Module for EDI processes the documents based on their FA statuses, see [“Defining Actions Module for EDI Takes Based on FA Status” on page 249](#).

Non-standard

Note:

When you are using non-standard processing, Module for EDI 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 Module for EDI home page. For more

information about accessing and using this screen, see [“Defining Interchange-Level Sender/Receiver Pair Information”](#) on page 335.

Defining Actions Module for EDI Takes Based on FA Status

Based on the *splitOption* EDITPA variable, Module for EDI can be working with an Interchange document, a Group document, or a Transaction document. For more information, see [“splitOption Variable”](#) on page 67.

- 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 [“Determining How Module for EDI Reports FA Status”](#) on page 247.

You set how you want Module for EDI to process a document based on its FA status. The module can:

- Process documents normally when they have FA statuses that you define as acceptable. For more information, see [“Processing Documents with Acceptable FA Statuses”](#) on page 249. For information about designating acceptable FA statuses, see [“Defining FA Statuses as Acceptable and Unacceptable”](#) on page 247.
- Handle documents differently when they have FA statuses that you define as unacceptable. For more information, see [“Processing Documents with Unacceptable FA Statuses”](#) on page 250. For information about designating unacceptable FA statuses, see [“Defining FA Statuses as Acceptable and Unacceptable”](#) on page 247.

Processing Documents with Acceptable FA Statuses

To process documents with acceptable FA statuses, Module for EDI performs the following steps:

Step	Description
1	Sets the EDI Status custom attribute for the Interchange, Group, or Transaction document to Processed.
2	Splits the document, if appropriate: <ul style="list-style-type: none"> ■ For an Interchange document <p>If the <i>splitOption</i> is Group or Transaction and the UN/EDIFACT document contains groups, Module for EDI creates Group documents and processes the Group documents.</p>

Step	Description
	<p>If the <i>splitOption</i> is <i>Group</i> or <i>Transaction</i> and the UN/EDIFACT document does not contain groups, Module for EDI creates Transaction documents and processes the Transaction documents.</p> <ul style="list-style-type: none"> ■ For a Group document <p>If the <i>splitOption</i> is <i>Transaction</i>, Module for EDI creates the Transaction documents and processes the Transaction documents.</p>
3	Sends the Interchange, Group, or Transaction document to Trading Networks processing rules.

Processing Documents with Unacceptable FA Statuses

To process documents with unacceptable FA statuses, Module for EDI performs the following steps:

Step	Description
1	<p>Sets the EDI Status custom attribute for the Interchange, Group, or Transaction document based on the documents FA status. When the FA status is:</p> <ul style="list-style-type: none"> ■ Not Allowed, the module sets the EDI Status custom attribute to Generate FA - Not Allowed ■ Rejected, the module sets the EDI Status custom attribute to Generate FA - Rejected ■ Partially Accepted, the module sets the EDI Status custom attribute to Generate FA - Partially Accepted ■ Accepted, But Errors Were Noted, the module sets the EDI Status custom attribute to Generate FA - Accepted, But Errors Were Noted
2	<ul style="list-style-type: none"> ■ 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 preprocessing 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 preprocessing option.
3	<p>Does not split an Interchange or Group document regardless of the setting of the EDITPA <i>splitOption</i> variable.</p> <ul style="list-style-type: none"> ■ For an Interchange document, if the <i>splitOption</i> is <i>Group</i> or <i>Transaction</i>, the module does not create Group or Transaction documents. ■ For a Group document, if the <i>splitOption</i> is <i>Transaction</i>, the module does not create Transaction document.

Step	Description
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 252.

Defining a Processing Rule to Deliver FAs

Use the following procedure to create a processing rule to deliver the outbound FA that Module for EDI generates.

➤ To define a processing rule to deliver outbound FAs

1. In My webMethods: **Administration > Integration > B2B > Processing Rules**.
2. On the Processing Rules Details page, complete the information based on the following table:

Processing Rule Tab	Description						
Criteria	<p>Use to select outbound FAs that Module for EDI generated, based on 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 “Output from the generateFA Service” on page 166. ■ 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 Envelope ■ In the Selected box, select the appropriate TN document type. 						
Extended Criteria	<p>Use to select documents based on the value of custom attributes. In this case, you select documents based on the value of the EDI Outbound FA custom attribute, as shown below.</p> <table border="1"> <thead> <tr> <th>Attribute</th> <th>Operator</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>EDI Outbound FA</td> <td>Equals</td> <td>true</td> </tr> </tbody> </table>	Attribute	Operator	Value	EDI Outbound FA	Equals	true
Attribute	Operator	Value					
EDI Outbound FA	Equals	true					
Action	<p>Use to specify how to deliver the outbound FA. Use either of the following processing rule actions:</p> <ul style="list-style-type: none"> ■ Execute a Service. Invokes the service you created to deliver the FA. 						

Processing Rule Tab	Description
	<ul style="list-style-type: none"> ■ Deliver Document By. Batches the outbound FA or delivers it to a VAN. For more information, see “Batching EDI Documents” on page 275 and “Retrieving and Delivering EDI Documents from and to VANs” on page 297.

For more information about defining processing rules, see the *webMethods Trading Networks Administrator’s Guide* for your release.

Defining Processing Rules for Documents with Unacceptable FA Statuses

Trading Networks provides the **EDI Status** custom attribute to indicate whether a document has an unacceptable FA status. When defining processing rules for documents with unacceptable FA statuses, use EDI Status in the criteria. For more information about how the **EDI Status** custom attribute is set, see information about handling documents with unacceptable FA statuses in [“Defining Actions Module for EDI Takes Based on FA Status” on page 249](#).

Use the information in the following table to assist you in defining a rule to process documents with an FA status of unacceptable. For more information about defining processing rules, see the *webMethods Trading Networks Administrator’s Guide* for your release.

Processing Rule Tab	Description
Extended Criteria	<p>Use 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 FA status of unacceptable, the EDI Status custom attribute will be set 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>The actual values available for the EDI Status custom attribute are based on how you set up automatic FA generation, specifically how you define acceptable and unacceptable FA statuses. For more information, see “Defining FA Statuses as Acceptable and Unacceptable” on page 247.</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, the value of the EDI Status custom attribute is set to Generate FA - Rejected. When specifying Value, be sure to use the exact combination of uppercase and lowercase characters.</p>

Processing Rule Tab	Description		
	Attribute	Operator	Value
	EDI Status	Equals	Generate FA - Rejected
	<p>Important: When you set up processing rules, you must set up a different processing rule for each unacceptable FA status. Do not 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 all 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>		
Action	<p>Use to define the actions that you want to take to handle EDI documents that have an FA status of unacceptable. 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. 		

Automatically Generating Interchange Acknowledgments

Module for EDI can optionally generate interchange acknowledgments (TA1s) automatically when a payload is submitted to Trading Networks.

➤ To automatically generate interchange acknowledgments

1. Set the `X12TA1Generation/autoGenerateTA1` EDITPA variable to on. The module generates an envelope containing the TA1 segment.

Note:

When both `autoGenerateTA1` and `autoGenerateFA` are turned on, the module creates two envelopes. The FA and the TA1 are not combined into one envelope.

For more information about the `X12TA1Generation/autoGenerateTA1` EDITPA variable, see [“FAGeneration Variables” on page 75](#).

2. Define a processing rule to deliver this envelope to the sender of the inbound document.

Performing Optional Processing of Inbound EDI Documents

When Module for EDI receives an EDI document, it performs the following steps:

Step	Description
1	The module parses an interchange from the EDI document.
2	<p>The module determines whether it is to validate control numbers. If the control numbers are:</p> <ul style="list-style-type: none"> ■ Not to be validated, the module continues with step 3. ■ To be validated, the module validates them. If the interchange control number is: <ul style="list-style-type: none"> ■ Invalid, the 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 the process stops at this step. ■ Valid, the process continues with the next step. <p>For more information about validating control numbers, see “Validating Inbound Control Numbers” on page 230.</p>
3	<p>Module for EDI determines whether it is to automatically generate FAs for the sender/receiver pair. If automatic FA Generation is:</p> <ul style="list-style-type: none"> ■ Off, the process stops at this step. ■ On, the process continues with the next step.
4	<ul style="list-style-type: none"> ■ For an ANSI X12 document, the module generates the FAs for the groups in the interchange. ■ For a UN/EDIFACT document, the module generates the FAs for the interchange.

Note:

The FAs are not sent to Trading Networks processing rules at this time.

In addition to generating the FAs during this step, Module for EDI 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 Module for EDI has information about the syntax errors and logical errors for transactions, it can also determine the child rejection error status for groups and interchanges.

Step	Description
5	<p>Module for EDI determines whether it is to automatically generate TA1s for the sender/receiver pair. If automatic TA1 generation is:</p> <ul style="list-style-type: none"> ■ Off, the process stops at this step. ■ On, the process continues with the next step.
6	<p>For an ANSI X12 document, the module generates the TA1s for the interchange.</p> <p>Note: The TA1s are not sent to Trading Networks processing rules at this time.</p>
7	<p>For ANSI X12 documents, if the interchange contains groups, Module for EDI works with each Group document that was split from the interchange, as follows:</p> <ol style="list-style-type: none"> 1. The module determines the FA status of each group and then uses this information to determine how to handle the Group document. For more information, see “Determining How Module for EDI Reports FA Status” on page 247. <ul style="list-style-type: none"> ■ If the group FA status is acceptable, the module processes the Group document normally. ■ If the group FA status is unacceptable, the module handles the Group document differently. <p>For more information, see “Defining Actions Module for EDI Takes Based on FA Status” on page 249.</p> 2. If the Group document is being processed normally, that is, it has an acceptable FA status, and the <i>splitOption</i> EDITPA variable is defined as Transaction, Module for EDI works with each Transaction document that was split from the group, as follows: <ol style="list-style-type: none"> a. The module determines the FA status of the transaction and then uses this information to determine how to handle the Transaction document. For more information, see “Defining Actions Module for EDI Takes Based on FA Status” on page 249. <ul style="list-style-type: none"> ■ If the transaction FA status is acceptable, the module processes the Transaction document normally. ■ If the transaction FA status is unacceptable, the module handles the Transaction document differently. <p>For more information, see “Defining Actions Module for EDI Takes Based on FA Status” on page 249.</p> 3. Module for EDI sends all Transaction and Group documents that are processed normally to Trading Networks processing rules.

Step	Description
8	<p>For a UN/EDIFACT document, if the <i>splitOption</i> is Transaction, Module for EDI works with each Transaction document that was split from the interchange (that is, transactions that are not within group envelopes).</p> <ol style="list-style-type: none"> 1. Module for EDI determines the FA status of the Transaction and then uses this information to determine how to handle the Transaction document. <ul style="list-style-type: none"> ■ If the FA status of the transaction is acceptable, the module processes the Transaction document normally. ■ If the FA status of the transaction is unacceptable, the module handles the Transaction document differently. <p>For more information, see “Defining FA Statuses as Acceptable and Unacceptable” on page 247 and “Defining Actions Module for EDI Takes Based on FA Status” on page 249.</p> 2. If the Transaction document is processed normally, the module sends it to Trading Networks processing rules.
9	<p>Module for EDI sends the generated FAs to Trading Networks processing rules.</p> <p>When the FAs are processed through Trading Networks, the EDI recognizer updates the EDITRACKING table for the FAs. Module for EDI uses the EDITRACKING table to track FAs and related groups/transactions for FA reconciliation. For more information about FA reconciliation, see “Reconciling Functional Acknowledgments” on page 303.</p>
10	<p>Module for EDI sends the Interchange document to Trading Networks processing rules.</p>

14 Forming EDI Documents to Send Outbound When Using Trading Networks

■ Overview	258
■ Preparing Trading Networks to Process Internal-Format Documents	258
■ Creating the Service to Form the EDI Document	258
■ Delivering the Outbound EDI Document	264

Overview

You can send documents to webMethods Trading Networks (Trading Networks) from internal applications (for example, back-end systems) and map information from them into a standard EDI format document. When you use Trading Networks with webMethods Module for EDI (Module for EDI), when forming the EDI documents, you can access information for delimiters and headers that Module for EDI maintains in the Trading Networks database.

This chapter describes how to:

- Prepare Trading Networks to process internal-format documents. For more information, see [“Preparing Trading Networks to Process Internal-Format Documents” on page 258](#).
- Create a service to form the EDI document from an internal-format document. For more information, see [“Creating the Service to Form the EDI Document” on page 258](#).
- Deliver the outbound EDI document. For more information, see [“Delivering the Outbound EDI Document” on page 264](#).

To learn more about forming outbound EDI documents when you are using Trading Networks with Module for EDI, see *webMethods Module for EDI Concepts Guide*.

Preparing Trading Networks to Process Internal-Format Documents

To prepare 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 information about defining profiles, see the *webMethods Trading Networks Administrator’s Guide* for your release.
- **TN document type for the internal-format document.** For information about defining TN document types, see the *webMethods Trading Networks Administrator’s Guide* for your release.
- **Processing rule for the internal-format document.** For information about defining processing rules, see the *webMethods Trading Networks Administrator’s Guide* for your release. 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 258](#).

After forming the EDI document, you can deliver it. For more information, see [“Delivering the Outbound EDI Document” on page 264](#).

Creating the Service to Form the EDI Document

You can send documents to Trading Networks from internal applications (for example, back-end systems) and create a service that maps information from the internal-format documents into an EDI document. Module for EDI provides built-in services that you can use to form the EDI document.

Additionally, because you are using Trading Networks, you can have the module maintain the following information that you can access and use when forming 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:	EDITPA
<ul style="list-style-type: none"> ■ Authorization information qualifier ■ Authorization information ■ Security information qualifier ■ Security information 	
Control numbers to use in group and interchange headers (or batch, transmission, and file headers)	EDIControlNumber table

Non-standard
Note:

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 Module for EDI home page. For more information about defining this information, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#). For more information about the difference between standard and non-standard processing, see *webMethods Module for EDI Concepts Guide*.

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 Module for EDI installed the flat file schema as well. If you did not install the TN document type for this EDI document, see [“Defining TN EDI Document Types” on page 34](#).
- 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 Software AG Designer 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 information about creating IS document types, see the *webMethods Service Development Help* for your release.
- Trading partner profiles of the internal-format document. For information about creating profiles, see the *webMethods Trading Networks Administrator's Guide* for your release.
- Trading partner profiles for sender and receiver of the outbound EDI document. For information about creating profiles for senders and receivers of EDI documents, see [“Defining Trading Partner Profiles” on page 55](#).
- Default and, optionally, partner-specific EDITPAs for the sender/receiver pairs in the outbound EDI document. For information about creating EDITPAs, see [“Defining EDI Trading Partner Agreements” on page 59](#).

Non-standard**Note:**

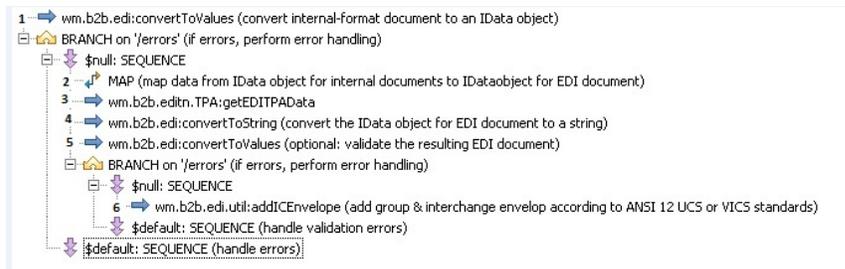
For ANSI X12 and UN/EDIFACT users: When using non-standard processing, you also have to define: 1) Interchange sender/receiver pair information and 2) Group-level sender/receiver pair associations. For more information, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#) and [“Defining Group-Level Sender/Receiver Pair Associations” on page 346](#). For more information about the difference between standard and non-standard processing, see *webMethods Module for EDI Concepts Guide*.

Creating the Service

Non-standard**Note:**

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 354](#).

The following sample code illustrates how to form an ANSI X12 EDI document from an internal-format document when using Trading Networks with Module for EDI.



Flow operation Description

- 1** Invoke the `wm.b2b.edi:convertToValues` service (or the `wm.b2b.edi.tradacoms:convertToValues` service for TRADACOMS documents) 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 parameters of the `convertToValues` service to have it validate the structure of the internal-format document.

The inputs to the `convertToValues` 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 an internal-format document rather than a flat file schema; however, it is recommended that you use flat file schemas. For more information about the `convertToValues` service, see *webMethods Module for EDI Built-In Services Reference*.

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 362.

- 2** 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.
- 3** Invoke the `wm.b2b.edi.tn.TPA:getEDITPAData` 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 `getEDITPAData` service, see *webMethods Module for EDI Built-In Services Reference*.
- 4** Invoke the `wm.b2b.edi:convertToString` service (or the `wm.b2b.edi.tradacoms:convertToString` service for TRADACOMS) to convert the EDI document from an `IData` object to `String` format.

The inputs to the `convertToString` service include:

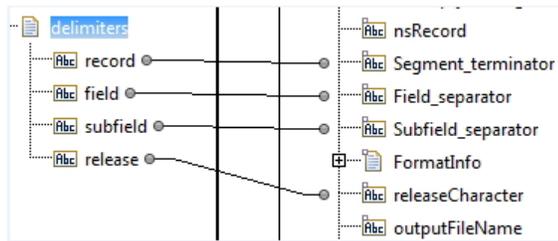
- The `IData` object that contains the data for your EDI document. Map this `IData` object to the *values* input parameter of the `convertToString` service.

Flow operation	Description
----------------	-------------

- 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.

- The delimiters to use when creating the EDI document. Map the delimiters from the EDITPA to the inputs of the `convertToString` service:



- 5 Optionally, invoke the `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 *string* output parameter from the `convertToString` service contains the EDI document. Map this to the *edidata* input parameter of the `convertToValues` service.
- The flat file schema that defines the structure for the EDI document.

- 6 Invoke the `wm.b2b.edi.util:addICEEnvelope` service to add the interchange and group envelope to the EDI document. If you are creating a UN/EDIFACT EDI document, use the `wm.b2b.edi.util:addICEEnvelopeEDIFACT` service. For more information, see [“Adding UN/EDIFACT Envelopes” on page 362](#).

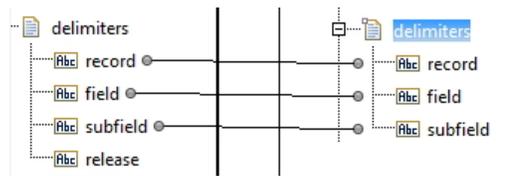
Note:

If you are creating a TRADACOMS EDI document, use the services in the `wm.b2b.edi.tradacoms.compose` folder to add a transmission envelope and a batch envelope (if specified) and to obtain control numbers. For more information, see [“Adding TRADACOMS Envelopes and Obtaining Control Numbers” on page 264](#).

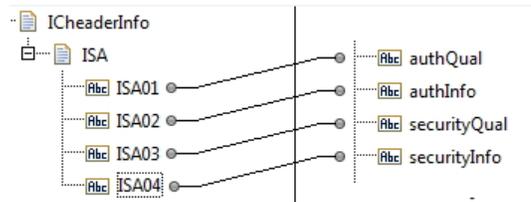
When setting the inputs to the `addICEEnvelope`:

- Map the delimiters from the EDITPA to the inputs of the `addICEEnvelope` service:

Flow operation Description



- Map the information from the EDITPA to the inputs of the addICEEnvelope service:



- Set the *ctlFromTable* parameter to `true` to have the service obtain the control number. For more information, see [“Obtaining Control Numbers for Outbound Processing \(ANSI X12 and UN/EDIFACT\)”](#) on page 263.

For information about delivering the document, see [“Delivering the Outbound EDI Document”](#) on page 264.

Obtaining Control Numbers for Outbound Processing (ANSI X12 and UN/EDIFACT)

Module for EDI maintains information about control numbers in the EDIControlNumber table, which is a Module for EDI-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 are provided with Module for EDI 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:addICEEnvelope	wm.b2b.edi.util:addICEEnvelopeEDIFACT

When invoking one of the above services to add a header, use the *ctlFromTable* or *grpCtlNumber* input parameter 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 Module for EDI home page. For instructions, see [“Configuring Control Number Settings”](#) on page 116.

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`

1. Invoke this service to add the header message to the TRADACOMS transmission object that the `startTradacomsTransmission` service returned.
2. Invoke the `addTradacomsTransmission` service a second time to add the detail messages to the TRADACOMS transmission object.
3. Invoke the `addTradacomsTransmission` service a third time 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.

- `wm.b2b.edi.tn.batch:getControlNumber` (provided in the WmEDIforTN package)

Invoke this service to obtain the current control number from the EDIControlNumber table. This table is a Module for EDI-specific table in the Trading Networks database.

For more information about these services, see *webMethods Module for EDI Built-In Services Reference*.

Delivering the Outbound EDI Document

This section describes three methods you can use to deliver the outbound EDI document:

- Add logic to the service that forms the EDI document to deliver it. For more information, see [“Delivering the EDI Document Directly from the Service that Forms It” on page 265](#).
- 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 265](#).
- Route the EDI document back to Trading Networks processing rules to use a processing rule to deliver the document. For more information, see [“Routing the Outbound EDI Document to Trading Networks” on page 270](#).

To learn more about these three methods for delivering outbound EDI documents, see *webMethods Module for EDI Concepts Guide*.

Delivering 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 an EDI document directly from the service that forms the document and you want to use the functional acknowledgment (FA) reconciliation feature of Module for EDI, you must also invoke the `wm.b2b.editn:trackEDIdocs` service.

Note:

Functional acknowledgments (FAs) are not applicable to the TRADACOMS and VDA standards.

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 about the `FAReconciliation` EDITPA Variable, see [“FAReconciliation Variable” on page 68](#). For more information about how to generate FA reconciliation reports, see [“Reconciling Functional Acknowledgments” on page 303](#).

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 258](#), with additional flow operations added to invoke:

- The `wm.b2b.editn:trackEDIdocs` service for FA reconciliation
- A service that you create to deliver the EDI document

```

➔ wm.b2b.edi.util:addJCEnvelope (add group & interchange envelope according to ANSI 12 UCS or VICS standards)
➔ wm.b2b.editn: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 Module for EDI-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

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 188](#).

Note that the EDI recognizer will automatically perform FA reconciliation if enabled by the `FAReconciliation` EDITPA variable. For more information, see [“FAReconciliation Variable” on](#)

page 68. For more information about how to generate FA reconciliation reports, see [“Reconciling Functional Acknowledgments” on page 303](#).

The following table lists the EDITPA variables that you should set differently when they are used for outbound EDI documents. If you are using TRADACOMS, follow the instructions in [“Specifying Variables that Affect Outbound Processing When Using TRADACOMS” on page 267](#).

EDITPA variable	Description
<i>splitOption</i>	<p>The EDI recognizer will split the EDI document based on the value of the <i>splitOption</i> EDITPA variable. For an outbound document, set the <i>splitOption</i> variable to one of the following:</p> <ul style="list-style-type: none"> <li data-bbox="456 611 1385 674">■ Interchange—The EDI document is delivered and FA reconciliation is not performed. <li data-bbox="456 705 1385 768">■ Group—The EDI document is delivered and FA reconciliation is performed. <div data-bbox="456 789 1365 957" style="background-color: #f0f0f0; padding: 5px;"> <p>Note: When the <i>splitOption</i> variable is set to Interchange but the <i>FAReconciliation</i> variable is set to true to enable FA reconciliation, the EDI recognizer will split at the group level.</p> </div> <p>For more information about this EDITPA variable, see “splitOption Variable” on page 67.</p>
<i>ControlNumberManagement/validateInterchangeControlNumbers</i>	<p>For an outbound document, set this EDITPA variable to false so Module for EDI does not attempt to validate the interchange control numbers in the outbound document.</p> <p>For more information about this EDITPA variable, see “ControlNumberManagement Variables” on page 83.</p>
Non-standard	<p>When you are using non-standard processing, Module for EDI 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 Module for EDI home page.</p> <p>For more information about this screen, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 335.</p>
<i>ControlNumberManagement/validateGroupControlNumbers</i>	<p>For an outbound document, set this EDITPA variable to false so Module for EDI does not attempt to validate the group control numbers in the outbound document.</p> <p>For more information about this EDITPA variable, see “ControlNumberManagement Variables” on page 83.</p>
<i>EAGeneration/autoGenerateFA</i>	<p>For an outbound document, set this EDITPA variable to Off so Module for EDI does not automatically generate FAs for your outbound document.</p>

Specifying Variables that Affect Outbound Processing When Using 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 When Using TRADACOMS” on page 213](#).

You should set the following EDITPA variables for outbound EDI documents:

- *TRADACOMS/splitOption*
- *TRADACOMS/OutBound/createReconciliationMessage*
- *TRADACOMS/OutBound/recipientTransmissionReference*
- *TRADACOMS/OutBound/applicationReference*
- *TRADACOMS/OutBound/applicationReferenceFromFile*

For more information about these variables, see [“Defining Trading Partner Information When Using TRADACOMS” on page 97](#).

Preparing 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 when using standards other than TRADACOMS, see [“Defining Trading Partner Profiles” on page 55](#)
 - For instructions when using TRADACOMS, see [“Defining Trading Partner Profiles When Using TRADACOMS” on page 99](#)
 - 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 Administrator’s Guide* for your release.
 - Processing rule for the Interchange (or Transmission) document. For instructions about how to define processing rules, see the *webMethods Trading Networks Administrator’s Guide* for your release. When you define the processing rule, use one of the following:
 - **Execute a Service** processing action to invoke the service you created to deliver the EDI document.
 - **Deliver Document By** processing action to batch the outbound EDI document or deliver it to a VAN. For more information, see [“Batching EDI Documents” on page 275](#), and [“Retrieving and Delivering EDI Documents from and to VANs” on page 297](#).
- 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 when using standards other than TRADACOMS, see [“Defining Trading Partner Profiles” on page 55](#)
 - For instructions when using TRADACOMS, see [“Defining Trading Partner Profiles When Using TRADACOMS” on page 99](#)
- 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 defining TN document types, see the *webMethods Trading Networks Administrator’s Guide* for your release.
- Processing rule for the Interchange and Group (or Transmission and Batch) documents. For instructions about defining processing rules, see the *webMethods Trading Networks Administrator’s Guide* for your release. When you define the processing rule for the Interchange (or Batch) document, use one of the following:
 - **Execute a Service** processing action to invoke the service you created to deliver the EDI document.
 - **Deliver Document By** processing action to batch the outbound EDI document or deliver it to a VAN. For more information, see [“Batching EDI Documents” on page 275](#), and [“Retrieving and Delivering EDI Documents from and to VANs” on page 297](#).

Adding Logic to the Service that Forms the EDI Document to Submit 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 258](#), with an additional INVOKE flow operation added to invoke the `wm.tn.doc.xml:routeXml` service.

```

→ wm.b2b.edi.util:addICEEnvelope (add group & interchange envelope according to ANSI 12 UCS or VICS standards)
→ wm.tn.doc.xml:routeXml (send the document to TN recognition)
↕ MAP (Drop variables associated with delivering the document)

```

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` parameter in **Pipeline In**. Previous operations in this service placed data in an `edidata` parameter. The following describes how to map an ANSI X12 document to deliver into the `edidata` parameter without losing the value of the current `edidata` parameter.

Step Description

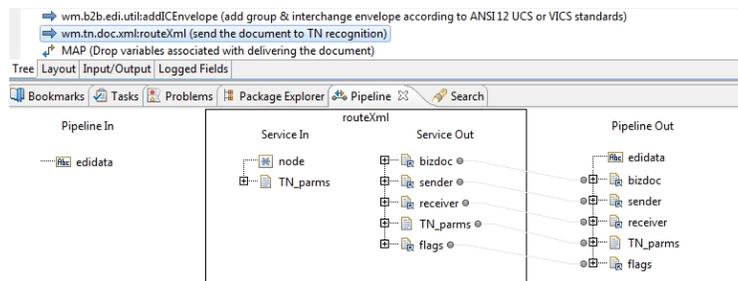
- 1 Select the **INVOKE `wm.b2b.edi.util:addICEEnvelope`** flow operation.

Step Description

- 2 In **Pipeline Out**, create a document. (In the figure below, the new document is named DocToDeliver.) Within the document, create a string named *edidata*. Then map the output from the `addICEnvelope` service, which is in the *outDocument* parameter, to your new *edidata* parameter, as shown:

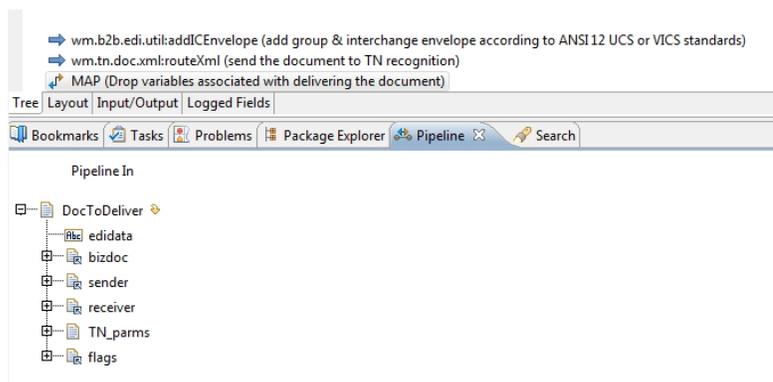


- 3 Click the **INVOKE `wm.tn.doc.xml:routeXml`** flow operation and view its properties. In the Properties panel, set the **Scope** property to the name of the document that you created to hold the output from the `addICEnvelope` service (for example, `DocToDeliver`). This causes the **INVOKE `wm.tn.doc.xml:routeXml`** flow operation to execute using only the pipeline variables within the specified scope document.
- 4 After setting the scope, when you select the **INVOKE `wm.tn.doc.xml:routeXml`** flow operation, **Pipeline In** contains only the *edidata* parameter, as illustrated in the figure below.



- 5 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 (for example, `DocToDeliver`). After invoking the `wm.tn.doc.xml:routeXml` service, you can drop all parameters in the document that you created, as shown:

Step	Description
------	-------------



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. Because the result of document recognition is a BizDocEnvelope that is sent to the processing rules and you are bypassing this step, 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 271](#).

Additionally, when you route a document through Trading Networks, if you want to use the functional acknowledgment (FA) reconciliation feature of Module for EDI, you must invoke the `wm.b2b.editn:trackEDIdocs` service. This service updates the EDITTRACKING 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 Variable” on page 68](#). For more information about how to generate FA reconciliation reports, see [“Reconciling Functional Acknowledgments” on page 303](#).

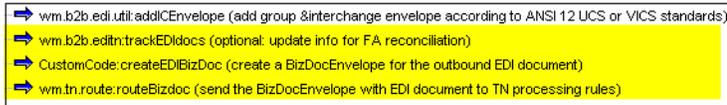
Adding Logic to the Service that Forms the EDI Document that Routes through Trading Networks

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 258](#), with additional flow operations added to invoke:

- The `wm.b2b.editn: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 271](#).

- The `wm.tn.route:routeBizdoc` service to send the `BizDocEnvelope` to the Trading Networks 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 the Service that Forms the EDI Document that Routes through Trading Networks](#)” on page 270.

The service creates a `BizDocEnvelope` and adds the outbound EDI document to the `BizDocEnvelope` as a content part.

The following table lists the input parameters for the `createEDIBizDoc` service:

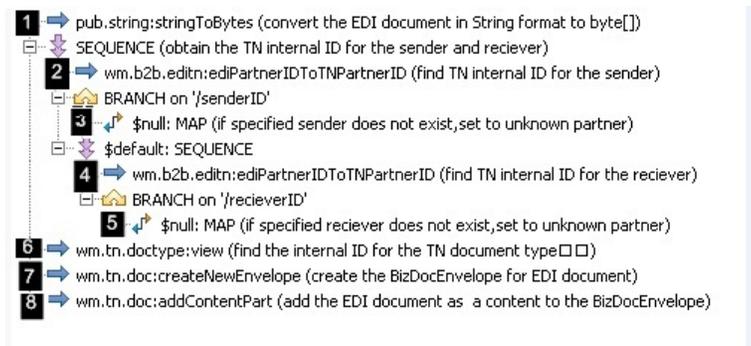
Input Parameter	Description
<i>OutboundEDIDoc</i>	The outbound EDI document in string format.
<i>sender</i>	The sender ID for the sender of the outbound EDI document. This is the sender at the interchange or transmission level.
<i>senderQualifier</i>	The EDI ID qualifier for the sender, for example, 01 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, 01 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"> ■ x12 for an ANSI X12 document ■ UNEDIFACT for a UN/EDIFACT document ■ VDA for a VDA document ■ TRADACOMS for a TRADACOMS document
<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 (for example, ISO-8859-1) that is used as input to the <code>pub.string:stringToBytes</code> service and then used to convert the EDI document into <code>byte[]</code> format. To use the

Input Parameter	Description
	default encoding, set this value to autoDetect. If you specify an unsupported encoding, an exception will be thrown.

The following table shows the output parameter for the createEDIBizDoc service:

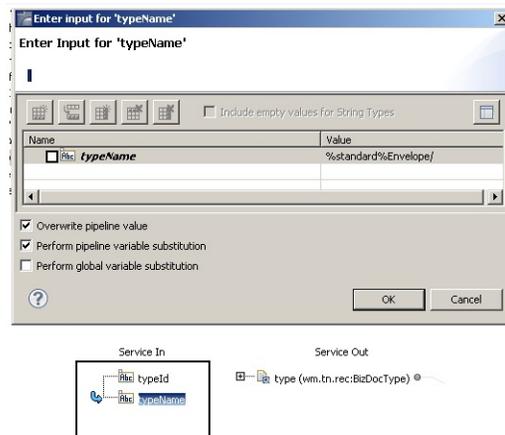
Output Parameter	Description
<i>bizdoc</i>	The BizDocEnvelope for the outbound EDI document. This is a Document Reference to the IS document type, <i>wm.tn.rec:BizDocEnvelope</i> .

The following figure illustrates the flow operations used for the service that creates a BizDocEnvelope for the outbound EDI document.



Flow operation	Description
1	<p>Invoke the <i>pub.string:stringToBytes</i> service to convert the outbound EDI document from string format to a <i>byte[]</i>. Map the input parameters of the <i>createBizDoc</i> service, <i>OutboundEDIDoc</i> and <i>encoding</i>, to the input parameters of the <i>stringToBytes</i> service, <i>string</i> and <i>encoding</i>, respectively.</p> <p>Map the output of the <i>stringToBytes</i> service (bytes).</p> <p>Convert the format to a <i>byte[]</i> because the service to add the content part to the <i>BizDocEnvelope</i> requires the content part to be in <i>byte[]</i> format.</p>
2	<p>Invoke the <i>wm.b2b.edittn:ediPartnerIDToTNPartnerID</i> service to retrieve the Trading Networks internal ID for the sender. Map the input parameters of the <i>createBizDoc</i> service, <i>sender</i> and <i>senderQualifier</i>, to the input parameters of the <i>ediPartnerIDToTNPartnerID</i> service, <i>identifier</i> and <i>qualifier</i>, respectively.</p> <p>Map the output of the <i>ediPartnerIDToTNPartnerID</i> service (<i>id</i>), to <i>senderID</i>, which is a parameter 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 00000000000000000000000000 (24 zeros), which indicates the Unknown partner.</p>

Flow operation	Description
4	<p>Invoke the <code>wm.b2b.editn:ediPartnerIDToTNPartnerID</code> service to retrieve the Trading Networks internal ID for the receiver. Map the input parameters of the <code>createBizDoc</code> service, <code>receiver</code> and <code>receiverQualifier</code>, to the input parameters of the <code>ediPartnerIDToTNPartnerID</code> service, <code>identifier</code> and <code>qualifier</code>, respectively.</p> <p>Map the output of the <code>ediPartnerIDToTNPartnerID</code> service, <code>id</code>, to <code>receiverID</code>, which is a parameter that you define in Pipeline Out.</p>
5	<p>Use a BRANCH operation to branch based on whether the <code>receiverID</code> is null. If <code>receiverID</code> is null, use a MAP flow operation to set the value of <code>receiverID</code> to 00000000000000000000000000 (24 zeros), which indicates the Unknown partner.</p>
6	<p>Invoke the <code>wm.tn.doctype: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.</p> <p>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 <code>typeName</code> input parameter for the <code>view</code> service as follows:</p>

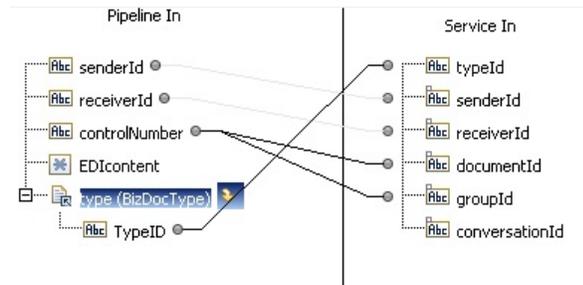


The variable substitution `%standard%` picks up the value of the `standard` input parameter of the `createBizDoc` service, which should be X12, UNEDIFACT, or TRADACOMS.

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> parameters, to the Service In variables, <code>senderID</code> and <code>receiverID</code>, respectively.
---	--

Flow operation	Description
----------------	-------------

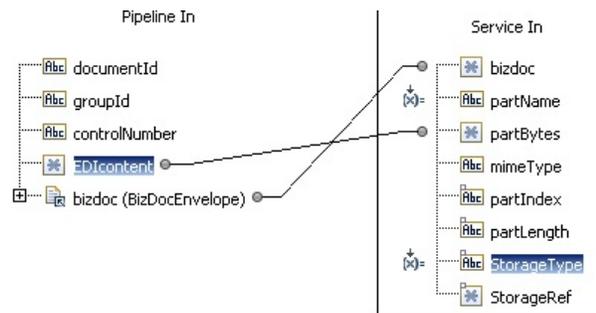
- The *controlNumber* input parameter of the `createBizDoc` service, which is in **Pipeline In**, to the **Service In** variables, *documentID* and *groupID*.
- The Trading Networks internal ID for the TN document type, which is in **Pipeline In** in the *type/TypeID* variable to the **Service In** *typeID* parameter.



8

Invoke the `wm.tn.doc:addContentPart` service to add the outbound EDI document (in `byte[]` format) to the `BizDocEnvelope`. To set the inputs to the `addContentPart` service:

- Map the content of the outbound EDI document that is in `byte[]` format, which is in **Pipeline In** in the *EDIcontent* parameter to the **Service In** *partBytes* parameter.
- Set the value of the **Service In** *partName* parameter to `EDIdata`.
- Set the value of the **Service In** *storageType* parameter to `database`.



15 **Batching EDI Documents**

■ Overview	276
■ Indicating How Many Batch EDI Documents to Create	276
■ How Documents Are Combined to Create the Batch EDI Document	277
■ Extracting Transactions and Sorting the Batched Documents	277
■ Recombining the Transactions into the Batch EDI Document	284
■ Before You Can Batch EDI Documents	289
■ Controlling How the batchProcess Service Forms the Batch Document	290
■ Preparing to Batch EDI Documents	291
■ Delivering the Batch EDI Document	293
■ Viewing Batched EDI Documents	294

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” in *webMethods Module for EDI Concepts Guide*.

p

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. 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.edi.bn.batch:batchProcess` service, which is provided with webMethods Module for EDI (Module for EDI). The `batchProcess` service acts on the EDI documents in the scheduled delivery queue to create a batch EDI document.

Trading Networks uses large document handling for batching EDI documents. For more information about large document handling, see [“Handling Large Documents When Using Trading Networks” on page 149](#).

Note:

You cannot batch VDA documents.

To learn more about the basics of batching EDI documents, see the *webMethods Module for EDI Concepts Guide*.

Indicating How Many Batch EDI Documents to Create

Use the `batchProcess` service `oneBatchQueue` input parameter to indicate how many output batch EDI documents you want the `batchProcess` service to create. You can set the `oneBatchQueue` parameter 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 Module for EDI. For more information, see the *webMethods Trading Networks Built-In Services Reference* for your release.

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. For more information, see [“Extracting Transactions and Sorting the Batched Documents” on page 277](#).
2. Recombines the transactions or messages (or TRADACOMS files) into the batch EDI document(s). For more information, see [“Recombining the Transactions into the Batch EDI Document” on page 284](#).

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 list describes how the `batchProcess` service processes the EDI documents in the queue when the documents have interchange headers or TRADACOMS transmission headers. 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 280](#).

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 290](#).
3. The `batchProcess` service sorts the transactions. For more information, see [“Sorting Transactions in the Queued Documents” on page 277](#).
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 for Failed Tasks” on page 283](#).

Sorting Transactions in the Queued Documents

Module for EDI sorts the transactions in EDI documents in the batch queue. This section describes how Module for EDI sorts the EDI documents when they have interchange or transmission headers. For information about how Module for EDI sorts documents when they do not have interchange

or transmission headers, see [“When EDI Documents Have No Interchange or Transmission Header”](#) on page 280.

Collection Areas Representing an Interchange in the Final Batch EDI Document

You can think of 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 (for example, Production or Testing)
- Each collection area becomes an interchange or transmission segment in the final batch EDI document.

Subcollections Representing a Group in the Final Batch EDI Document

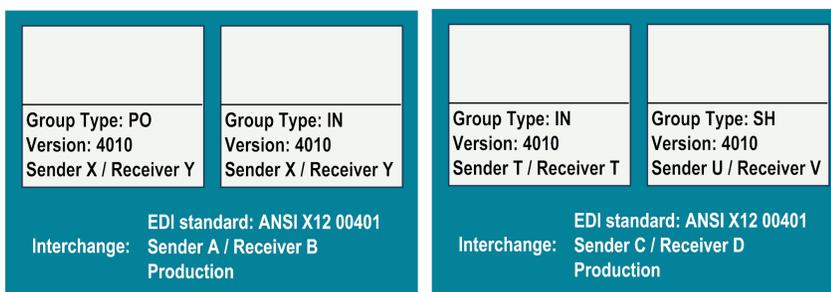
Within a collection area are *subcollections* that are associated with the following items, which the `batchProcess` service obtains from the group header (or TRADACOMS batch header) of the document.

- Each subcollection area has the same:
 - Group type (or TRADACOMS batch type)
 - Version of the EDI standard (for example, 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.

The following figure illustrates an 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.

The delimiters for the collection area are determined as follows:

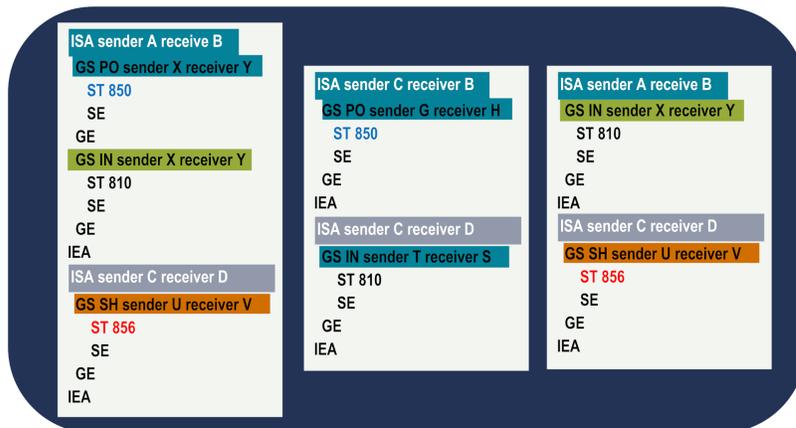
1. The batchProcess service uses the value from its *delimiters* input parameter.
2. If you do not specify a value for the *delimiters* input parameter, the batchProcess service uses the sender and receiver from the interchange or transmission header to locate the EDITPA values and uses the EDITPA *delimiters* variables.
3. 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 282](#).

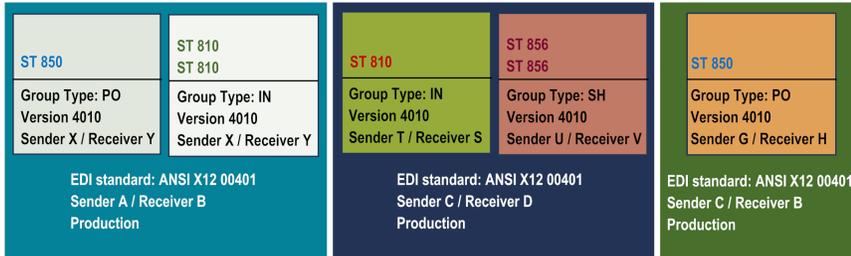
Using the interchange or transmission header from the original EDI document, the batchProcess service determines whether the transaction uses the same delimiters as those associated with the collection area. If the transaction uses the same delimiters, the batchProcess service places the document in 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.



The following figure shows how the transactions from the above three documents are 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 in [“How Documents Are Combined to Create the Batch EDI Document” on page 277](#), the `batchProcess` service typically uses information from the interchange or transmission headers when sorting the documents. The following lists 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 parameters to determine 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 290](#).
3. The `batchProcess` service sorts the transactions. For more information, see [“Sorting Transactions when Queued Documents Have No Interchange Headers” on page 280](#).
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 for Failed Tasks” on page 283](#).

Sorting Transactions when Queued Documents Have No Interchange Headers

This section describes how Module for EDI sorts the transactions of queued EDI documents when they have no interchange or transmission headers. For information about how Module for EDI sorts transactions when the EDI documents have interchange or transmission headers, see [“Sorting Transactions in the Queued Documents” on page 277](#).

Default Collection Area (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 parameters:

Item	From Where It Is Obtained
EDI standard	The <i>standard</i> input parameter 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	The <i>version</i> input parameter 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	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 (for example, Production or Testing)	The <i>environment</i> input parameter 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 (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 Representing a Group in the Final Batch EDI Document” on page 278](#). If the EDI document does not have group headers, it uses the following:

Item	From Where It Is Obtained
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
Version of the EDI standard (for example, 4010)	Obtained from the <i>version</i> input parameter 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.

The delimiters for the default collection area are determined as follows:

1. The batchProcess service uses the values from its *delimiters* input parameter.

2. If you do not specify a value for the *delimiters* input parameter, the `batchProcess` service uses its *senderIDQualifier*, *senderID*, *receiverIDQualifier*, and *receiverID* input parameters to locate the EDITPA values and uses the EDITPA *delimiters* variables.
3. 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 282.

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; therefore, the service 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, Module for EDI 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 for Failed Tasks”](#) on page 283.

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 parameter. If you specify the `batchProcess` service *delimiters* input parameter, 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.

- The *delimiters* EDITPA variables. If you leave the `batchProcess` service *delimiters* input parameter null, the `batchProcess` service uses the *delimiters* EDITPA variable to determine the delimiters to use for a collection area. Leave the `batchProcess` service *delimiters* input parameter 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, as follows:

- 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 parameters to locate the EDITPA.
- Module for EDI defaults for delimiters. If the `batchProcess` service cannot obtain delimiters from either its *delimiters* input parameter or the *delimiters* EDITPA variable, 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	* (asterisk)	+ (plus sign)
Subfield	: (colon)	: (colon)
Release character	(not used by ANSI X12)	? (question mark)

Updating the Task Status and Publishing Documents for Failed Tasks

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 Monitoring > Integration B2B > Tasks page in My webMethods. 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.

Task Status	Description
fail	<p>If the <code>batchProcess</code> service cannot 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 not be included in the final batch EDI document.</p> <p>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 true. For more information, ANSI X12 and UN/EDIFACT users should see “publishBatchFailEvent Variable” on page 70 and TRADACOMS users should see “TRADACOMS/publishBatchFailEvent EDITPA Variable” in “TRADACOM Variables” on page 103.</p> <p>The format of the IS document is defined by the <code>wm.b2b.editn.rec:batchFailRecord</code> IS document type. To view the format of this IS document, see the description of the <code>wm.b2b.editn.rec:batchFailRecord</code> IS document type in the <i>webMethods Module for EDI Built-In Services Reference</i>.</p> <p>To handle the failure, you can use Software AG Designer to create Integration Server triggers that subscribe to the published documents. For details, see the <i>Publish-Subscribe Developer’s Guide</i> for your release.</p>

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 (for example, ISA01, ISA02, UNB01, UNB07) that the `batchProcess` service obtains from the `ICheaderInfo` EDITPA 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. For information about the `ICheaderInfo` variables, see [“ICheaderInfo Variables” on page 73](#).
- 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 `ICheaderInfo/GS/GS07` EDITPA 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. For information about the `ICheaderInfo` variables, see [“ICheaderInfo Variables” on page 73](#).
- 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 acknowledgment (that is, 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 Variable” on page 68](#). For more information about FA reconciliation, see [“Reconciling Functional Acknowledgments” on page 303](#).

Note:

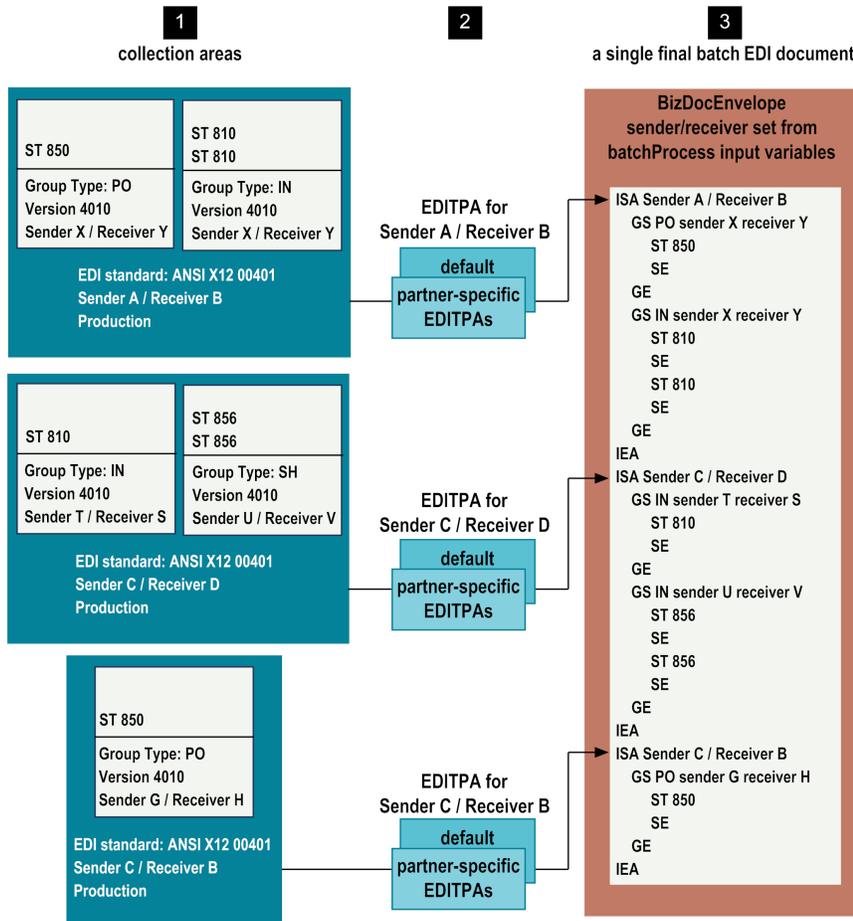
Functional acknowledgments (FAs) are not applicable to TRADACOMS.

- Creates one or more output batch EDI documents. The `batchProcess` service determines how many batch EDI documents to create based on the `oneBatchQueue` input parameter. 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 276](#).

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 293](#).

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 parameter 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.



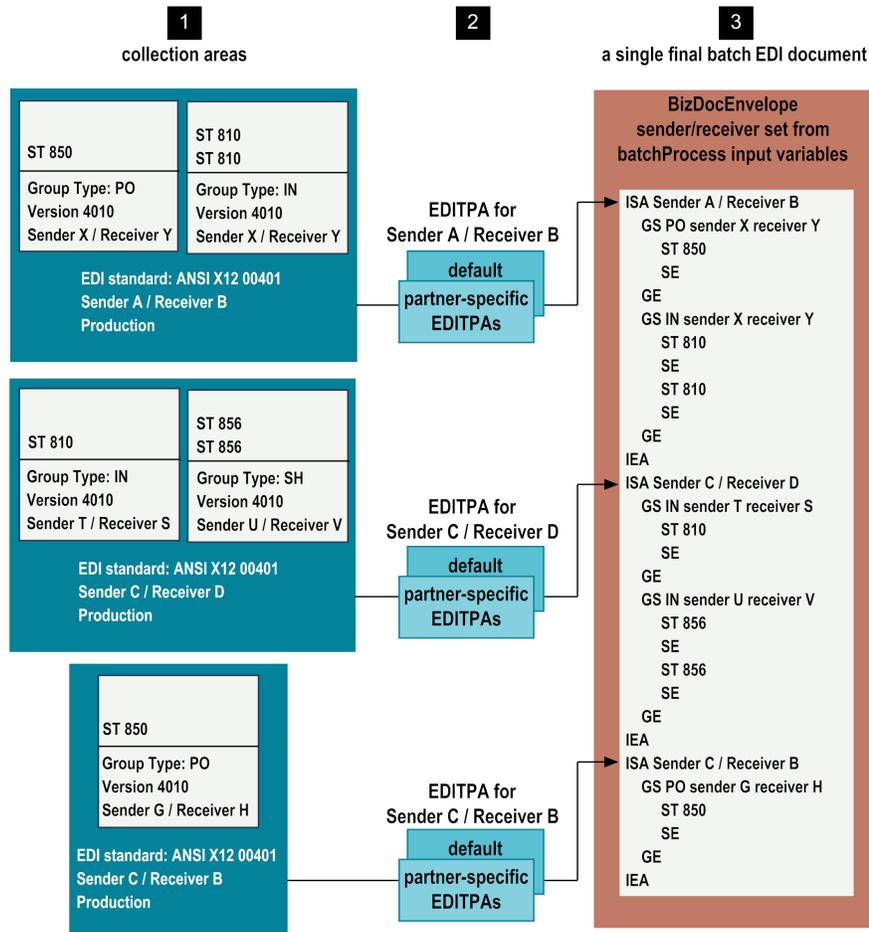
Step	Description
------	-------------

- | | |
|---|---|
| 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 associated with the collection area. It uses information associated with the collection area, the subcollections, and the EDITPA to create the interchange and group headers as follows:</p> <ul style="list-style-type: none"> ■ Uses the sender/receiver, EDI standard, production mode, and delimiters associated with the collection area for the interchange headers. ■ Obtains fields for the interchange header from the EDITPA <i>ICHeaderInfo</i> variables. If an <i>ICHeaderInfo</i> EDITPA variable does not have a value, the batchProcess service uses the corresponding value from the interchange header of the original document. For more information about the <i>ICHeaderInfo</i> variables, see “ICHeaderInfo Variables” on page 73. ■ Uses the sender/receiver, group type, version of the EDI standard associated with a subcollection for a group header. |

Step	Description
	<ul style="list-style-type: none"> 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 Variables” on page 73.
3	<p>After recombining the information for all collection areas, the <code>batchProcess</code> service creates a <code>BizDocEnvelope</code> for the final batch EDI document. The <code>batchProcess</code> service sets the sender and receiver for the <code>BizDocEnvelope</code> to the sender/receiver identified in its <code>senderIDQualifier</code>, <code>senderID</code>, <code>receiverIDQualifier</code>, and <code>receiverID</code> input parameters. It then sends the <code>BizDocEnvelope</code> to Trading Networks processing rules for routing.</p> <p>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 <code>BizDocEnvelope</code>.</p>

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 parameter is `MULTIPLEOUTPUTS`. The `batchProcess` service combines the transactions from each collection area into a single batch EDI document that contains a single interchange. The result is multiple output batch EDI documents.



Step	Description
1	For each collection area, the <code>batchProcess</code> 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	As when <code>oneBatchQueue</code> is <code>SINGLEOUTPUT</code> , when <code>oneBatchQueue</code> is <code>MULTIPLEOUTPUTS</code> , the <code>batchProcess</code> 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 subcollections, and the EDITPA to create the interchange and group headers. For more information, see step 2 in “Creating the Batch EDI Document when Using SINGLEOUTPUT” on page 285.
3	For each collection area, after recombining the transactions in the collection area, the <code>batchProcess</code> service creates a <code>BizDocEnvelope</code> for the final batch EDI document. The <code>batchProcess</code> service sets the sender and receiver for the <code>BizDocEnvelope</code> to the sender/receiver associated with the collection area. Then it sends the <code>BizDocEnvelope</code> to Trading Networks processing rules for routing.

Note:

Step	Description
	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 Module for EDI can batch EDI documents, you must perform the following high-level steps.

Step	Description	Where to Find More Information
1	Install the TN document types for the EDI documents that you will place in the scheduled delivery queues.	“Defining TN EDI Document Types” on page 34
2	Install the TN document types for the final batch EDI document. You need to install the TN document types for the envelope (for example, the TN document type X12 Envelope) and the group or TRADACOMS batch (for example, the TN document type X12 Group).	“Defining TN EDI Document Types” on page 34
3	Define the default and, optionally, the partner-specific EDITPAs for the interchange or transmission sender/receiver pairs.	<ul style="list-style-type: none"> ■ ANSI X12 and UN/EDIFACT users: “Defining EDI Trading Partner Agreements” on page 59 ■ TRADACOMS users: “Defining EDI Trading Partner Agreements When Using TRADACOMS” on page 101 ■ For more information about the EDITPA variables that affect batching EDI documents: “EDITPA Variables that the batchProcess Service Uses” on page 290.
4	Configure the <code>watt.server.tspace.timeToLive</code> property. Because Trading Networks uses large document handling for EDI batching, the <code>watt.server.tspace.timeToLive</code> property must be set so that Integration Server can access the file from Tspace and store it in the Trading Networks database.	“Configuring Module for EDI for Large Document Handling” on page 150

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 parameters of the batchProcess service. You define the input parameters 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 parameters that you specify. For more information, see [“Defining the Scheduled Delivery Queues” on page 292](#).
- Defining EDITPA variables that are used during batch processing. For more information, see [“EDITPA Variables that the batchProcess Service Uses” on page 290](#). For more information about EDITPAs in general, ANSI X12 and UN/EDIFACT users should see [“Defining EDI Trading Partner Agreements” on page 59](#) and TRADACOMS users should see [“Defining EDI Trading Partner Agreements When Using TRADACOMS” on page 101](#).

EDITPA Variables that the batchProcess Service Uses

The batchProcess service uses EDITPA variables 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 Transactions

The following table lists the EDITPA variables that the batchProcess service uses when sorting the transactions.

EDITPA Variable	Description
<i>publishBatchFailEvent</i>	The batchProcess service uses this variable to determine whether you want the service to publish an IS document to notify you when it cannot 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 Variable” on page 70 and “Updating the Task Status and Publishing Documents for Failed Tasks” on page 283 .
<i>delimiters</i>	ANSI X12 and UN/EDIFACT only. The batchProcess service uses this variable 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 282 .

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	Description
<i>UNAmode</i>	The batchProcess service uses this variable to determine whether to create a UNA segment prior to the interchange in the final batch EDI document. The <i>UNAmode</i> variable is specific to UN/EDIFACT. For more information, see “UNAmode Variable” on page 69 .
<i>ICheaderInfo</i>	<p>The batchProcess service uses these variables to construct the interchange 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.</p> <p>For more information about the <i>ICheaderInfo</i> variables, see “ICheaderInfo Variables” on page 73.</p> <p>Note: If the <i>ICheaderInfo</i> variables do not have a value, the batchProcess service uses the values from the headers in the original document.</p>

Preparing to Batch EDI Documents

The following table describes the tasks you must perform to batch EDI documents, as well as where to find more information about each task.

Task	Description	Where to Find More Information
1	Determine the number of scheduled delivery queues you need to define.	“Determining the Number of Queues You Need to Define” on page 291
2	Define the scheduled delivery queues for batching EDI documents.	“Defining the Scheduled Delivery Queues” on page 292
3	Define the processing rules to deliver documents to the scheduled delivery queues.	“Defining Processing Rules to Batch EDI Documents” on page 293

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 MCI VAN, and a third queue for batch data directed to a specific company.

Defining the Scheduled Delivery Queues

To create a scheduled delivery queue, define a public queue in My webMethods. The following table describes information that you need to supply when defining the queue.

For more information about creating public queues, see the *webMethods Trading Networks Administrator's Guide* for your release.

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 Module for EDI, the Module automatically registers the <code>wm.b2b.editn.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 .
Inputs	The inputs for the <code>batchProcess</code> service. When Trading Networks invokes the <code>batchProcess</code> service, Trading Networks passes the inputs that you specify in these input parameters. For more information, see the description of the <code>wm.b2b.editn.batch:batchProcess</code> service in <i>webMethods Module for EDI Built-In Services Reference</i> .
State	The state of the queue. Select Enable , Disable , Drain , or Suspend Delivery . For more information about these states, see the <i>webMethods Trading Networks Administrator's Guide</i> for your release.
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: It is recommended 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

Use My webMethods to 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 following table provides specific information.

For more detailed information about creating processing rules, see the *webMethods Trading Networks Administrator's Guide* for your release.

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> <ol style="list-style-type: none"> 1. Select Deliver Document By processing action. 2. For the Deliver Document By processing action, select Scheduled Delivery. 3. 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

In My webMethods, use the Administration > Integration > B2B > Processing Rules page to define a processing rule in Trading Networks. The following table provides details about how to define the processing rules.

On this Processing Specify... Rules tab...

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> <div style="background-color: #f0f0f0; padding: 5px;"> <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 <i>oneBatchQueue</i> input parameter is SINGLEOUTPUT or MULTIPLEOUTPUTS. For more information, see step 3 in “Creating the Batch EDI Document when Using SINGLEOUTPUT” on page 285 and “Creating the Batch EDI Document when Using MULTIPLEOUTPUTS” on page 287.</p> </div>						
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" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Attribute</th> <th style="text-align: left;">Operator</th> <th style="text-align: left;">Value</th> </tr> </thead> <tbody> <tr> <td>EDI Batch</td> <td>Equals</td> <td>Interchange</td> </tr> </tbody> </table>	Attribute	Operator	Value	EDI Batch	Equals	Interchange
Attribute	Operator	Value					
EDI Batch	Equals	Interchange					
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 “Preparing to Deliver Documents to VANs” on page 299. 						

For more information about creating processing rules, see the *webMethods Trading Networks Administrator's Guide* for your release.

Viewing Batched EDI Documents

Trading Networks can show the relationship between EDI batched documents and the individual documents that were used to create them. The name of this relationship is *EDI Batch*.

> To view the relationship between EDI batched documents and individual documents

1. In My webMethods: **Monitoring > Integration > B2B > Transactions**

2. Search for the EDI batch document for which you want to view the related documents. For information about searching documents, see the *webMethods Trading Networks User's Guide* for your release.
3. In the search results, locate the EDI batch document for which you want to view related documents and then click  or **View Related Documents**.

Trading Networks displays the Related Document page.

16 Retrieving and Delivering EDI Documents from and to VANs

■ Overview	298
■ Preparing to Retrieve Documents from a VAN	298
■ Preparing to Deliver Documents to VANs	299
■ Handling Conventions Required by Specific VANs	301
■ Customizing Built-In Services to Connect to Another VAN	301

Overview

When using webMethods Module for EDI (Module for EDI) with webMethods Trading Networks (Trading Networks), you can connect to Value Added Networks (VANs) to retrieve inbound EDI documents and to deliver outbound EDI documents.

Software AG has tested and certified Module for EDI to connect with the GXS and MCI VANs. If you need to connect to another VAN, you might need to customize the services that Module for EDI provides to suit the specific VAN connectivity.

To learn more about VANs and how Module for EDI works with VANS, see the *webMethods Trading Networks Administrator's Guide* for your release.

Preparing to Retrieve Documents from a VAN

To retrieve inbound EDI documents from a VAN, invoke the `VAN.VANConnectivity.getFromVAN` service. You can schedule this service to run when 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 Integration Server Administrator to set up a user task, see the *webMethods Integration Server Administrator's Guide* for your release.

The input parameters 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 parameters to the `getFromVAN` service that you can use to specify optional actions that you might want the service to perform in addition to retrieving the inbound EDI documents. To learn about all the input parameters to this service, see *webMethods Module for EDI Built-In Services Reference*.

Use this input parameter...	To have the <code>getFromVAN</code> service perform this optional action...
<code>saveInboundToTN</code>	Submit the retrieved inbound EDI documents to Trading Networks for processing. If you set the <code>saveInboundToTN</code> parameter to yes, the retrieved inbound EDI documents are processed through Trading Networks like any other inbound EDI documents.
<code>getReport</code>	Get VAN-generated reports after receiving the inbound EDI documents.

In addition to, or instead of, invoking 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 [“Preparing to Deliver Documents to VANs” on page 299](#).

Note:

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.

Preparing to Deliver Documents to VANs

To send outbound EDI documents to a VAN, you define a:

- Scheduled delivery queue in Trading Networks 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 My webMethods. You define one queue for each VAN to which you want to deliver documents. For example, if you want to connect to GXS and MCI, you must define a public queue for GXS and MCI. 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. Module for EDI 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 a state. Define whether Trading Networks can add delivery tasks into a queue and deliver the documents that correspond to the delivery tasks in the queues.
- 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.

The following table describes information that you need to supply when defining the public queue in My webMethods.

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 MCI VAN, you might name the queue "MCI Queue".						
Delivery service	The delivery service you want to associate with the queue. Select VANFTP .						
Inputs	<p>The inputs for the VANFTP service. When Trading Networks invokes the VANFTP service, Trading Networks passes the inputs that you specify in the input parameters.</p> <p>The input parameters 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 parameters 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 all the input parameters to the VANFTP service, see the description of the VAN.VANConnectivity.putToVAN service in <i>webMethods Module for EDI Built-In Services Reference</i>.</p> <table border="1"> <thead> <tr> <th>Use this input parameter...</th> <th>To have the VANFTP service perform this optional action...</th> </tr> </thead> <tbody> <tr> <td><i>getReport</i></td> <td>Get VAN-generated reports after delivering the outbound EDI documents.</td> </tr> <tr> <td><i>getInbound</i></td> <td>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.</td> </tr> </tbody> </table>	Use this input parameter...	To have the VANFTP service perform this optional action...	<i>getReport</i>	Get VAN-generated reports after delivering the outbound EDI documents.	<i>getInbound</i>	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.
Use this input parameter...	To have the VANFTP service perform this optional action...						
<i>getReport</i>	Get VAN-generated reports after delivering the outbound EDI documents.						
<i>getInbound</i>	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.						
State	The state of the queue. Select Enable , Disable , Drain , or Suspend Delivery . For more information about these states, see the <i>webMethods Trading Networks Administrator's Guide</i> for your release.						
Schedule	When you want Trading Networks to invoke the VANFTP service to deliver the EDI documents in the queue you are defining.						

Note:

You should 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 processing rules that instruct Trading Networks to place the outbound EDI document into a scheduled delivery queue that is associated with the VANFTP service.

Use My webMethods to define a processing rule in Trading Networks. For more information, see the *webMethods Trading Networks Administrator's Guide* for your release.

- 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 parameters to the VAN:VANConnectivity:getFromVAN service to send EDI documents to a VAN, as well as when setting the input parameters 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.
- The availability and variety of reports, as well as the ways in which you access them, differs from VAN to VAN.

Customizing Built-In Services to Connect to Another VAN

When connecting to a VAN other than GXS, ICC.NET, or MCI, always begin by attempting to connect to the VAN using the information described in [“Preparing to Retrieve Documents from a VAN” on page 298](#). If you encounter problems executing the VAN.VANConnectivity:getFromVAN service and/or the VAN.VANConnectivity:putToVAN (VANFTP) service, you need to create customized services.

Create your customized services using the VAN.VANConnectivity:getFromVAN and the VAN:VANConnectivity:putToVAN (VANFTP) services as templates. Be sure to give the customized services different names. Your customized services should use the FTP connectivity services that are provided in the VAN.VANConnectivity folder of the WmEDIforTN package.

- 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 Integration Server Administrator to set up a user task, see the *webMethods Integration Server Administrator's Guide* for your release.
- If you created a customized service to replace the VAN.VANConnectivity:putToVAN (VANFTP) service, do the following:

1. 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* for your release.
2. 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. In My webMethods, do this on the Public Queue Information page (Administration > Integration > B2B > Public Queues, select the queue, and click **Edit**).

17 Reconciling Functional Acknowledgments

■ Overview	304
■ Turning FA Reconciliation On or Off	305
■ Information Module for EDI Records to Reconcile FAs	305

Overview

When you use Module for EDI with Trading Networks, Module for EDI can reconcile Functional Acknowledgments (FAs) to their related EDI documents.

Note:

Functional acknowledgments are not applicable to TRADACOMS or VDA.

To learn more about:

- What a functional acknowledgment (FA) is, see *webMethods Module for EDI Concepts Guide*.
- The Module for EDI `wm.b2b.edi.util.generateFA` service, which generates FAs, see [“Generating Acknowledgments” on page 153](#).
- How to generate an FA when using Trading Networks, see [“Automatically Generating Functional Acknowledgments” on page 246](#).

FA Reconciliation

To perform FA reconciliation, Module for EDI records each Group/Interchange EDI document that it sends and receives through Trading Networks. Module for EDI supports reconciliation of FA status for both batch and non-batch EDI documents. Whether it records a Group or an Interchange document depends on the EDI standard of the document.

- For ANSI X12, Module for EDI records each Group document that it sends or receives through Trading Networks.
- For UN/EDIFACT, Module for EDI records each Interchange-level document.

Module for EDI records information about these documents in the EDITRACKING table, which is a module-specific table in the Trading Networks database.

When Module for EDI receives the FA that corresponds to a Group/Interchange EDI document, the 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 Module for EDI 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.

Turning FA Reconciliation On or Off

FA reconciliation is enabled on a partner pair (sender/receiver) or group type basis.

- For partner pairs, you define whether you want to reconcile FAs using the *FARconciliation* EDITPA variable. Set this variable to On if you want Module for EDI to reconcile and auto-generate FAs based on partner pairs; set it to Off if you do not. If you want to turn FA reconciliation on or off for all partners, set the *FARconciliation* EDITPA variable in the default EDITPA and leave *FARconciliation* blank in all partner-specific EDITPAs.
- For group types, you also define whether you want to reconcile FAs using the *FARconciliation* EDITPA variable. In addition, you can define how the FA for different group types is reconciled using the *FAGeneration/DocumentSettings/autoGenerateFA* and *FAGeneration/DocumentSettings/groupType* EDITPA variables. The following table summarizes the use cases for FA reconciliation of group types.

When <i>FARconciliation</i> is set to...	And the <i>FAGeneration/DocumentSettings</i> variables are...	Module for EDI...
true	Not present	Reconciles the FA for all groups
true	Present	Reconciles the FA for each group according to the values in the <i>FAGeneration/DocumentSettings</i> variables
false	Not present	Does not reconcile FAs
false	Present	Does not reconcile FAs

For more information about EDITPA variables, see [“FARconciliation Variable” on page 68](#) and [“FAGeneration Variables” on page 75](#). For more information about default and partner-specific EDITPAs, see [“Defining EDI Trading Partner Agreements” on page 59](#).

Information Module for EDI Records to Reconcile FAs

When an ANSI Group or UN/EDIFACT Interchange document is sent or received through Trading Networks, Module for EDI records an entry for the document in the EDITRACKING table. Regardless of whether FA reconciliation is on or off, the module always records information in the EDITRACKING table for Group/Interchange EDI documents that are exchanged through Trading Networks. However, the value that the 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, Module for EDI attempts to locate an entry in the EDITRACKING table for the corresponding Group/Interchange EDI document. The 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 214](#).

Note:

For Module for EDI 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 Module for EDI records in the EDITRACKING table.

Type of document exchanged through Trading Networks	When <i>FARconciliation</i> is true, Module for EDI...	When <i>FARconciliation</i> is false, Module for EDI...
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 Module for EDI locates 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), Module for EDI updates the FA status in the EDITRACKING table to Accept. For a list of values for the FA status in the EDITRACKING table, see “FA Statuses” on page 306. ■ If Module for EDI does not 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 Module for EDI locates the Group/Interchange document: <ul style="list-style-type: none"> ■ If the current FA status for the Group/ Interchange document is Disable, the Module does nothing. ■ If the current FA status for the Group/ Interchange document is None, the Module updates the FA status to Disable. ■ If Module for EDI does not 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	Description
None	Module for EDI has not yet received or sent an FA to acknowledge this document.
Disable	The <i>FAReconciliation</i> EDITPA variable is set to false, which disables FA reconciliation and reporting. For more information about this variable, see “FAReconciliation Variable” on page 68 .
Duplicate	Module for EDI 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 307 .
Error	Module for EDI received an FA for which it could not locate a matching document in the EDITRACKING table. The module adds an entry to the EDITRACKING table for the FA and assigns it this status.
Duplicate FA/Errors	Module for EDI received more than one FA that matches this document.
Accept	Module for EDI 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	Module for EDI received a single FA that matches this document and the FA has an “E” (Errors) status on the confirmed level.
Partially Accept	Module for EDI received a single FA that matches this document and the FA has a “P” (Partially Accept) status on the confirmed level.
Reject	Module for EDI 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	Module for EDI encountered other unknown errors.
Interchange Received	Module for EDI received an interchange.

Controlling FA Status for Documents Submitted Multiple Times

Perform the following procedure to control how Module for EDI 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. In My webMethods: **Administration > Integration > B2B Settings > Configure Properties**.

2. In the Module for EDI Configuration Properties section, add or update the EDIResolveDuplicates property as follows:

Value	Description
false	<p>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 Module for EDI has not yet received an FA to acknowledge the document.</p> <p>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.</p> <p>Note: This behavior also applies to documents that you receive.</p>
true	<p>When you send a document the first time, its status is set to None ("100") in the EDITRACKING table. This indicates that Module for EDI 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 second 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> <p>Note: This behavior also applies to documents that you receive.</p>

3. Click **Save**. Module for EDI updates the configuration in memory so the changes take effect immediately and in the WmEDI/config/properties.cnf file.

Setting FA Status Manually

You can manually set the FA status of EDI documents that did not receive any acknowledgments if the following circumstances are met:

- The document is a group-level transaction, for example, X12 Group and UNEDIFACT Group document types.
- FA reconciliation is enabled (that is, the *FAReconciliation* EDITPA variable is set to true). For more information about this variable, see [“FAReconciliation Variable” on page 68](#).

- Edit EDI FA Status Attributes data permission is set to true.

➤ **To manually set the FA status**

- In My webMethods: **Monitoring > Integration > B2B > Transactions**.
- Search for the transaction of interest, and then click the Details icon.
- In the Transaction Details section of the Transactions page, click the **Attributes** tab.
- From the **EDI FA Status** list, select the desired FA status.
- Click **Save**.

18 Viewing Log Information for EDI Documents

■ Overview	312
■ Viewing the Server Log	312
■ Viewing the Trading Networks Activity Log	312

Overview

During processing, webMethods Module for EDI (Module for EDI) logs information to the server log. Additionally, if you are using Module for EDI with webMethods Trading Networks (Trading Networks), Module for EDI also logs information to the Trading Networks activity log.

Viewing the Server Log

Module for EDI logs informational, warning, and error messages to the server log of Integration Server. The messages that Module for EDI adds to the server log have one of the following keys:

Key	For messages issued from the...
EDICOR	WmEDI package of Module for EDI
EDIFTN	WmEDIforTN package of Module for EDI

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 Module for EDI and uses the key, EDICOR:

```
[EDICOR.000020.000404] No transaction set defined for SEF
```

To view the server log, use Integration Server Administrator. For more information, see the *webMethods Integration Server Administrator's Guide* for your release.

Viewing the Trading Networks Activity Log

Module for EDI writes information about EDI document processing activity to the Trading Networks activity log. Examples of activities that are written to this log include the following:

- **EDI document processing.** When an interchange, group, or transaction document is processed, the activity log entry with the activity class **Processing** states that the document was processed and identifies the sender, receiver, and agreement ID.
- **EDITPA used.** The activity log entry with the activity class **Processing** specifies the EDITPA Trading Networks used to process the EDI document.
- **TPA additions and changes.** If the Trading Networks configuration property *tn.tpa.EnableLogTPA* is set to true, the activity log entry with the activity class **TPA Usage and Settings** specifies when the TPA was created or revised and by whom, describes which parameters changed, and identifies the sender, receiver, and agreement ID. For more information about the *tn.tpa.EnableLogTPA* property, see the *webMethods Trading Networks Administrator's Guide* for your release.

Note:

Whenever you reset the value of *tn.tpa.EnableLogTPA*, you must reload the WmTN package to reflect the changes.

For more information on searching the activity log entries based on the activity class associated with the entries you want to view, see the *webMethods Trading Networks User's Guide* for your release.

You can view the activity log from My webMethods. For instructions, see the *webMethods Trading Networks User's Guide* for your release.

19 Viewing Information about EDI Documents

■ Overview	316
■ EDI Documents You Can View	316
■ Viewing EDI Documents	317
■ Viewing Information about EDI Documents in Queues	319
■ Viewing Related Documents with Relationship Labels	319

Overview

When you use webMethods Module for EDI (Module for EDI) 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 (transactions) from My webMethods.

To learn more about viewing documents, see the *webMethods Trading Networks User's Guide* for your release.

EDI Documents You Can View

By default, all Interchange, Group, and Transaction documents that Module for EDI 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 Module for EDI 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 Module for EDI 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.

Refer to the following table to learn where to go for more information.

For this EDI standard...	To learn more about...	See...
ANSI X12 and UN/EDIFACT	How Module for EDI uses the EDITPA variable <i>splitOption</i> to split documents into Interchange, Group, and Transaction documents	<i>webMethods Module for EDI Concepts Guide</i> and “splitOption Variable” on page 67
	EDITPAs and how to define them	“Defining EDI Trading Partner Agreements” on page 59
	The <i>persistMultipleDocEnvelope</i> EDITPA variable	“PersistMultipleDocEnvelope Variable” on page 65
TRADACOMS	How the Module for EDI uses the EDITPA variable <i>TRADACOMS/splitOption</i> to split documents into Transmission, Batch, and File documents	<i>webMethods Module for EDI Concepts Guide</i> and “TRADACOM Variables” on page 103

For this EDI standard...	To learn more about...	See...
	EDITPAs and how to define them	“Defining EDI Trading Partner Agreements When Using TRADACOMS” on page 101
	The <i>persistMultipleDocEnvelope</i> EDITPA variable for TRADACOMS documents	“TRADACOM Variables” on page 103

Viewing EDI Documents

You can view the following types of information about your EDI documents on the Transactions page in My webMethods:

- Sender and receiver of the EDI document
- Date the document was received
- Processing status of the EDI document
- User status
- Document type
- Related documents (for example, the functional acknowledgment [FA] that is associated with an ANSI X12 Group document)

By default, My webMethods displays the related documents identified by their control numbers. If a document does not contain a control number (for example, it is an invalid document), the document relationships cannot be displayed 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 319](#).

For more information about viewing documents, see the *webMethods Trading Networks User's Guide* for your release.

Trading Networks Processing Status and EDI Documents

The following table describes the processing statuses that Trading Networks sets while processing EDI documents. You can use these statuses as criteria for selecting the EDI documents that you want to view.

Processing Status	Description
NEW	Trading Networks has received the EDI document but it has not yet 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, “Batching EDI Documents” on page 275. For more information about delivering documents to VANs, see “Retrieving and Delivering EDI Documents from and to VANs” on page 297.</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.
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> for your release.

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. 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 page in My webMethods.

For more information about...	See...
Delivery tasks	The <i>webMethods Trading Networks User's Guide</i> for your release.
How to set up to batch EDI documents	"Batching EDI Documents" on page 275
How to set up to deliver EDI documents to VANs	"Retrieving and Delivering EDI Documents from and to VANs" on page 297

Viewing Related Documents with Relationship Labels

When you view related documents, 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 property to true, as described below.

Note:

All documents that were submitted to Trading Networks before you set this property to true will continue to be identified by their control numbers, not by relationship labels.

Note:

Relationship labels do not apply to the VDA standard.

➤ To set the relationship label property

1. In My webMethods: **Administration > Integration > B2B Settings > Configure Properties**.
2. In the Module for EDI Configuration Properties section, set the EDIUseNewRelationshipLabel property to true.
3. Click **Save**.

When you set this property, Module for EDI 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...

UN/EDIFACT and all supported sub-standards

If you view related documents for...

Envelope

- 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 one of the following:
 - **Envelope - DUP_FA**, where the FA (either sent or received) acknowledges a document that has already been acknowledged
 - **Envelope - Duplicate_Doc**, where the FA (either sent or received) can acknowledge more than one identical document that has not been acknowledged

Transaction

- Related envelopes are shown labeled **Envelope - Transaction**
- Related groups are shown labeled **Group - Transaction**

Group

- Related groups are shown labeled **Envelope - Group**
- Related transactions are shown labeled **Group - Transaction**

Functional Acknowledgment (FA)

- Related envelopes are shown labeled **Envelope - FA**
- If the FA is a duplicate FA, related envelopes are shown labeled **Envelope - DUP_FA** and **Envelope - Duplicate_Doc**

TRADACOMS

Transmission

- Related files are shown labeled **Transmission - File**
- Related batches are shown labeled **Transmission - Batch**

Batch

- Related files are shown labeled **Batch - File**

For...	If you view related documents for...
Multiple-envelope documents	<ul style="list-style-type: none"><li data-bbox="602 258 1386 348">■ Related batches are shown labeled Transmission - Batch File<li data-bbox="602 380 1386 443">■ Related transmissions are shown labeled Transmission - File<li data-bbox="602 474 1386 506">■ Related batches are shown labeled Batch - File <p data-bbox="602 531 1386 602">Multiple-envelope documents and their child documents are shown labeled Multiple Envelope - Envelope</p>

20 Including Documents in a Business Process

■ Overview	324
■ Designing the Process Model	324
■ Assigning Conversation IDs to EDI Documents	328
■ Sample Business Process	330
■ Monitoring Your Business Processes	330

Overview

Using Software AG Designer (Designer), 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 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 Module for EDI (Module for EDI) receives an EDI document, it splits the original inbound EDI document based on the EDITPA *splitOption* variable. Module for EDI 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.

Module for EDI always assigns the Interchange and Group documents conversation IDs. Consequently, the Interchange and Group documents are always passed to Process Engine, which is the facility of Integration Server that executes and manages business processes in Integration Server.

Module for EDI 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 (for example, X12 4010 850) using the WmEDIforTN home page.

Note:

You cannot process TRADACOMS documents in a business process.

For more information about processing EDI documents in business processes, including an illustration of how EDI documents are passed to a business process, see *webMethods Module for EDI Concepts Guide*.

For more information about using process models in Designer, see the *webMethods BPM Process Development Help* for your release.

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, see the *webMethods BPM Process Development Help* for your release.

Sample process models that involve EDI documents are provided in the Knowledge Center on the Empower Product Support website at <https://empower.softwareag.com>. You can import and view the sample models in Designer.

Business processes that involve EDI documents typically only involve the Transaction document and the level of the 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 always involves waiting 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.

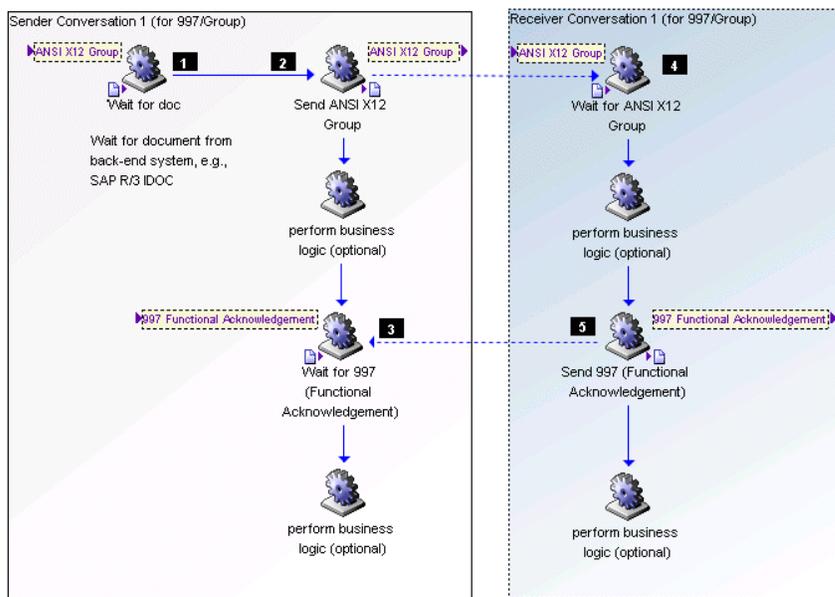
Sample Process Models for ANSI X12

The following diagrams illustrate how a sender and a receiver might set up criteria and actions for business processes involving ANSI X12 documents. See the tables after the diagrams 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.

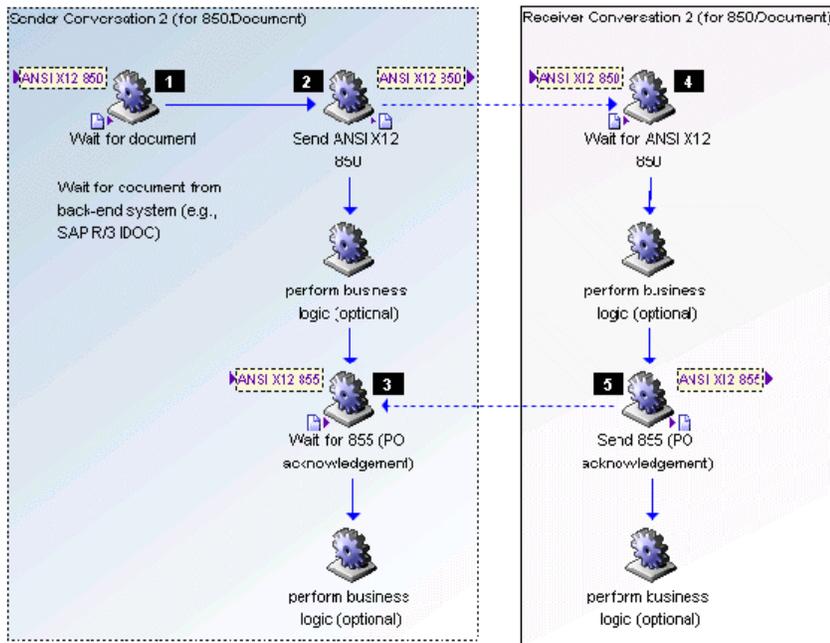
The following diagram illustrates the processing of an ANSI X12 Group document.



Step	Description
1	The business process starts when a document (for example, 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.

The following diagram illustrates the processing of an ANSI X12 850 Transaction document.



Step	Description
1	The business process starts when a document (for example, 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 (for example, 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 (for example, 855 PO Acknowledgment) back to the sender.

Following any send or wait step, partners can choose to perform optional business logic.

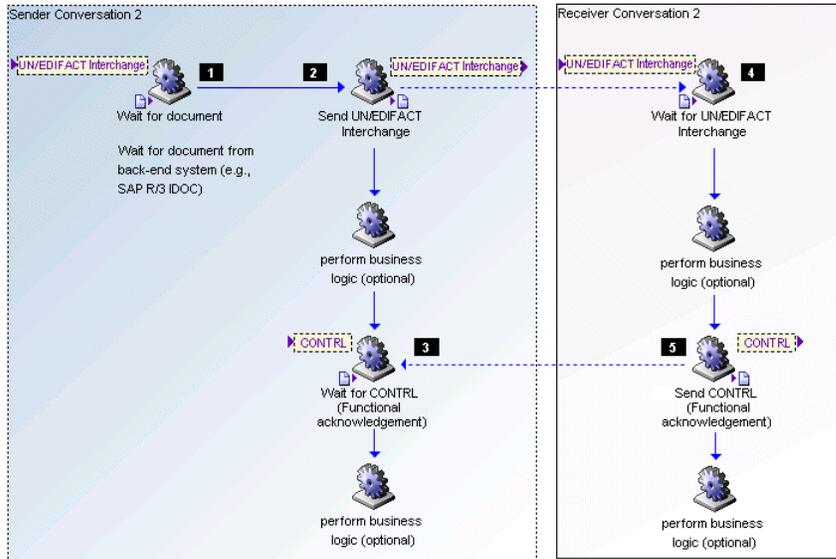
Sample Process Models for UN/EDIFACT

The following diagrams illustrate how a sender and a receiver might set up criteria and actions for business processes involving UN/EDIFACT documents. See the tables after each diagram 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.

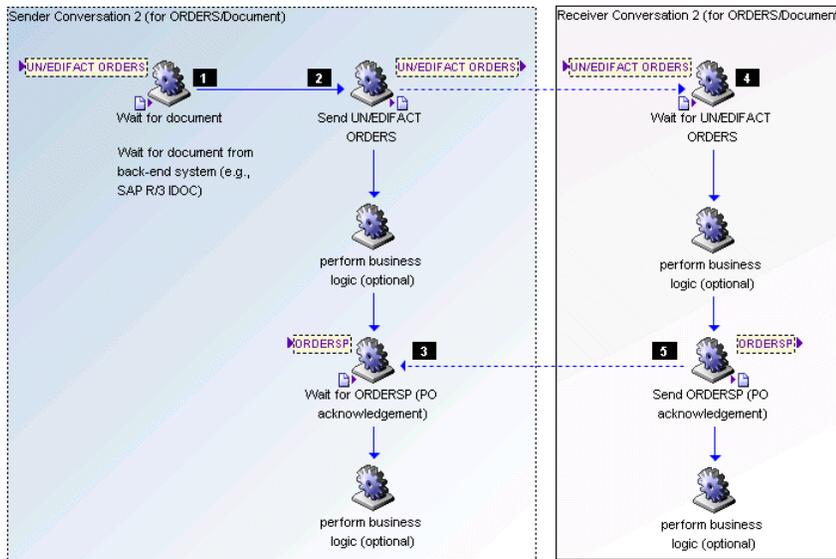
The following diagram illustrates the processing of the UN/EDIFACT Interchange document.



Step	Description
1	The business process starts when a document (for example, SAP R/3 IDOC) is received from the sender's back-end system.
2	The sender sends a UN/EDIFACT Interchange document.
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.

The following diagram illustrates processing for the UN/EDIFACT Transaction document.



Step	Description
1	The business process starts when a document (for example, SAP R/3 IDOC) is received from the sender's back-end system.
2	The sender sends a UN/EDIFACT ORDERS message.
3	The sender waits for the response message (for example, 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 (for example, ORDERSP receipt) back to the sender.

Following any send or wait step, partners can choose to perform optional business logic.

Assigning 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 Module for EDI assigns a conversation ID to the Interchange, Group, and Transaction documents. The general format of a conversation ID is either of the following:

- receiverID-senderID-instanceID
- senderID-receiverID-instanceID

When assigning the conversation ID, Module for EDI 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, Module for EDI 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 Module for EDI uses for the `instanceID` portion of the conversation ID.

Type of Document	Instance ID	Additional Information
Interchange	Control number from interchange envelope	Module for EDI always assigns a conversation ID to the Interchange document. The conversation ID has one of the following formats: <code>receiverID-senderID-controlNum</code> <code>senderID-receiverID-controlNum</code>
Group	Control number from group envelope	Module for EDI always assigns a conversation ID to the Group document. The conversation ID has the one of the following formats: <code>receiverID-senderID-controlNum</code> <code>senderID-receiverID-controlNum</code>
Transaction	Value of instance ID query	Module for EDI 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 Module for EDI 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. <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: This feature is deprecated in Module for EDI. Use the procedure to set a conversation ID as described in “Flat File Schema Namespace Conventions” on page 41 instead.</p> </div>

Instance IDs for Functional Acknowledgments

It is recommended 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 WmEDIsamples package, which you can download from the Technical community area of the Empower Product Support website at <https://empower.softwareag.com>, provides a sample that illustrates 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.

Note:

If you intend to also run the sample provided for Module for EDIINT (which is also included in the WmEDIsamples package), set up the Module for EDIINT sample before setting up this sample business process.

➤ **To learn about, set up, and run the sample**

1. Download the WmEDIsamples package from the Technical community area of the Empower Product Support website at <https://empower.softwareag.com>

Important:

Delete the WmEDIsamples package before going into production.

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

Use webMethods Monitor to monitor business processes that involve EDI documents. For more information, see *webMethods Monitor User's Guide*.

A Non-Standard Processing

■ Overview	332
■ Defining Partner Information When Using Non-Standard Processing	332
■ Defining Control Number Information for Partners When Using Non-Standard Processing	350
■ Processing Inbound EDI Documents using Trading Networks and Non-Standard Processing	352
■ Forming EDI Documents to Send Outbound When Using Trading Networks and Non-Standard Processing	353

Overview

Module for EDI 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**, Module for EDI 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. Module for EDI processes the groups and the transactions sets within each group using the settings you define for the group sender/receiver pairs.

For more information about standard versus non-standard processing, see *webMethods Module for EDI Concepts Guide*.

This appendix contains only information for when non-standard processing differs from standard processing.

Defining Partner Information When Using Non-Standard Processing

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 [“Defining Trading Partner Information” on page 53](#).

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	■ Interchange sender receiver pair information that you set using the Module for EDI home page. For more information, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 335 .
Group sender/receiver pairs	■ Trading Networks profiles for each group sender and receiver. For more information, see “Defining Trading Networks Profiles When Using Non-Standard Processing” on page 333 .
Relationship of group sender/receiver pairs to the interchange sender/receiver	■ All the group sender/receiver pairs that are associated with the interchange sender/receiver pair. You define this information using the Module for EDI home page. For more information, see “Defining Group-Level Sender/Receiver Pair Associations” on page 346 .
Tailoring how Module for EDI processes documents	■ Default EDITPA that defines the settings that you want to use for most partner pairs.

For...	Define...
	<ul style="list-style-type: none"> 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 59. For information related specifically to non-standard processing, see “Defining a Partner-Specific EDITPA When Using Non-Standard Processing” on page 333.</p>
How Module for EDI 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 When Using Non-Standard Processing” on page 350. Actions Module for EDI is to take when it encounters invalid control numbers. For more information, see “Defining Actions for Invalid Control Numbers” on page 351. Control number validation settings (control number maximum, minimum, increment, and window). For more information, see “Configuring Control Number Settings” on page 116.

Defining Trading Networks Profiles When Using Non-Standard Processing

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 Partner Profiles”](#) on page 55.

Defining a Partner-Specific EDITPA When Using Non-Standard Processing

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 59. For more information about defining partner-specific EDITPAs, see [“Defining a Partner-Specific EDITPA”](#) on page 60.

Note:

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

wm.b2b.editn.TPA:EDITPA IS Document Type When Using Non-Standard Processing

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 62](#).

***splitOption* EDITTPA Variable Default: Transaction**

The *splitOption* variable indicates how you want the Module for EDI to split an interchange segment within an EDI document. You can specify Interchange, Group, or Transaction.

When you are using non-standard processing and you specify Interchange, Module for EDI will split the document at the Group level.

For more information, see [“splitOption EDITPA Variable” in “splitOption Variable” on page 67](#).

***ControlNumberManagement/validateInboundEnvelopeControlNumbers* EDITPA Variable Default: false**

When you are using non-standard processing, Module for EDI 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 Module for EDI home page.

For more information about accessing and using this screen, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#).

***ControlNumberManagement/duplicateControlNumberAction* EDITPA Variable Default: Error & Continue**

The *ControlNumberManagement/duplicateControlNumberAction* variable indicates the action you want Module for EDI 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 Module for EDI home page. For more information about accessing and using this screen, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#).

You can specify one of the following for *ControlNumberManagement/duplicateControlNumberAction*:

- Error & Continue
- ProcessNormally
- Reject

For more information, see [“ControlNumberManagement/duplicateControlNumberAction EDITPA Variable” in “ControlNumberManagement Variables” on page 83](#).

***ControlNumberManagement/outOfSequenceControlNumberAction* EDITPA Variable Default: Error & Continue**

The *ControlNumberManagement/outOfSequenceControlNumberAction* variable indicates the action you want Module for EDI 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 Module for EDI home page. For more information about accessing and using this screen, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#).

You can specify one of the following for *ControlNumberManagement/outOfSequenceControlNumberAction*:

- Error & Continue
- ProcessNormally
- Reject

For more information, see [“useReverseRoute EDITPA Variable” in “useReverseRoute Variable” on page 86](#).

FAGeneration EDITPA Variables

When you are using non-standard processing, Module for EDI does not use any of the *FAGeneration* EDITPA variables. Instead, Module for EDI uses the corresponding values that you specify on the Interchange Information Detail screen of the Module for EDI home page. For more information about accessing and using this screen, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#).

Defining Interchange-Level Sender/Receiver Pair Information

When you are using non-standard processing, Module for EDI 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 Module for EDI 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 Module for EDI 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.

Module for EDI stores the interchange-level sender/receiver pair information that you define in the EDIEnvelope table, which is a Module for EDI-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 Module for EDI 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 345.

➤ **To define information for an interchange-level sender/receiver pair**

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

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 Module for EDI uses when processing an inbound EDI document:

For this Inbound Information field...	Specify...
Create Doc	Whether you want Module for EDI 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 Module for EDI 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 189.

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 When Using Non-Standard Processing](#)” on page 334.

For this Inbound Control Number Validation field..	Specify...								
Validate inbound envelope control numbers	Whether you want Module for EDI 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 Module for EDI to take when it encounters a duplicate control number in an interchange header.								
	<table border="1"> <thead> <tr> <th data-bbox="492 590 784 636">Select...</th> <th data-bbox="784 590 1377 636">To have Module for EDI</th> </tr> </thead> <tbody> <tr> <td data-bbox="492 646 784 762">Error & Continue</td> <td data-bbox="784 646 1377 762">Module for EDI logs the error; then continues to process the EDI document that contains the invalid control number normally.</td> </tr> <tr> <td data-bbox="492 772 784 888">Process Normally</td> <td data-bbox="784 772 1377 888">Module for EDI logs a warning; then continues to process the EDI document that contains the invalid control number normally.</td> </tr> <tr> <td data-bbox="492 898 784 1869">Reject</td> <td data-bbox="784 898 1377 1869"> <p>Module for EDI logs the error and does not process the document normally. Module for EDI does not split the EDI document. Typically, Module for EDI 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, Module for EDI sends the document without splitting it to Trading Networks processing rules.</p> <p>Additionally, Module for EDI sets 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</p> </td> </tr> </tbody> </table>	Select...	To have Module for EDI	Error & Continue	Module for EDI logs the error; then continues to process the EDI document that contains the invalid control number normally.	Process Normally	Module for EDI logs a warning; then continues to process the EDI document that contains the invalid control number normally.	Reject	<p>Module for EDI logs the error and does not process the document normally. Module for EDI does not split the EDI document. Typically, Module for EDI 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, Module for EDI sends the document without splitting it to Trading Networks processing rules.</p> <p>Additionally, Module for EDI sets 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</p>
Select...	To have Module for EDI								
Error & Continue	Module for EDI logs the error; then continues to process the EDI document that contains the invalid control number normally.								
Process Normally	Module for EDI logs a warning; then continues to process the EDI document that contains the invalid control number normally.								
Reject	<p>Module for EDI logs the error and does not process the document normally. Module for EDI does not split the EDI document. Typically, Module for EDI 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, Module for EDI sends the document without splitting it to Trading Networks processing rules.</p> <p>Additionally, Module for EDI sets 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</p>								

For this Inbound Control Number Validation field...

Specify...

document. For information, see [“Defining Processing Rules to Handle Documents with Invalid Control Numbers”](#) on page 236.

Out of sequence control number action

The action you want Module for EDI 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.

You can later force processing of the out-of-sequence document if you want. For more information, see [“Reprocessing Documents with Out-of-Sequence Control Numbers”](#) on page 240.

9. In the **Inbound Information - FA Generation** section of the screen, specify the following information that Module for EDI uses to determine whether it should automatically generate functional acknowledgments (FAs) for inbound EDI document and settings for generating the FAs. For more information about automatic FA generation, see [“Automatically Generating Functional Acknowledgments”](#) on page 246.
 - a. In the **Auto Generate FA** field, specify whether you want to turn automatic FA generation on as follows:

Specify...	To have Module for EDI...
On	Always automatically generate FAs.
Per Document	Automatically generate FAs based on the indicator flag in the interchange header (ISA14 ore UNB09).
Off	Never automatically generate FAs.

- b. In the **FA Level** field, specify the level of detail that you want Module for EDI to acknowledge in the FAs that it generates as follows:

Specify...	To have Module for EDI acknowledge at the...
Default	Envelope level (group for ANSI X12 and interchange for UN/EDIFACT)
TransactionSet	Transaction set level
Segment	Segment level
Element	Element level

Specify... To have Module for EDI acknowledge at the...**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 156.

- c. In the **Process Document** field, specify how you want Module for EDI 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.

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

Specify... If you want...

- | | |
|---------------|---|
| 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 [“Defining Actions Module for EDI Takes Based on FA Status”](#) on page 249.

- d. In the **Generate Control Number** field, specify how Module for EDI is to generate the control numbers that are used in the interchange and group headers of the FAs that are automatically generated. Select one of the following:

Specify...	To have Module for EDI...
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.

- e. In the **Syntax Error Status** field, specify how you want Module for EDI to report the syntax error status for a transaction, group, or UN/EDIFACT interchange. The Module for EDI 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.

Specify...	To have Module for EDI...
Rejected	Report the syntax error status as Rejected if syntax errors are encountered. Select this option 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. Select this option if you want to know whether there are syntax errors, but do not want to reject an element because of them.
Accepted	Report the syntax error status as Accepted regardless of any syntax errors that might be encountered. Select this option if you do not want to check for syntax errors.

- f. In the **Logical Error Status** field, specify how you want Module for EDI to report the logical error status for a transaction, group, or UN/EDIFACT interchange.

Module for EDI uses the logical error status along with the syntax error status and child transaction rejected status (if applicable) to determine the FA status for a transaction, group, or UN/EDIFACT interchange.

Set the status as follows:

Specify...	To have Module for EDI report the logical errors as...
Rejected	Rejected if they are encountered. Specify this option if you want to reject elements that have logical errors.
Accepted, But Errors Were Noted	Accepted, But Errors Were Noted if they are encountered. Specify this option if you want to know whether there are logical errors, but do not want to reject an element because of them.

Specify... To have Module for EDI report the logical errors as...

Accepted Accepted regardless of any logical errors that might be encountered. Specify this option if you do not want to check for logical errors.

- g. In the **Child Transaction Rejected Status** field, specify how you want Module for EDI 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".

Specify one of the following:

Specify... To have Module for EDI...

Rejected Report the child transaction rejected status as:

- 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 Report the child transaction rejected status as:

- 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, Accepted, But Errors Were Noted, or both.
- Accepted, But Errors Were Noted if the FA statuses of the child transactions are Rejected, Accepted, But Errors Were Noted, or both, *and* no child transactions are Accepted.
- Accepted if the FA statuses of all the child transactions are Accepted.

Accepted, But Errors Were Noted Report the child transaction rejected status as:

- 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.

10. In the **Outbound Information - FA Generation** section of the screen, specify the following information that Module for EDI 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	Whether to add a group to the FA.						
	<table border="1"> <thead> <tr> <th>Select...</th> <th>To have Module for EDI...</th> </tr> </thead> <tbody> <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> </tbody> </table>	Select...	To have Module for EDI...	Yes	Add a group to the FA.	No	Not add a group to the FA. This is the default.
Select...	To have Module for EDI...						
Yes	Add a group to the FA.						
No	Not add a group to the FA. This is the default.						
Control Number With Leading Zero	Whether to pad control numbers in the FA with leading zeros.						
	<table border="1"> <tbody> <tr> <td>Yes</td> <td>Pad control numbers in the FA with leading zeros.</td> </tr> <tr> <td>No</td> <td>Do not pad control numbers with leading zeros.</td> </tr> </tbody> </table>	Yes	Pad control numbers in the FA with leading zeros.	No	Do not pad control numbers 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:

For this Outbound Information - Delimiters field...	Specify...
Segment	The segment terminator for the outbound EDI document (for example, "+"). The default is the new line character.
Field	The field separator for each EDI segment (for example, !). 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 (for example, "?").

For this Outbound Information - Delimiters field...	Specify...
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:

For this Outbound Information - Envelope Information field...	Specify...
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.

For this Outbound Information - Envelope Information field...	Specify...
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.

- Click **Save Changes**. Module for EDI 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 Module for EDI display its information. Then you can edit, delete, or copy the information.

➤ To edit existing Interchange-level sender/receiver pair information

- Open Integration Server Administrator if it is not already open.
- In the **Solutions** menu of the navigation panel, click **EDI**. Integration Server Administrator opens a new browser window to display the Module for EDI home page.
- From the Module for EDI home page, in the **Partner Set Up** menu of the navigation panel, click **Manage Interchange Info**.
- 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.

For this field...	Specify...
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 Module for EDI 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 Module for EDI to return.

- Click **Search**. Module for EDI displays the list of interchange-level sender/receiver pairs that match your criteria at the bottom of the screen.
- 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. Module for EDI 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. Module for EDI displays the information for the new interchange sender/receiver pair. You can edit fields as necessary.
- 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		
D	E	X	Z
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 requires the interchange-level sender/receiver pair. To make a correlation between group-level sender/receiver to interchange-level sender/receiver, your service 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, the service 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.

Module for EDI saves the group-level sender/pair associations in the EDIGroup table, which is a Module for EDI-specific table in the Trading Networks database. The EDIGroup table works together with the EDIEnvelope table, which contains the interchange-level sender/receiver pair information that is described in [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#). The following graphic illustrates how the EDIEnvelope and EDIGroup tables work together.

EDIGroup Table (group-level sender/receiver pair associations)				EDIEnvelope Table (interchange-level sender/receiver pair information)		
Group sender	Group receiver	Interchange sender	Interchange receiver	Interchange sender	Interchange receiver	
A	B	X	Y	X	Y	outbound interchange header information
C	B					
D	B					
A	C					
D	E	X	Z	X	Z	outbound interchange header information
D	F					

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 335.

➤ **To define Group-level sender/receiver pair associations**

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

5. Click **Search**. Module for EDI 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. Module for EDI 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**. Module for EDI displays the Add GS Pairs screen.
 - b. Specify the following about the group-level sender/receiver pair:

For this field...	Specify...
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 for existing group-level sender/receiver pair associations

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

For this field...	Specify
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.

For this field...	Specify
Sort By	How you want Module for EDI to sort the returned list of group sender/receiver pairs.
Maximum Results	The maximum number of matching group-level sender/receiver pairs you want Module for EDI to return

- Click **Search**. Module for EDI 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 When Using Non-Standard Processing

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 [“Defining Control Number Information for Trading Partners”](#) on page 113.

Turning Inbound Control Number Validation On or Off When Using Non-Standard Processing

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 Module for EDI home page. For more information about how to define this setting, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 335.
Group	Use the EDITPA variable <code>ControlNumberManagement/validateInboundGroupControlNumbers</code> . To turn group control number validation: <ul style="list-style-type: none"> ■ On, set to true ■ Off, set to false <p>For more information, see “ControlNumberManagement/validateInboundGroupControlNumbers EDITPA Variable” in “ControlNumberManagement Variables” on page 83.</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 action, see [“Defining Actions Module for EDI Can Take for Invalid Control Numbers”](#) on page 124.

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 Module for EDI home page. For instructions for defining this setting, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 335.
	Out-of-sequence	Use the Out of sequence control number action setting on the Module for EDI home page. For instructions for defining this setting, see “Defining Interchange-Level Sender/Receiver Pair Information” on page 335.
Group	Duplicate	Use the EDITPA variable <i>ControlNumberManagement/duplicateControlNumberAction</i> . For a list of the settings for this EDITPA variable, see “ControlNumberManagement/duplicateControlNumberAction EDITPA Variable” in “ControlNumberManagement Variables” on page 83.
	Out-of-sequence	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 Module for EDI performs for each, see “useReverseRoute EDITPA Variable” in “useReverseRoute Variable” on page 86.

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 the Control Number Settings”](#) on page 351.

You define control number settings for a unique combination of:

- sender/receiver pair

- production mode (for example, Testing for Production)
- EDI standard and version
- type, that is, "Envelope" or group type (for example, 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, for example, 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, for example, 01 to identify a D-U-N-S number.

Processing Inbound EDI Documents using Trading Networks and Non-Standard Processing

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 [“Optional Inbound Processing When Using Trading Networks”](#) on page 229.

Before You Can Process Inbound EDI Documents

In addition to the items listed in [“Preparing to Receive EDI Documents”](#) on page 187 in [“Preparing to Process Inbound Non-TRADACOMS Documents Using Trading Networks”](#) on page 185, 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 335.

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 Module for EDI home page. For more information about these settings including how to define them, see [“Defining Interchange-Level Sender/Receiver Pair Information”](#) on page 335.

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 188](#).

Interchange Sender/Receiver Pair Inbound Information Setting	Description
Create Doc	Whether you want the Module for EDI 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 Module for EDI 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 Module for EDI 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 Module for EDI 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 Module for EDI 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 and Non-Standard Processing

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 [“Forming EDI Documents to Send Outbound When Using Trading Networks ” on page 257](#).

Before Creating the Service to Form an EDI Document

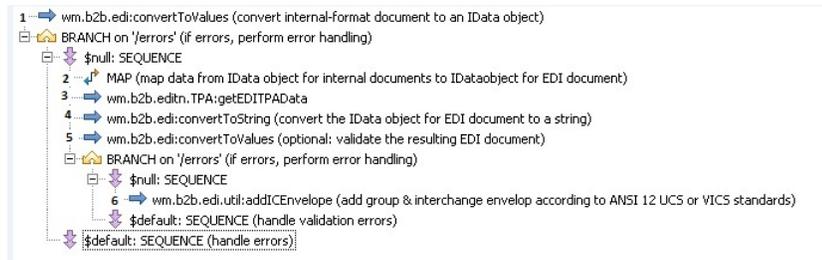
In addition to the items listed in [“Before Creating the Service to Form an EDI Document” on page 259](#), when using non-standard processing you must also define:

- Interchange sender/receiver pair information. For instructions on defining this information, see [“Defining Interchange-Level Sender/Receiver Pair Information” on page 335](#).

- Group-level sender/receiver pair associations. For instructions on defining this information, see [“Defining Group-Level Sender/Receiver Pair Associations”](#) on page 346.

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 Module for EDI and when using non-standard processing.



B Using Module for EDI Decoupled from Trading Networks

■ Overview	356
■ Configuring Module for EDI Decoupled from Trading Networks	356
■ Preparing Module for EDI Decoupled from Trading Networks	358
■ Creating Flat File Schemas for EDI Documents	358
■ Forming EDI Documents and Sending Them Outbound	361
■ Example	363

Overview

webMethods Module for EDI offers built-in services in WmEDI and WmEDIforTN packages. The WmEDI package contains the core EDI document processing built-in services. The WmEDIforTN package contains additional built-in services for processing EDI documents using webMethods Trading Networks (Trading Networks).

In a distributed environment, to handle heavy EDI document processing loads, you can configure some of your servers to do the job of just translating the EDI documents by decoupling Module for EDI from Trading Networks. You can have a centralized Trading Networks server for storing the transactions in your EDI network and routing the EDI documents to your partners.

This appendix describes how to configure your EDI network nodes to use only the basic EDI document parsing and validating services available in the WmEDI package.

Note:

When you use Module for EDI decoupled from Trading Networks, you must install Module for EDI on each Integration Server in the cluster, and each installation must be identical. For information on running Integration Server in a clustered environment, see the *webMethods Integration Server Clustering Guide* for your release.

Using Only the WmEDI Package with Integration Server

The basic EDI engine of Module for EDI (that is, the WmEDI package) functions as a toolkit that you can use to build your own EDI solution. Module for EDI provides the following functionality when you use only the WmEDI package with webMethods Integration Server:

- Process most EDI standards, such as ANSI X12, VICS, UCS, UN/EDIFACT, ODETTE, EANCOM, and VDA

Note:

The WmEDI package does not support TRADACOMS document processing.

- Parse, convert, format, and validate EDI documents
- Process EDI documents containing multiple interchanges/groups/transactions with multiple versions
- Generate functional acknowledgments (FAs), if they are applicable to your standard
- Create envelopes for EDI documents
- Transport EDI documents using the FTP, HTTP, and HTTPS protocols

Configuring Module for EDI Decoupled from Trading Networks

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, Module for EDI uses flat file schemas. You create flat file schemas that define the structure of EDI documents. Also, you can 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.

The following high-level steps must be taken to create your own EDI solution.

Step	Description	Where to Find More Information
1	Set up Module for EDI decoupled from Trading Networks.	“Preparing Module for EDI Decoupled from Trading Networks” on page 358
2	Create flat file schemas for EDI documents.	“Creating Flat File Schemas for EDI Documents” on page 358
3	If you want to use EDI built-in services to convert internal-format documents (for example, 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.	“Creating Flat File Schemas for Internal-Format Documents” on page 34
4	Configure how you want format services to convert field values in documents from EDI format to internal format, and vice versa.	“Configuring Format Services to Convert Field Values” on page 30
5	Specify how to associate format services to fields defined in a flat file schema for an EDI document.	“Associating the EDI Format Services with EDI Data Types” on page 32
6	Create clients to send the EDI documents to Integration Server.	“Creating Clients that Send EDI Documents to Integration Server” on page 131
7	Create the services to process your EDI documents. The WmEDI package of Module for EDI 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.	“Creating the Service the Client Invokes to Handle EDI Documents” on page 133
8	Use the <code>wm.b2b.edi.util:generateFA</code> service to generate and send functional acknowledgments (FAs) for EDI documents.	“Generating Acknowledgments” on page 153

Step	Description	Where to Find More Information
9	Form EDI documents to be sent outbound.	“Forming EDI Documents and Sending Them Outbound” on page 361

Preparing Module for EDI Decoupled from Trading Networks

To set up Module for EDI decoupled from Trading Networks, you must perform these high-level steps:

Step	Description	Where to Find More Information
1	Install Module for EDI.	“Installing Module for EDI ” on page 12
2	Disable WmEDIforTN and WmTN packages. By disabling the WmEDIforTN and WmTN packages, Module for EDI does not route the EDI documents to Trading Networks for additional EDI document processing. You can now speed up EDI document processing by using only the core built-in services in the WmEDI package.	The <i>webMethods Integration Server Administrator’s Guide</i> for your release.
Note: The user interface in My webMethods is not available when you disable the WmTN package. Use the Module for EDI home page in Integration Server Administrator to configure Module for EDI.		
3	Define the Module for EDI properties in the properties.cnf configuration file. For handling large documents, set an appropriate value for the EDIBigDocThreshold property.	“Defining Module for EDI Properties” on page 24

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 both of the following are true:

- You are not using the Module for EDI in conjunction with Trading Networks

- 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:

- You are using Module for EDI in conjunction with Trading Networks. The flat file schemas are created automatically for you when you use the procedure described in [“Defining TN EDI Document Types” on page 34](#).

Important:

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

- You are processing non-standard EDI formats or a user-defined format (such as a non-EDI flat file). You must create flat file schemas from Software AG Designer using the functionality in the WmFlatFile package instead of using one of the methods described in [“Creating Flat File Schemas for EDI Documents” on page 358](#). For more information, see *Flat File Schema Developer’s Guide*.

Creating a Flat File Schema from a SEF File

Module for EDI 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 with EDI Data Types” on page 32](#).

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

Note:

If you are using Module for EDI in conjunction with 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 [“Defining TN EDI Document Types” on page 34](#).

To create a flat file schema from a SEF file, you must perform these high-level steps:

Step	Description
------	-------------

- | | |
|---|--|
| 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 Module for EDI. When you install Module for EDI, the installer stores each SEF file as:

```
wmEDI/pub/SEFs/standard/version.sef
```

Step	Description
------	-------------

where

standard represents the EDI standard (for example, X12) and *version* represents the EDI standard version (for example, 4010).

For example:

```
WmEDI/pub/SEFs/X12/4010.sef
```

Note:

The installer stores each SEF file as: `WmEDI/pub/SEFs/Standard_SefDesc/version.sef`. When **InstallSEFSchemaWithDescription** is set to `true`.

- 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 Designer. 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 *targetSchema* input parameter 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), and *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) and *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.

Step	Description
	This temporary flat file schema will be deleted upon successful execution of the <code>wm.b2b.edi.tradacoms.ui:modifyTradacomsSchema</code> service.
4	For descriptions of all the input parameters of these services, see <i>webMethods Module for EDI Built-In Services Reference</i> .
5	From Designer, 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 associated 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

Module for EDI 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 more information about this service, see *webMethods Module for EDI Built-In Services Reference*.

To create a flat file schema from an SAP IDOC, you must perform the following high-level steps.

Step	Description
1	Invoke the <code>wm.b2b.edi:createIDOCtemplate</code> service (e.g, from Designer). 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 parameters, see the descriptions of the <code>wm.b2b.edi:createIDOCtemplate</code> service in <i>webMethods Module for EDI Built-In Services Reference</i> .
2	From Designer, 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.

Forming EDI Documents and Sending Them Outbound

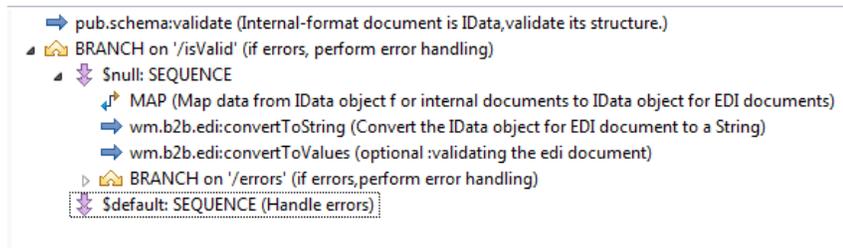
You can send documents to Integration Server from internal applications (for example, back-end systems) and map information from them into a standard EDI format document. The basic EDI engine of Module for EDI (that is, the WmEDI package) provides tools that you need to create the EDI document.

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

For more information about forming EDI documents to send outbound, see *webMethods Module for EDI Concepts Guide*.

Validating the Input Internal-Format Document When It Is an IData Object

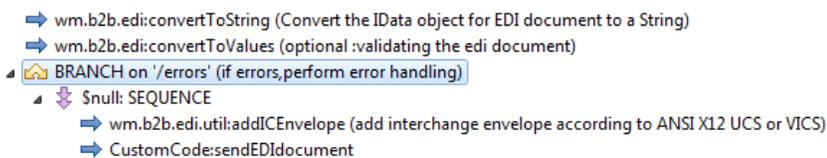
The logic in the scenario described above assumes the internal-format document is passed to your service as a String or InputStream. In this situation, you can have the `wm.b2b.edi:convertToValues` service validate the document. If your client passes the internal-format document to your service as an 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 in the following diagram.



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* for your release.

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:addICEEnvelopeEDIFACT` service rather than invoke the `wm.b2b.edi.util:addGroupEnvelope` and `wm.b2b.edi.util:addICEEnvelope` services, as shown in the following diagram.



For more information about the `wm.b2b.edi.util:addICEEnvelopeEDIFACT` service, see *webMethods Module for EDI 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 EDI document envelope as input to the `wm.b2b.edi:envelopeProcess` service or the `wm.b2b.edi.util:generateFA` service. For more information about these services, see *webMethods Module for EDI 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 Center Samples area on the Empower Product Support website at <https://empower.softwareag.com>. The samples in this folder have been certified, meaning that they have been tested by Software AG.

Important:

Delete the WmEDIsamples package before going into production.

