



webMethods EDI Module

Concepts Guide

VERSION 6.5.2

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

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

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

All other marks are the property of their respective owners.

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

Contents

About This Guide	5
Document Conventions	5
Additional Information	5
Chapter 1. Overview of webMethods EDI Module	7
webMethods EDI Module Introduction	8
Architecture	10
Chapter 2. Using the EDI Module without Other webMethods Components	13
Overview	14
Processing of Inbound EDI Documents	14
EDI Client	15
Service to Process the Inbound EDI Document	16
Flat File Schemas	18
Flat File Schemas that Define EDI Documents	18
Flat File Schemas that Define Internal-Format Documents	19
Functional Acknowledgments	19
Forming EDI Documents to Send Outbound	21
Chapter 3. Using the EDI Module with Trading Networks	23
Overview	24
Processing Inbound EDI Documents with Trading Networks	25
EDI Client	25
EDI Documents Inbound Processing	26
Initial Processing of the EDI Recognizer	28
Processing Documents in Trading Networks	33
Processing Rules for Inbound EDI Documents	36
Pre-Processing Actions	36
Processing Actions	37
EDI Trading Partner Agreements (EDITPAs)	38
Forming EDI Documents and Sending Them Outbound	39
Delivering the EDI Document Directly from the Service that Forms It	40
Submitting the Outbound EDI Document to Trading Networks to Deliver It	42
Routing the Outbound EDI Document to Trading Networks to Deliver It	44
Working with VANs	46
Inbound Processing: Retrieving Documents from VANs	47
Outbound Processing: Delivering Documents to VANs	48

- Batching Outbound EDI Documents 51
 - Creating the Batched EDI Documents 53
 - Delivering the Batched Document 54
- Chapter 4. EDI Documents in Business Processes 57**
 - Overview 58
 - How You Define the Business Process 58
 - Conversation IDs for EDI Documents 59
 - How EDI Documents are Passed to a Business Process 60
- Appendix A. Glossary 63**
- Index..... 71**

About This Guide

This guide is for users of the webMethods EDI Module. It provides an overview of the EDI Module and its features. It describes the functionality of the EDI Module when you use it without other webMethods components and the additional functionality you gain when you use it with Trading Networks and other webMethods components.

Document Conventions

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

Additional Information

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

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

Overview of webMethods EDI Module

■ webMethods EDI Module Introduction	8
■ Architecture	10

webMethods EDI Module Introduction

The webMethods EDI Module (EDI Module) enables business partners to exchange EDI documents within and across the extended enterprise, providing comprehensive EDI support as a key part of the webMethods total business process automation solution.

The EDI Module provides services and transformation management tools that enable you to recognize, transform, convert, validate, and map multi-transactional EDI documents in real time or in batch.

When you use the EDI Module along with other webMethods components, you can extend its capabilities. The following table shows the capabilities of the EDI Module based on the components you use:

Using the webMethods EDI Module with:	Provides the functionality to...
Only the functionality available in the Integration Server	<ul style="list-style-type: none">■ Process most EDI standards, such as ANSI X12, VICS, UCS, UN/EDIFACT, ODETTE, and EANCOM.
	<p>Note: Support for the TRADACOMS standard is provided when you use the EDI Module in conjunction with webMethods Trading Networks.</p> <ul style="list-style-type: none">■ Parse, convert, format, and validate EDI documents■ Process EDI documents containing multiple interchanges/groups/transactions with multiple versions■ Generate functional acknowledgements (FAs), if they are applicable to your standard■ Create envelopes for EDI documents■ Transport EDI documents using the FTP, HTTP, and HTTPS protocols

Using the webMethods EDI Module with:	Provides the functionality to...
webMethods Trading Networks	<ul style="list-style-type: none"> ■ Maintain information about your trading partners that exchange EDI documents ■ Send EDI documents to and retrieve EDI documents from trading partners. ■ Connect to VANs (Value Added Networks) to pick up and deliver EDI documents ■ Process EDI documents using the features of Trading Networks (e.g., using processing rules) ■ Batch the sending of EDI documents rather than sending them to their destinations in real time, as each is received ■ Automatically generate functional acknowledgments (FA) for inbound documents ■ Reconcile EDI documents with their corresponding functional acknowledgements (FAs) to create FA reconciliation reports ■ View EDI documents that have passed through your system
webMethods Trading Networks	■ Leverage webMethods business process modeling and management capabilities for EDI documents
webMethods Modeler	You can use webMethods Modeler to create EDI-based business process models. Trading Networks passes documents to process management to perform the actions you specify in your process models. You then can monitor the progress of your business process using webMethods Monitor.
webMethods Monitor	

Architecture

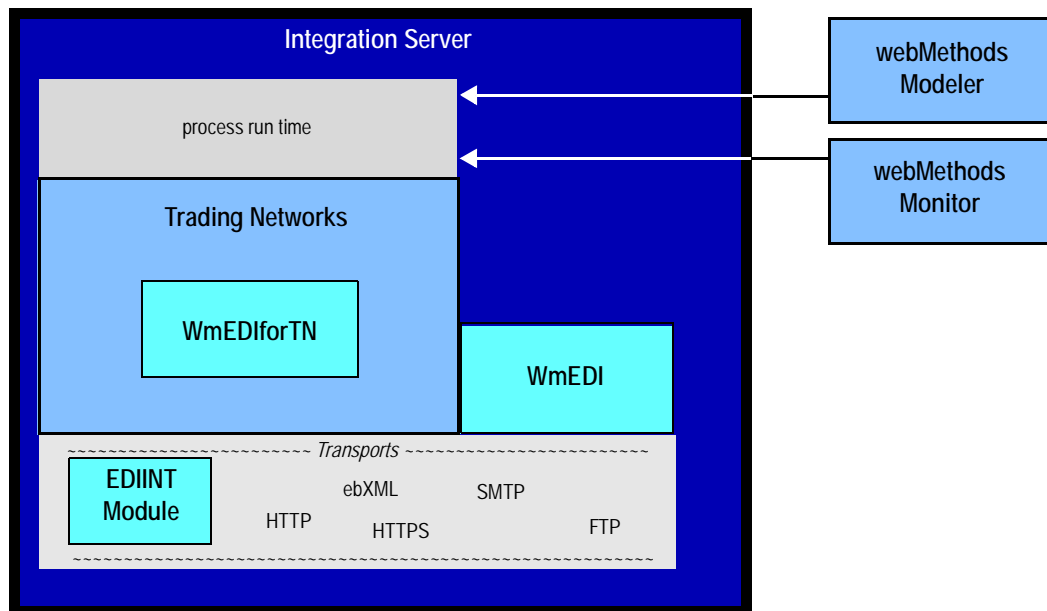
When you install the EDI Module, two packages are installed into the Integration Server: the WmEDI package and the WmEDIforTN package.



Note: A package named WmEDIsamples contains sample EDI flow services, mappings, and IS document types that demonstrate how to use the EDI Module and webMethods Developer to execute typical EDI processing scenarios. This package is located in the Knowledge Base Samples area on the Advantage Web site at <http://advantage.webmethods.com>. Its samples have been certified, meaning that they have been tested by webMethods. EDI developers can use the WmEDIsamples as a reference. Before going into production, you should delete the WmEDIsamples package.

The following diagram illustrates how the EDI Module fits into the webMethods architecture. For more information, see the text after the diagram.

EDI Module and the webMethods Architecture



- **Integration Server** is the underlying foundation of the webMethods components.
- **Trading Networks** is a webMethods component that enables your enterprise to link with other companies (buyers, suppliers, strategic partners) and marketplaces to form a business-to-business trading network. For more information, see the *webMethods Trading Networks Concepts Guide* and the *webMethods Trading Networks User's Guide*.

- **EDI Module** in a production environment is comprised of the following two packages:
 - The **WmEDI package** is the basic functionality that provides support for the EDI standard to the webMethods components.
 - The **WmEDIforTN package** allows for the interaction between the WmEDI package and Trading Networks. This interaction allows you to use Trading Networks as a gateway for EDI document exchange. The EDI Module uses the functionality of Trading Networks to provide additional features, such as support for VANs, reconciling FAs, and batching the sending of EDI documents.

- **webMethods EDIINT Module** adds support for the EDIINT exchange protocol, which is a standard for how to transport documents. Documents using the EDIINT protocol are processed through Trading Networks.

The EDIINT Module supports EDIINT AS1 (SMTP (e-mail)), EDIINT AS2 (HTTP or HTTP/S), and EDIINT AS3 (FTPS (FTP over SSL)) messages, including MDN (receipt) exchange. The EDIINT Module exclusively uses the S/MIME version 2 cryptographic format to package, encrypt, and provide a digital signature to outbound data, as well as to unpack, decrypt, and verify the authenticity of inbound data.

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

- **webMethods Modeler** is a design-time tool that you can use to create process models that define how to include EDI documents in *business processes* (also called *conversations*). After you design the process models, you generate them to create the run-time elements (e.g., flow services, triggers, etc.) that reside in the Integration Server. The process run time facility of the Integration Server executes the business processes (conversations) at run time.

To use the process management facility of the Integration Server for EDI documents, you must use Trading Networks. At run time, after Trading Networks performs its processing, it can pass documents to the process run time to perform the logic that you designed in a process model. For more information about designing process models, see *webMethods Modeler User's Guide*.

- **webMethods Monitor** allows you to monitor the progress and status of the business processes (conversations) involving EDI documents. webMethods Monitor interacts with the process run time to obtain the status information.

Using the EDI Module without Other webMethods Components

■ Overview	14
■ Processing of Inbound EDI Documents	14
■ Forming EDI Documents to Send Outbound	21

Overview

When you use webMethods EDI Module (EDI Module) without other webMethods' components, you are using only the WmEDI package of the EDI Module. The WmEDI package provides a toolkit of built-in services that you use as the building blocks for creating your own EDI solution.

There are two aspects to your EDI solution:

- How to send EDI documents to the Integration Server and how you process them; that is, inbound processing
- How you form EDI documents from internal-format documents (e.g., documents from back-end systems) and send the EDI documents outbound; that is, outbound processing

Processing of Inbound EDI Documents

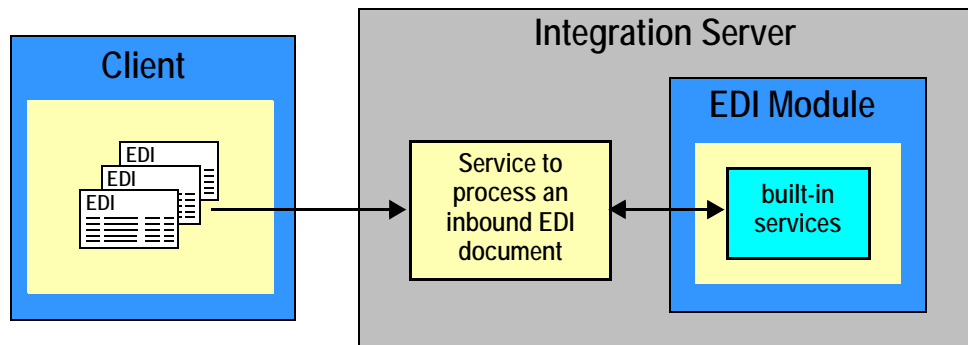
For inbound processing, you create:

- Clients that send EDI documents to the Integration Server
- Services that process the inbound EDI document

The EDI documents are documents in standard EDI format, such as ANSI X12, UCS, VICS, UN/EDIFACT, ODETTE, or EANCOM.

 **Note:** Support for the TRADACOMS standard is provided when you use the EDI Module in conjunction with webMethods Trading Networks, as described in [Chapter 3, “Using the EDI Module with Trading Networks”](#).

Inbound EDI document processing



EDI Client

You create one or more clients to send EDI documents to the Integration Server. A client can use one of the following transports to send the EDI document to the Integration Server:

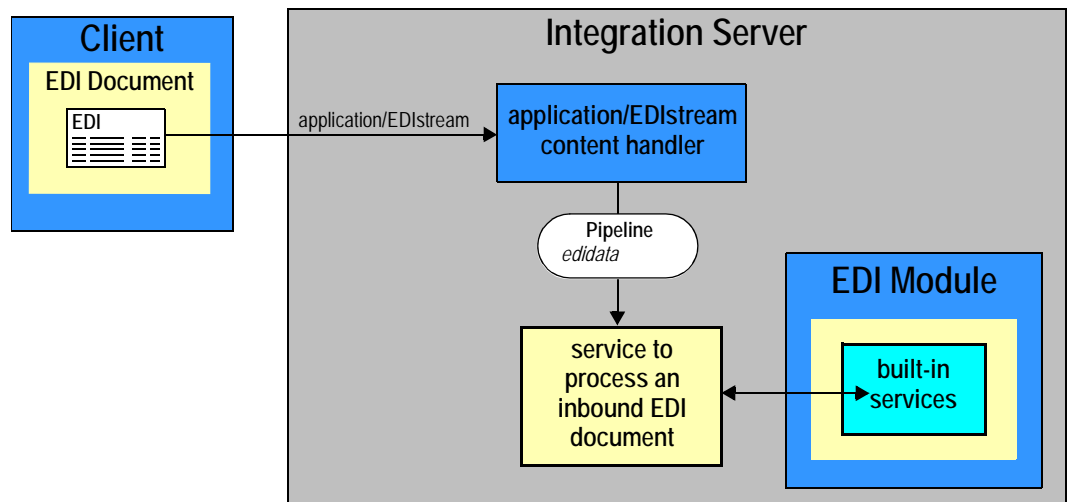
- HTTP or HTTPS
- FTP
- File Polling (File Polling is a feature of the Integration Server that allows your client to drop the document into a directory that the Integration Server monitors. When documents arrive in the directory, the EDI Module can process them.)



Note: If you want to use EDIINT to transport your EDI documents, use the EDIINT Module and Trading Networks. For more information, see the *webMethods EDIINT Module User's Guide*.

When your client sends the EDI document to the Integration Server, the client must associate the document with a content type that the EDI Module recognizes, for example, `application/EDistream`. When the Integration Server receives a document that has an EDI content type, it passes the document to the appropriate EDI content handler. The EDI Module contains built-in content handlers that the Integration Server loads when the WmEDI package is loaded.

Client sends EDI document to Integration Server



The EDI content handler performs initial processing on the document. Part of the initial processing is to create the pipeline. When it creates the pipeline, the EDI content handler creates the variable `edidata` in the pipeline. The service you create to process the EDI document can access the EDI document using the `edidata` variable. After performing the

initial processing, the EDI content handler invokes the service that you create to process the EDI document.

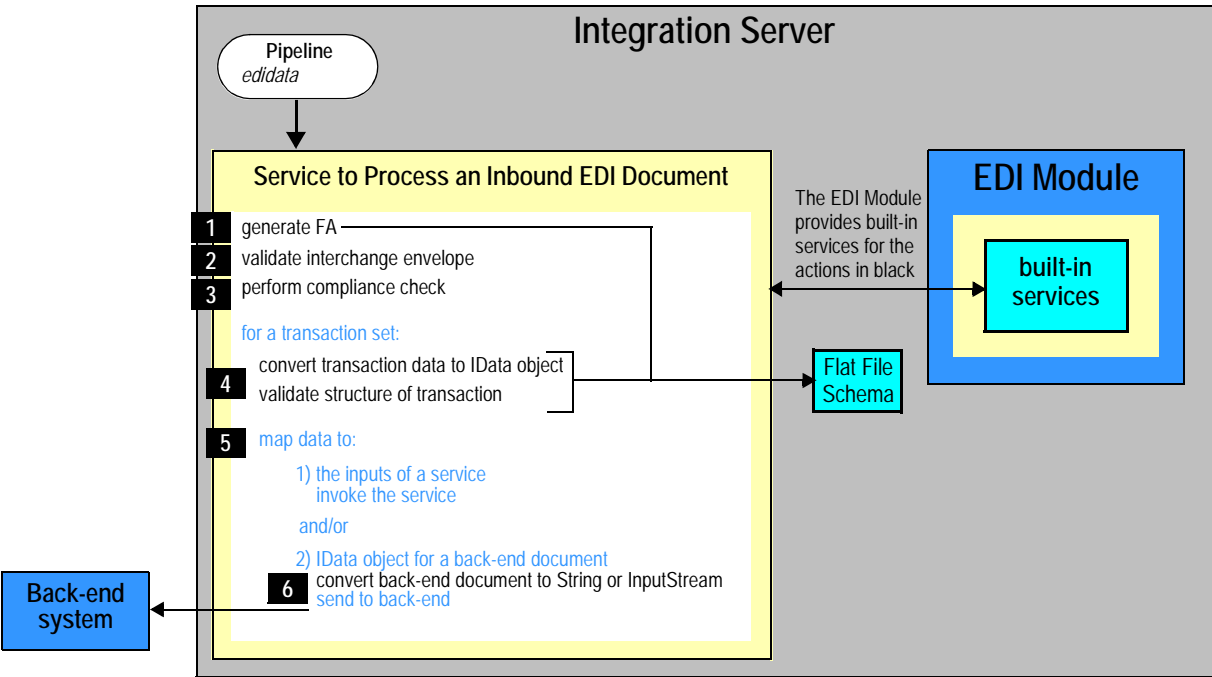
For information about: How to create the client, see Chapter 2, "Creating Clients that Send EDI Documents to the Integration Server" in the *webMethods EDI Module User's Guide*.

Service to Process the Inbound EDI Document

The EDI Module provides built-in services that you use as building blocks for creating services that process your inbound EDI documents. Typical ways to process an EDI document might be to map the data from the EDI document to another format (e.g., the format that a back-end system requires) or to map data from the EDI document to the inputs of a service.

The following diagram illustrates the basic processing you might want to include in a service that processes an EDI document. In the diagram, the service lists actions in both black and blue. The actions in black are those for which the EDI Module provides built-in services. For more information, see the tables after the diagram.

Service for processing inbound EDI documents



The service receives the EDI document in the *edidata* variable in the pipeline. The tables below provides more details about the type of processing the service can do:

Action	Description	Built-in service provided for?
1	Generate a functional acknowledgement (FA) for the EDI document. For more information, see “Functional Acknowledgments” on page 19 .	Yes
2	Perform an interchange envelope validation that includes validating field lengths, code lists, ranges, and partitions.	Yes
3	Perform a compliance check to check for matching interchange control numbers, matching group control numbers, matching transaction control numbers, segment counts, transaction counts, and group counts.	Yes

When processing an EDI document, the majority of the effort will most likely be in processing the individual transaction sets within the EDI document. You can perform the following when processing a transaction set within the EDI document:

Action	Description	Built-in service provided for?
4	Convert the EDI transaction set from a String or InputStream into an IData object and validate its structure. To be able to convert a transaction set to an IData object, the EDI Module uses a flat file schema that defines the structure of the transaction set. Additionally, it uses the flat file schema to validate that the structure of the EDI transaction is correct. For more information, see “Flat File Schemas that Define EDI Documents” on page 18 .	Yes
5	Map data from an EDI transaction to a target. After the transaction set is in an IData object, you can map the data from the IData object for the EDI transaction set. For example, you can map the data to the inputs of another service or to an internal-format document (e.g., a format required by a back-end system).	services to help map are provided

Action	Description	Built-in service provided for?
6	<p>If the service mapped EDI data to an internal-format document (e.g., the back-end system document), convert the internal-format document from an IData object to a String or InputStream.</p> <p>To be able to convert this document, the EDI Module uses a flat file schema that defines the structure of the your back-end system. For more information, see “Flat File Schemas that Define Internal-Format Documents” on page 19.</p> <p>The internal-format document as a String or InputStream is now in a format that you can use to deliver it (e.g., to the back-end system).</p>	Yes

For information about: How to create a service to process an inbound EDI document, see Chapter 3, "Receiving and Processing Inbound EDI Documents" in the *webMethods EDI Module User's Guide*.

Flat File Schemas

A *flat file schema* is the blueprint that contains the instructions for parsing or creating a flat file. It resides as a namespace element in the Integration Server. This blueprint details the structure of the document. A flat file schema also acts as the model against which you validate a flat file.

You can create two kinds of flat file schemas:

- Schemas that define the structure of EDI documents
- Schemas that define the structure of your internal-format documents (e.g., documents required for back-end systems)

Flat File Schemas that Define EDI Documents

The EDI Module uses this kind of flat file schema to convert documents from String or InputStream format to an IData object, and to validate the structure of the converted document. You create flat file schemas for EDI documents so you can convert and validate them.

The EDI Module provides built-in services that you can use to create flat file schemas that define the format of EDI documents. The built-in services allow you to create flat file schemas for EDI documents from:

- A Standard Exchange Format (SEF) file. A SEF file is a file in Standard Exchange Format, a guideline standard placed in public domain by Foresight Corporation (www.foresightcorp.com). A SEF file contains all of the metadata for the EDI standard. In other words, all of the transaction sets and segments for a particular version of EDI standard.
- An IDOC (for SAP system users).

For information about: How to create this kind of flat file schema, see Chapter 1, "Before You Can Process EDI Documents" in the *webMethods EDI Module User's Guide*.

Flat File Schemas that Define Internal-Format Documents

The EDI Module uses this kind of flat file schema to convert EDI data that you mapped to an internal-format document (e.g., a back-end system document). It converts the internal-format document from an IData object to a String or InputStream that you can send to your back-end system.



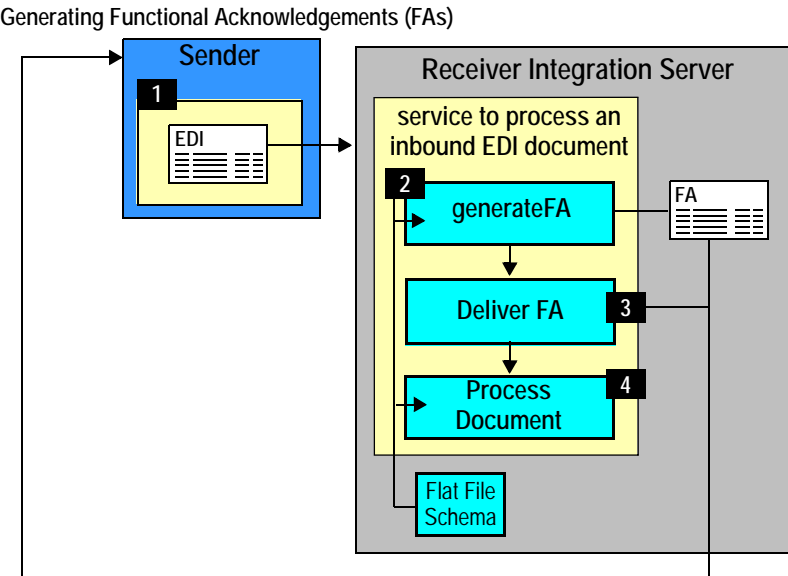
Note: To create this kind of schema, you use the functionality provided in the WmFlatFile package, *not* the EDI Module. For complete instructions, see the *Flat File Schema Developer's Guide*.

Functional Acknowledgments

A *functional acknowledgement (FA)* is a type of EDI transaction set that acknowledges the receipt, as well as the structural and syntactical validity of an EDI document. When you receive a document, you can choose to generate an FA, which sends an EDI FA document to the sender to acknowledge receipt of the document. FAs validate and acknowledge only the syntax of the document, not that the document has been processed or understood by the receiver.

The EDI Module provides a built-in service that you can invoke to generate an FA. The service does *not* specify what to do with the FA that it creates. You must add additional logic to your service to deliver the FA to the sender of the original document.

The following diagram illustrates the basics of the FA generation process. For more information, see the table following this diagram.



Step	Description
1	The sender creates a client to send an EDI document to the receiver.
2	<p>The EDI document is passed to the service that you create to process the EDI document in the pipeline variable <i>edidata</i>. You add logic to your service to invoke the built-in service that the EDI Module provides to generate the FA.</p> <p>To generate the FA, the built-in service uses the flat file schema associated with the inbound document’s EDI standard, version, and transaction set to validate the inbound EDI document. Additionally, the built-in service uses a flat file schema associated with the FA’s EDI standard and version to properly create the FA.</p>
3	Your service delivers the FA to the sender by performing logic that you define.
4	Your service continues its processing of the EDI document (for example, as described in “Processing of Inbound EDI Documents” on page 14).

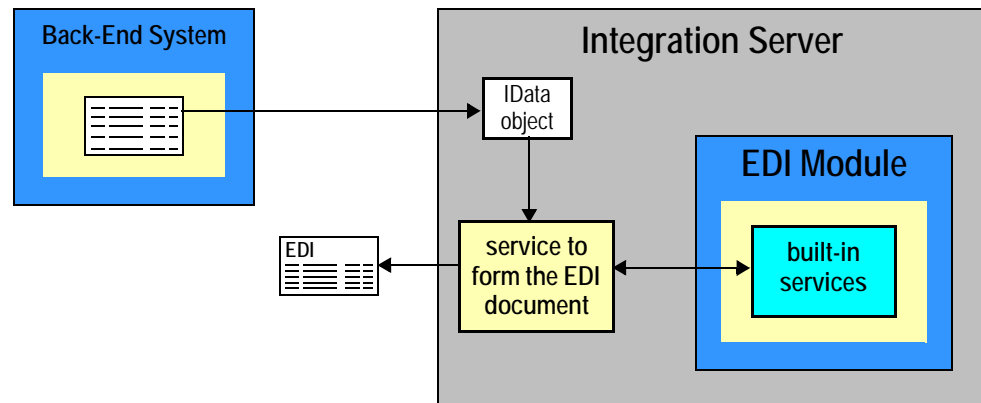
For information about: How to add logic to services to generate FAs, see Chapter 4, "Generating Functional Acknowledgements" in the *webMethods EDI Module User’s Guide*.

Forming EDI Documents to Send Outbound

For outbound processing, you form an EDI document that can be sent outbound. For example, you might use data from an internal document (e.g., a document from a back-end system) to form the EDI document.

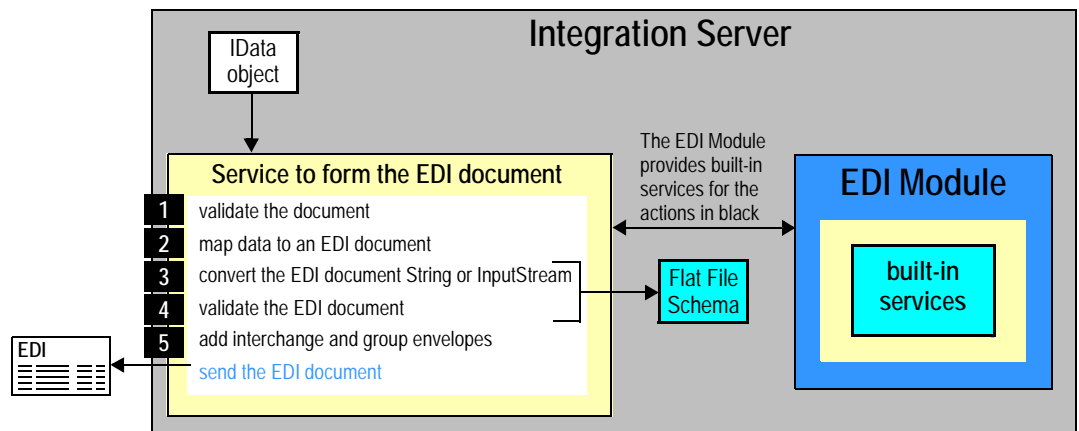
To form the EDI document, you create a service. The EDI Module provides built-in services that you can use as building blocks for creating the service.

Outbound EDI document processing



The following diagram illustrates the basic processing you might want to include in a service that forms an EDI document. In the diagram, the service lists actions in both black and blue. The actions in black are those for which the EDI Module provides built-in services. For more information, see the table after the diagram.

Service for forming outbound EDI documents



The above diagram shows the internal document being passed to the service as an IData object. This might be the case, for example, if the document is being passed to your service by an adapter service (e.g., one created with the webMethods SAP R/3 Adapter). The

internal document might also be passed to the service as a String or InputStream. If this is the case, you can use built-in services that are provided with the EDI Module to convert the String or InputStream to an IData object.

The tables below provides more details about the type of processing the service can do:

Action	Description	Built-in service provided for?
1	Validate the internal document. If you want, you can validate the incoming document when it is an IData object to ensure its structure is valid before you form the EDI document.	Yes, provided with Integration Server
2	Map data from the internal document to the EDI document.	services to help map are provided
3	Convert the EDI document from an IData object to a String or InputStream. To be able to convert the EDI document, the EDI Module uses a flat file schema that defines the structure of the EDI document it is forming. For more information, see “Flat File Schemas” on page 18 .	Yes
4	Validate the EDI document. Before you send the EDI document outbound, you can validate it to ensure its structure is correct. To be able to validate the EDI document, the EDI Module uses a flat file schema. For more information, see “Flat File Schemas” on page 18 .	Yes
5	Add interchange and group envelopes to the EDI document to form the final EDI document.	Yes

For information about: How to create a service to form an outbound EDI document, see Chapter 6, "Forming EDI Documents and Sending Them Outbound" in the *webMethods EDI Module User's Guide*.

Using the EDI Module with Trading Networks

■ Overview	24
■ Processing Inbound EDI Documents with Trading Networks	25
■ Forming EDI Documents and Sending Them Outbound	39
■ Working with VANs	46
■ Batching Outbound EDI Documents	51

Overview

When you use webMethods EDI Module (EDI Module) with webMethods Trading Networks (Trading Networks), you use the features of Trading Networks to exchange EDI documents with your trading partners. The following lists Trading Networks objects that the EDI Module uses to process EDI documents and that you must define. The rest of this chapter describes how the EDI Module uses the following Trading Networks objects:

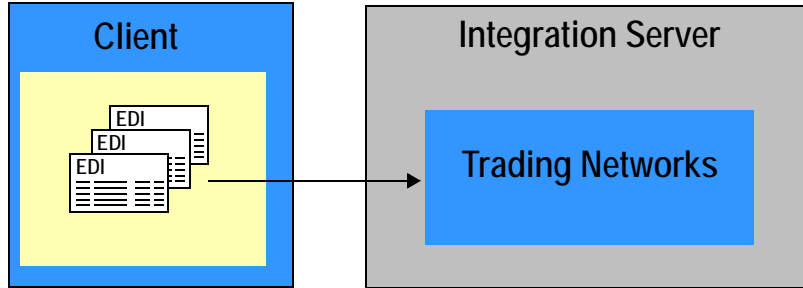
- **Profiles** for the partners with whom you want to exchange EDI documents.
- **Trading partner agreements (TPAs)** that contain EDI-specific information. The TPAs that the EDI Module uses are referred to as EDITPAs. The EDITPAs contain information that you specify to tailor how the EDI Module processes inbound EDI documents and how the EDI Module forms EDI documents to send outbound.
- **TN document types** that Trading Networks uses to recognize documents. The EDI Module provides the TN document types for EDI documents, and you can install them from the EDI Module home page. You only need to install the TN document types for the types of EDI documents you plan to exchange.
- **Processing rules** that specify how to process documents. You specify the actions you want Trading Networks to take against documents. For example, for an inbound EDI document, you might use the **Execute a Service** action invoke a service that you create; for an outbound EDI document, you might use the **Deliver Document By** action to have Trading Networks deliver the outbound EDI document.
- **Public queues** to hold EDI documents. You define public queues in Trading Networks if you want to deliver documents to a VAN or if you want to batch outbound EDI documents. The EDI Module provides the scheduled delivery services that you assign to the public queues and that acts on the documents in the queue. For more information, see [“Outbound Processing: Delivering Documents to VANs” on page 48](#) and [“Batching Outbound EDI Documents” on page 51](#).

For more information about the Trading Networks objects listed above, see the *webMethods Trading Networks Concepts Guide* and the *webMethods Trading Networks User’s Guide*.

Processing Inbound EDI Documents with Trading Networks

For inbound processing, you create clients that send EDI documents to Trading Networks and you set up information in Trading Networks to process the document.

Inbound EDI document processing when using Trading Networks



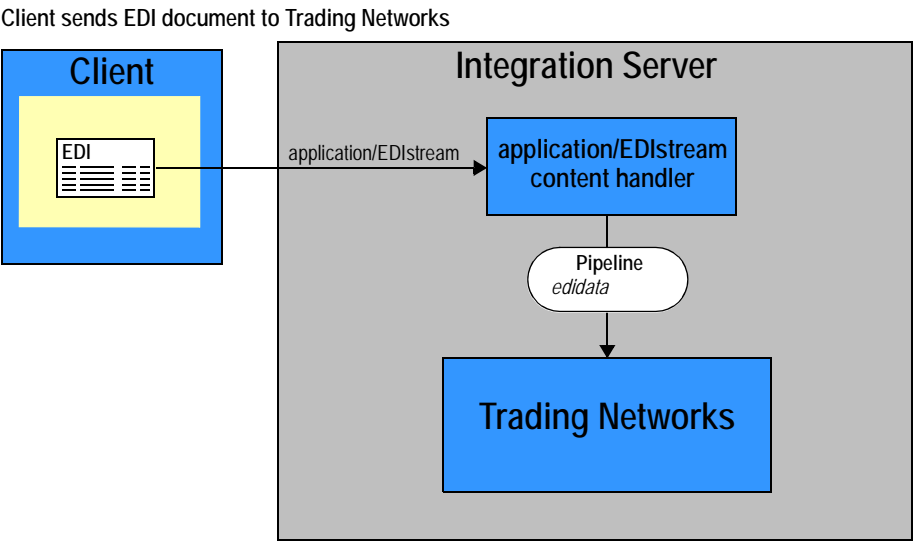
EDI Client

To send documents to the Integration Server to be processed through Trading Networks, you create a client. The client can use one of the following transports to send the EDI documents:

- HTTP or HTTPS
- FTP
- File Polling
- EDIINT AS1, EDIINT AS2, or EDIINT AS3

For more information about using EDIINT, see the *webMethods EDIINT Module User's Guide*. The rest of this section describes clients that use HTTP, HTTPS, FTP, or File Polling.

When your client sends the EDI document to the Integration Server, it must associate the document with a content type that the EDI Module recognizes, for example, `application/EDISTream`. When the Integration Server receives a document that has an EDI content type, it passes the document to the appropriate EDI content handler. The EDI content handler performs initial processing on the document, which includes creating the pipeline with the variable `edidata` that contains the EDI document. After performing the initial processing, the EDI content handler invokes the service that the client specifies, which is the `wm.in:receive` service, to have Trading Networks process the document.



For more information about how to create the client, see Chapter 12, "Creating Clients that Send EDI Documents to Trading Networks" in the *webMethods EDI Module User's Guide*.

EDI Documents Inbound Processing

Trading Networks processes ANSI X12 and UN/EDIFACT documents in an identical manner.

TRADACOMS documents are structured differently than ANSI X12 and UN/EDIFACT documents. Despite this difference, Trading Networks processes TRADACOMS documents in a manner very similar to the way it processes ANSI X12 and UN/EDIFACT documents. This is because the TRADACOMS document types are roughly analogous to the ANSI X12 or UN/EDIFACT document types, as follows:

ANSI X12 or UN/EDIFACT document type	Analogous TRADACOMS document type
Interchange	Transmission
Transaction set	File
Group	Batch
Functional Acknowledgement (FA)	No equivalent



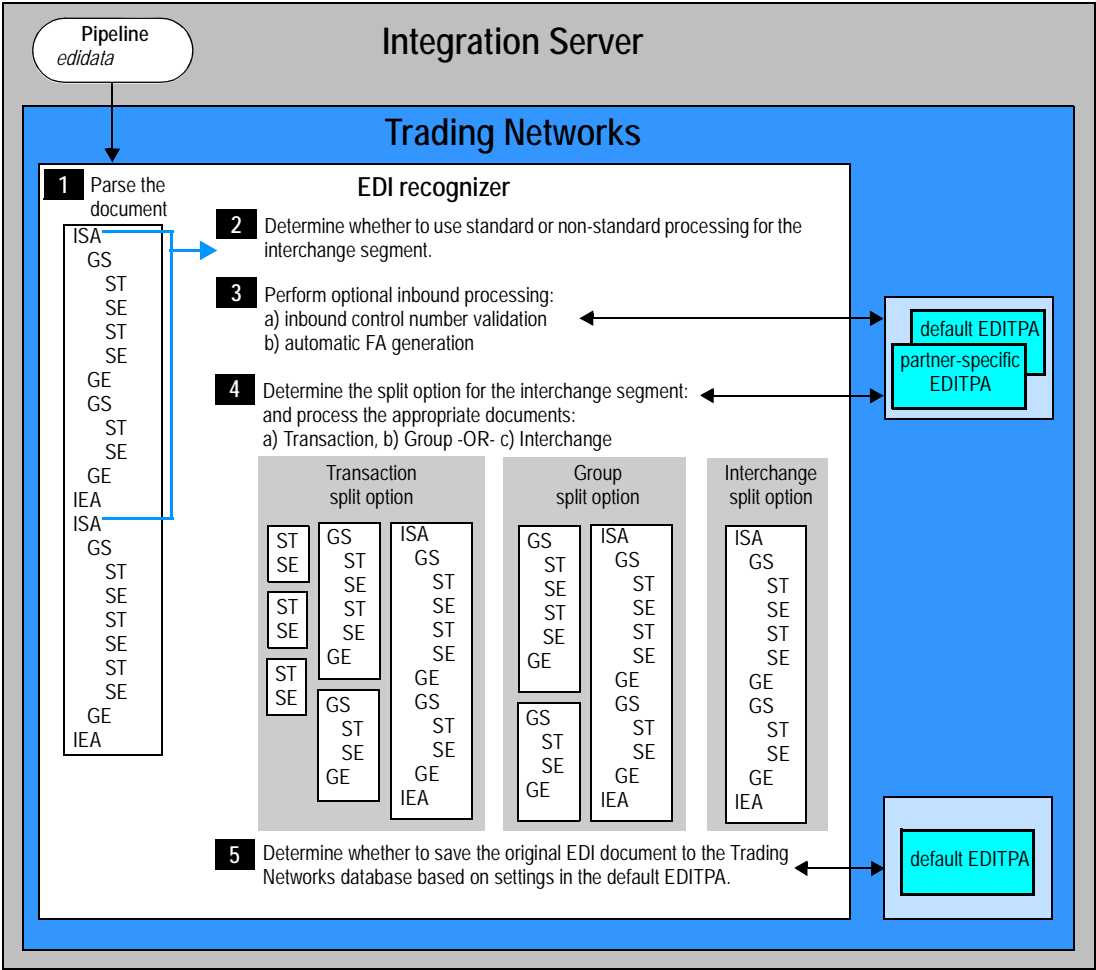
Important! Although Trading Networks can process documents of any supported EDI standard, it cannot properly process a mixture of TRADACOMS and non-TRADACOMS documents in a single submission. If the first inbound document is a TRADACOMS document, Trading Networks considers any subsequent non-TRADACOMS documents to be of the Unknown document type. Similarly, if the first inbound document is a non-TRADACOMS document, Trading Networks considers any subsequent TRADACOMS documents to be of the Unknown document type.

Initial Processing of the EDI Recognizer

When you install the EDI Module, the WmEDIforTN package of the EDI Module enhances the capabilities of Trading Networks to allow it to recognize and to begin processing EDI documents. The WmEDIforTN package adds an *EDI recognizer* to Trading Networks.

The following diagram illustrates the initial processing that the EDI recognizer performs against an ANSI X12 document. If you use the TRADACOMS standard, as you view this diagram remember that a reference to an *interchange* is a reference to a TRADACOMS transmission; a *transaction set* is a TRADACOMS file, and a *group* is a TRADACOMS batch.

Recognizing the original EDI document and beginning processing



Step	Description
1	<p>When Trading Networks receives an EDI document, it passes the EDI document to the EDI recognizer. The EDI recognizer parses the inbound EDI document, so subsequent processing can process the following types of documents:</p> <ul style="list-style-type: none"> ■ For ANSI X12 and UN/EDIFACT documents: <ul style="list-style-type: none"> ■ Transaction documents that each contain a single transaction set. ■ Group documents that contain a single group segment along with its transaction sets. ■ Interchange documents that contain a single interchange envelope along with its group segments and transaction sets. ■ For TRADACOMS documents: <ul style="list-style-type: none"> ■ File documents that each contain multiple TRADACOMS messages. ■ Batch documents that contain a single TRADACOMS batch segment along with its files. ■ Transmission documents that each contain a single TRADACOMS transmission envelope along with its batch segments and files. <p>When parsing, the EDI recognizer also performs envelope validation and compliance check and places the results in the <i>errors</i> variable in the pipeline.</p>
2	<p>For each interchange segment in the original EDI document, the EDI recognizer determines whether to use standard or non-standard processing for the interchange segment. The type of processing (standard or non-standard) determines which <i>EDI trading partner agreement (EDITPA)</i> the EDI recognizer retrieves to obtain settings that tailor how to process documents from the interchange segment. For more information, see “EDI Trading Partner Agreements (EDITPAs)” on page 38.</p> <hr/> <p>Note: Non-standard processing is not applicable with TRADACOMS documents.</p> <hr/>

Step	Description
------	-------------

- | | |
|--|---|
| | <ul style="list-style-type: none">■ For standard processing, the EDI Module uses all standard Trading Networks objects for processing. The EDI recognizer obtains profiles for the sender and receiver identified in the interchange header (or TRADACOMS STX segment). Then it retrieves the EDITPA for the sender/receiver pair specified in the interchange header or TRADACOMS STX segment. When you use standard processing, all documents (Transaction, Group, and Interchange) within the interchange segment or TRADACOMS STX segment are processed using the settings you defined for the sender/receiver specified in the interchange header or TRADACOMS STX segment.■ For non-standard processing (which is not applicable for TRADACOMS documents), the EDI Module uses custom interchange sender/receiver pair information that you must define. This interchange sender/receiver pair information is stored in the Trading Networks database, but is <i>not</i> associated with a standard Trading Networks object. |
|--|---|

The EDI recognizer uses the interchange sender/receiver pair information to determine the EDI ID qualifiers to use with the senders and receivers on the group headers within the interchange segment. Then for each group in the interchange segment, the EDI recognizer obtains the profiles for the sender and receiver identified in the group header. It then retrieves the EDITPA that corresponds to the sender/receiver specified in the group header. When you use non-standard processing, Transaction and Group documents are processed using the settings you defined for the sender/receiver specified in the group header.

Note: If you want to use non-standard processing, you must use the WmEDIforTN package home page to define interchange sender/receiver pair information. The EDI recognizer determines whether to perform standard or non-standard processing based on the existence of this interchange sender/receiver pair information. That is, if the sender/receiver pair information exists for the sender/receiver of the interchange segment, the EDI Module performs non-standard processing. If it does *not* exist, the EDI Module performs standard processing.

Step	Description
3	<p>Performs optional inbound processing based on how you set up the EDI Module. The EDI Module can:</p> <ul style="list-style-type: none">■ Perform inbound control number validation for control numbers in the interchange and/or group headers (or in the TRADACOMS transmission and/or batch) of the inbound document. <p>Note: The term <i>control number</i> is a term used in the EDI standards ANSI X12 and UN/EDIFACT. It refers to a number in the header of an EDI document that is used for validation and for the ordering of documents exchanged between trading partners. If you use the TRADACOMS EDI standard, the term <i>control number</i> is equivalent to the transmission reference numbers specified in the STX and BAT segments of your TRADACOMS documents. Whether your EDI standard includes control numbers or transmission reference numbers, you define them to Trading Networks in the same manner; the only difference is in the terminology. For simplicity, Trading Networks and the EDI Module use the term <i>control number</i> to mean either control number or transmission reference number.</p> <ul style="list-style-type: none">■ Automatically generate functional acknowledgments (FAs). For an ANSI X12 document, the EDI Module generates an FA for each group in the inbound document. For a UN/EDIFACT document, the EDI Module generates an FA for each interchange in the inbound EDI document. Functional acknowledgments are not applicable to the TRADACOMS EDI standard.

Step	Description
4	<p>Using the EDITPA that it retrieved, the EDI recognizer determines the types of documents that you want to process using the split option variable within the EDITPA. You can set it to one of the following:</p> <ul style="list-style-type: none">■ For ANSI X12 and UN/EDIFACT documents:<ul style="list-style-type: none">■ Transaction split option to have the EDI recognizer submit all Transaction, Group, and Interchange documents to Trading Networks for processing.■ Group split option to have the EDI recognizer submit all Group and Interchange documents to Trading Networks for processing.■ Interchange split option to have the EDI recognizer submit only Interchange documents to Trading Networks for processing. You cannot use this setting when using non-standard processing.■ For TRADACOMS documents:<ul style="list-style-type: none">■ File split option to have the EDI recognizer submit all File, Batch, and Transmission documents to Trading Networks for processing.■ Batch split option to have the EDI recognizer submit all Batch and Transmission documents to Trading Networks for processing.■ Transmission split option to have the EDI recognizer submit only Transmission documents to Trading Networks for processing. <p>See “Processing Documents in Trading Networks” below for information about processing each type of document that results from the split of the interchange segment.</p>
5	<p>If the original EDI document contains multiple interchanges or TRADACOMS transmissions, after splitting documents from the document and sending them for processing, the EDI recognizer can save the entire original EDI document to the Trading Networks database. It determines whether to save the original EDI document based on a setting that you define in the default EDITPA.</p>

For more information about:

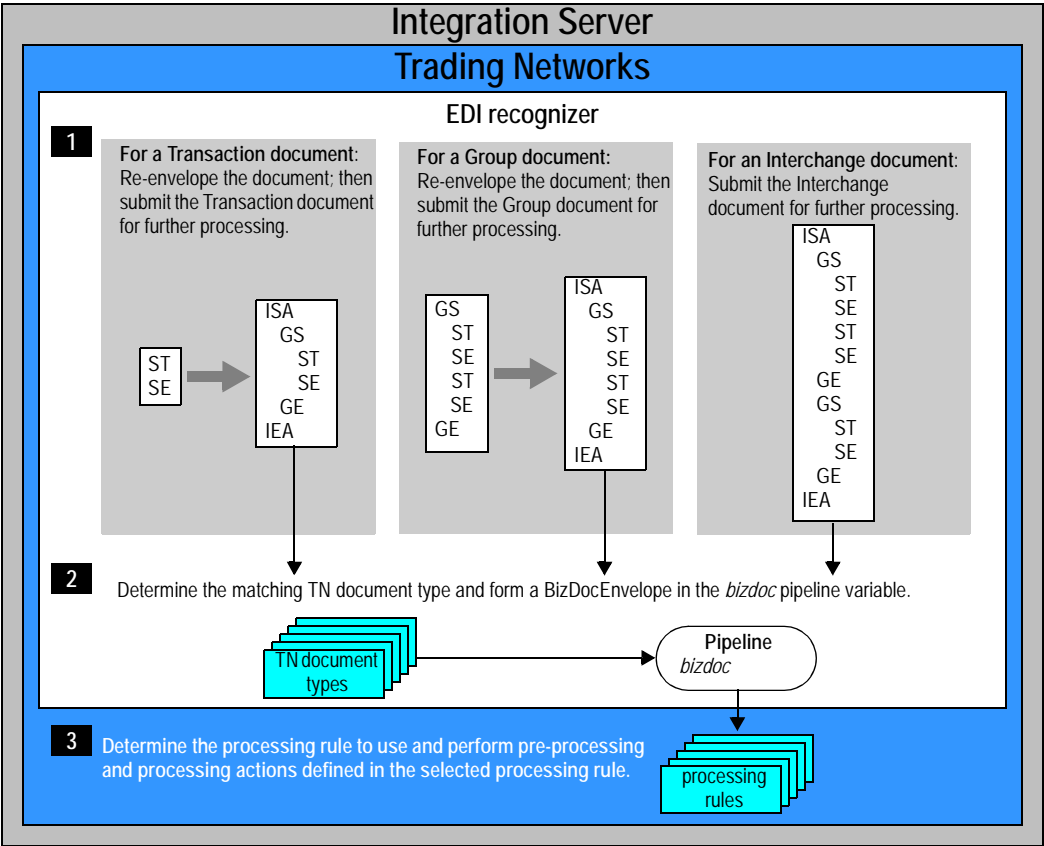
- Setting up the default and partner-specific EDITPAs and defining the variables in them (including the variable that governs the split option), see Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT)", or Chapter 10, "Defining Partner Information (TRADACOMS)" in the *webMethods EDI Module User's Guide*.
- Non-standard processing, see Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT)", and Appendix A, "Non-Standard Processing" in the *webMethods EDI Module User's Guide*.
- Inbound control number validation, see Chapter 15, "Optional Inbound Processing When using Trading Networks", in *webMethods EDI Module User's Guide*.
- Automatic FA generation, see Chapter 15, "Optional Inbound Processing When using Trading Networks", in *webMethods EDI Module User's Guide*.

Processing Documents in Trading Networks

After determining the split option from the EDITPA, the EDI recognizer creates the appropriate documents (e.g., Transaction, Group, and Interchange) based on the split option and submits each document individually to Trading Networks for processing.

The following diagram illustrates the processing for each type of ANSI X12 document. Again, if you use the TRADACOMS standard, as you view this diagram remember that a reference to an *interchange* is a reference to a TRADACOMS transmission; a *transaction set* is a TRADACOMS file, and a *group* is a TRADACOMS batch.

Further processing of Transaction, Group, and Interchange documents



Step	Description
1	<p>The EDI recognizer does the following:</p> <ul style="list-style-type: none">■ For an ANSI X12 or UN/EDIFACT document:<ul style="list-style-type: none">■ When the split option is Transaction, the EDI recognizer forms a Transaction document for each transaction set within an interchange segment. To form a Transaction document, the EDI recognizer re-envelopes a transaction set with:<ul style="list-style-type: none">■ A group envelope using the group header from the group segment in which the transaction set resided in the original EDI document-AND-■ An interchange envelope using the interchange header from the interchange segment in which the transaction set resided in the original EDI document

Step	Description
	<ul style="list-style-type: none"> ■ When the split option is Group, the EDI recognizer forms a Group document for each group segment within an interchange segment. To form a Group document, the EDI recognizer re-envelopes the group segment with an interchange envelope using the interchange header from the interchange segment in which the group resided in the original EDI document. ■ When the split option is Interchange, the EDI recognizer forms an Interchange document comprised of the information from the interchange segment in the original EDI document. ■ For a TRADACOMS document: <ul style="list-style-type: none"> ■ When the split option is File, the EDI recognizer forms a Transmission document, a Batch document (if present), and one File document for the file within the transmission. ■ When the split option is Batch, the EDI recognizer forms a Transmission document and a Batch document. ■ When the split option is Transmission, the EDI recognizer forms only a Transmission document.
2	<p>After the EDI recognizer forms each document, it resubmits the newly formed document (e.g., Transaction, Group, or Interchange) to Trading Networks for processing. Because the newly formed document is an EDI document, Trading Networks passes the document to the EDI recognizer for processing.</p> <p>The EDI recognizer does not parse the newly formed document again. The parsed version of the original EDI document remains in the pipeline. It uses the newly formed EDI document to recognize the type of document, to add as a content part to the BizDocEnvelope, and to save to the Trading Networks database.</p> <p>The EDI recognizer uses the TN document types to determine the type of document. The EDI Module provides the TN document types for EDI documents. You install the TN document types for the types of EDI documents that you will want to process. For example, if you want to process the 4010 version of the ANSI X12 850 EDI document, you would install the X12 4010 850, X12 Group, and X12 Envelope TN document types into Trading Networks.</p> <p>After recognizing the type of document using TN document types, the EDI recognizer forms a <i>BizDocEnvelope</i> for the EDI document. The BizDocEnvelope is in the <i>bizdoc</i> pipeline variable. A BizDocEnvelope contains the original document (e.g., Transaction, Group, or Interchange) and includes additional information that Trading Networks requires for routing and processing the document. In other words, the BizDocEnvelope represents a routable Trading Networks transaction.</p>

Step	Description
3	After forming the BizDocEnvelope, Trading Networks determines the processing rule to use to process the document and executes the processing rule. You create processing rules to define the processing you want performed on each type of document. For example, you can define a processing rule you that executes a service that you create to form an internal-format document based on information in the EDI document and send that internal-format document to your back-end system. For more information, see “ Processing Rules for Inbound EDI Documents ” below.

Processing Rules for Inbound EDI Documents

Processing rules specify two categories of actions: 1) pre-processing actions and 2) processing actions.

Pre-Processing Actions

Default pre-processing actions are defined in the TN document types for EDI documents. However, in the processing rule, you can override the default settings that are defined in the TN document type. If a pre-processing action fails, Trading Networks records the error and continues processing.

The pre-processing actions that you can use for EDI documents are:

- **Validate Structure.** Validate the structure of the EDI document against your flat file schema. By default, the TN document types for EDI documents indicate that validation should *not* be performed.
- **Check for Duplicate Document.** Determine whether the document is unique; that is, has Trading Networks previously received the document for processing. By default, the TN document types for EDI documents indicate that this check should *not* be performed.

The EDI Module does not add EDI Module-specific logic to Trading Networks to perform this pre-processing action. When you use this pre-processing action, Trading Networks performs its standard logic for the function. For more information, see information about pre-processing actions in the *webMethods Trading Networks Concepts Guide* and *webMethods Trading Networks User’s Guide*.

- **Save Document to Database.** Specify whether you want Trading Networks to save the document to its database. By default, the TN document types for EDI documents indicate that the content, attributes, and activity log information for EDI documents should be saved to the database.

The EDI Module does not add EDI Module-specific logic to Trading Networks to perform this pre-processing action. When you use this pre-processing action, Trading Networks performs its standard logic for the function. For more information, see information about pre-processing actions in the *webMethods Trading Networks Concepts Guide* and *webMethods Trading Networks User’s Guide*.



Note: You cannot use the **Verify Digital Signature** pre-processing action for EDI documents. This pre-processing action requires that values for the system attributes **SignedBody** and **Signature** be available to use to verify the signature. Values for these system attributes are not set for EDI documents.

Processing Actions

You can use all of the Trading Networks processing actions for EDI documents. For inbound EDI documents, typically you will use the **Execute a Service** action to process the inbound document. You create the service that the processing action invokes. The logic you add to the service depends on the split option and the type of document (e.g., Transaction, Group, or Interchange):

- **When the split option is Transaction (or File),** the EDI recognizer sends all three types of documents to Trading Networks for processing. You will create a processing rule for each type of document. The logic you add to the service for the **Execute a Service** action is based on the type of document:
 - **Transaction (or File) document.** This document contains a single transaction set or multiple TRADACOMS messages. For this level of document, add the logic that you want to perform against the transaction set or the messages. For example, you might convert the transaction set data to an IData object so you can map information from the transaction set to an internal-format document. Then the service could send the internal-format document to a back-end system.
 - **Group (or Batch) document.** If you want to generate a functional acknowledgment (FA) for an ANSI X12 or UN/EDIFACT Group document, the logic in the service might generate the FA and send the FA back to the sender.
 - **Interchange (or Transmission) document.** You might set up the processing rule to ignore this document (that is, perform no processing action) because you have completed the processing in the other documents.
- **When the split option is Group (or Batch),** the EDI recognizer sends Group and Interchange documents (or Batch and File documents) to Trading Networks for processing. You will create a processing rule for each type of document and specify processing actions based on the type of document:
 - **Group (or Batch) document.** In the logic you add to the service for the **Execute a Service** action, you might generate an FA and send the FA to the sender. Then, you might loop through each of the transaction sets in the group segment and perform processing against each. For example, you might convert the transaction set data to an IData object so you can map information from the transaction set to the inputs of another service, then invoke the service.
 - **Interchange (or Transmission) document.** You might set up the processing rule to ignore this document (that is, perform no processing action) because you have completed the processing in the logic for the Group or Batch document.

- When the split option is **Interchange** (or **Transmission**), the EDI recognizer sends only an Interchange or Transmission document to Trading Networks for processing. You will create a processing rule for the Interchange or Transmission document. In the logic you add to the service for the **Execute a Service** action, you might generate an FA and send the FA to the sender. Then, you might loop through document to access the information in the transaction sets, so you can perform processing against each.

For more information about:

- Defining processing rules for Transaction, Group, and Interchange documents, see Chapter 13, "Processing Inbound ANSI X12 and UN/EDIFACT EDI Documents using Trading Networks", in the *webMethods EDI Module User's Guide*.
- Defining processing rules for File, Batch, and Transmission documents, see Chapter 14, "Processing Inbound TRADACOMS EDI Documents using Trading Networks", in the *webMethods EDI Module User's Guide*.

EDI Trading Partner Agreements (EDITPAs)

An EDI Trading Partner Agreement (EDITPA) is a set of variables that you specify to tailor how the EDI Module processes documents that are exchanged between two trading partners. For example, an EDITPA contains the split option variable that indicates what level of documents (e.g., Transaction, Group, or Interchange) that you want to process.

The EDI Module supports partner-specific EDITPAs and a single default EDITPA.

- A partner-specific EDITPA has a specific sender and receiver associated with it. It is specific to which partner represents the sender and which represents the receiver. Therefore, you might have two EDITPAs for one trading partner pair. For example, for trading partners A and B, you might have one EDITPA where trading partner A is the sender and B is the receiver and another for when B is the sender and A is the receiver.

A partner-specific EDITPA contains partner-specific variables used by only the specific pair of trading partners (sender and receiver) specified in the EDITPA.

- A default EDITPA has a sender and receiver set to "unknown." It contains variables used by all trading partners when partner-specific information is not available. That is, the EDI Module uses the values in the default EDITPA if a partner-specific EDITPA does not exist or if the value in the partner-specific EDITPA is null or empty.

For more information about setting up the default and partner-specific EDITPAs and defining the variables in EDITPAs, see Chapter 9, "Defining Partner Information (ANSI X12 and UN/EDIFACT)", or Chapter 10, "Defining Partner Information (TRADACOMS)", in the *webMethods EDI Module User's Guide*.

Forming EDI Documents and Sending Them Outbound

For outbound processing, you form an EDI document that can be sent outbound. For example, you might use the data from an internal-format document (e.g., a document from a back-end system) to form the EDI document.

To form the EDI document you create a service. When you are using the EDI Module with Trading Networks, the way you create the service to form the EDI document is similar to the way you create the service when you are not using Trading Networks. For a description, see [“Forming EDI Documents to Send Outbound” on page 21](#).

A difference in how you create the service is that when you are using Trading Networks, you can access information for headers that the EDI Module maintains in the Trading Networks database, for example, in the EDITPAs. Additionally, to deliver an outbound EDI document, rather than write your own logic to deliver the EDI document, if you want to send the outbound document to a VAN or batch it for delivery, you can use features of the EDI Module and Trading Networks to do so.

To form the outbound EDI document and deliver it, you would:

- Define a TN document type for the internal-format document.
- Define a processing rule that processes the internal-format document. This processing rule should use the **Execute a Service** processing action to invoke the service that you create to form the outbound EDI document.

To deliver the outbound EDI document, you can:

Method	Notes
Add logic to deliver the document to your service that forms the outbound EDI document.	<ul style="list-style-type: none"> ■ The outbound EDI document is <i>not</i> saved in the Trading Networks database. ■ You <i>cannot</i> use this method if you want to batch the outbound EDI document or deliver the outbound EDI document to a VAN.
Submit the outbound EDI document to Trading Networks document recognition.	<ul style="list-style-type: none"> ■ This method requires that you have a TN document type for the outbound EDI document. ■ This method requires that you have a second processing rule to deliver the outbound EDI document. ■ This method allows you to save the outbound EDI document to the Trading Networks database before delivering it. ■ You can use this method to deliver the outbound EDI document to a VAN or to batch the outbound EDI document.

Method	Notes
Route the outbound EDI document to Trading Networks processing rules.	<ul style="list-style-type: none">■ This method bypasses document recognition, therefore you do <i>not</i> need a second TN document type for the outbound EDI document.■ This method requires that you have a second processing rule to deliver the outbound EDI document.■ This method allows you to save the outbound EDI document to the Trading Networks database before delivering it.■ You can use this method to deliver the outbound EDI document to a VAN or to batch the outbound EDI document.

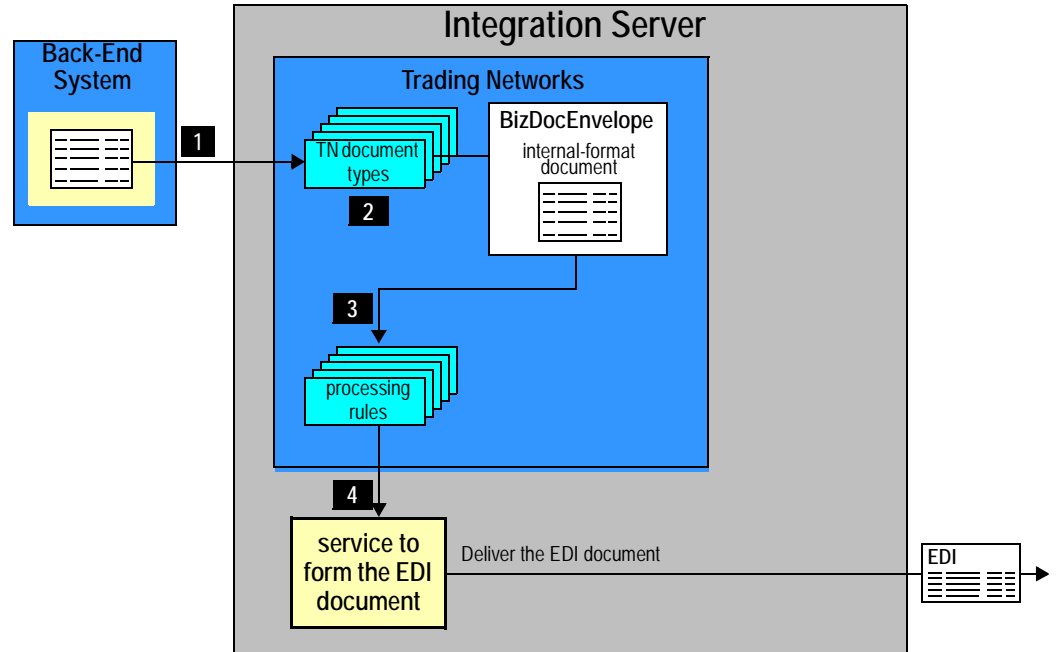
For more information about how to create a service to form an outbound EDI document when you are using Trading Networks, and how to set up Trading Networks to deliver the outbound EDI document, see Chapter 16, "Forming EDI Documents to Send Outbound When Using Trading Networks", in the *webMethods EDI Module User's Guide*.

Delivering the EDI Document Directly from the Service that Forms It

The following diagram illustrates the process of receiving an internal-format document and using a processing rule to invoke a service that you create to:

- Form the outbound EDI document based on the internal-format document.
- Deliver the outbound EDI document.

Delivering the EDI Document directly from the service that forms it



Step	Description
1	A back-end system or client sends an internal-format document to the Integration Server invoking the <code>wm.tn:receive</code> service to send the document to Trading Networks.
2	Trading Networks matches the internal-format document to its TN document types. You need to create a TN document type that will match your internal-format. For more information about creating TN document types, see the <i>webMethods Trading Networks User's Guide</i> . After selecting the appropriate TN document type, Trading Networks forms the BizDocEnvelope that contains the internal-format document as the content and places the BizDocEnvelope in the pipeline in the <i>bizdoc</i> variable.
3	Trading Networks searches its processing rules to find the appropriate rule to use to process the internal-format document. You should create a processing rule that uses the Execute a Service processing action to invoke a service that you create to form the EDI document. For more information about creating this service, see “Forming EDI Documents to Send Outbound” on page 21 .
4	The service you create to form the EDI document executes. After forming the EDI document, your service invokes logic that you create to deliver the outbound EDI document.

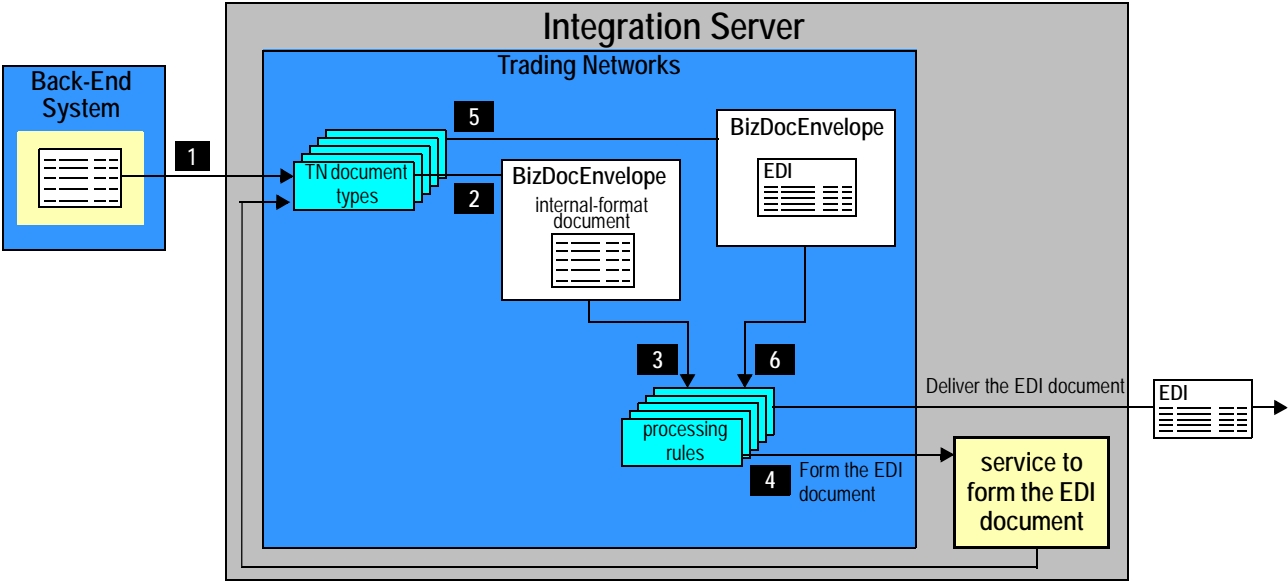
For more information about how to create the service to form an outbound EDI document and send the outbound EDI document directly from the service, see Chapter 16, "Forming EDI Documents to Send Outbound When Using Trading Networks", in the *webMethods EDI Module User's Guide*.

Submitting the Outbound EDI Document to Trading Networks to Deliver It

The following diagram illustrates the process of receiving an internal-format document and using a processing rule to invoke a service that you create to:

- Form the outbound EDI document based on the internal-format document.
- Submit the outbound EDI document back to Trading Networks document recognition to process the outbound EDI document through Trading Networks to deliver it.

Submitting the outbound EDI document to Trading Networks to deliver it



Step	Description
1	A back-end system or client sends an internal-format document to the Integration Server invoking the <code>wm.tn:receive</code> service to send the document to Trading Networks.

Step	Description
2	<p>Trading Networks matches the internal-format document to its TN document types. You need to create a TN document type that will match your internal-format. For more information about creating TN document types, see the <i>webMethods Trading Networks User's Guide</i>.</p> <p>After selecting the appropriate TN document type, Trading Networks forms the BizDocEnvelope that contains the internal-format document as the content and places the BizDocEnvelope in the pipeline in the <i>bizdoc</i> variable.</p>
3	<p>Trading Networks searches its processing rules to find the appropriate rule to use to process the internal-format document. You should create a processing rule that uses the Execute a Service processing action to invoke a service that you create to form the EDI document. For more information about creating this service, see “Forming EDI Documents to Send Outbound” on page 21.</p>
4	<p>The service you create to form the EDI document executes. After forming the EDI document, your service invokes the <code>wm.tn.doc.xml:routeXml</code> service to submit the outbound EDI document back into Trading Networks document recognition.</p>
5	<p>Trading Networks document recognition passes the document to the EDI recognizer. The EDI recognizer executes as described in “EDI Documents Inbound Processing” on page 26. That means it obtains EDITPA information and splits the document based on the split option in the EDITPA.</p> <p>For outbound documents, set the split option to:</p> <ul style="list-style-type: none"> ■ Group if you want the EDI Module to perform FA reconciliation in addition to delivering the outbound EDI document. ■ Interchange if you just want to deliver the outbound EDI document. <p>The EDI Module selects the TN document type for the outbound EDI document and forms the BizDocEnvelope that contains the outbound EDI document as the content and places the BizDocEnvelope in the pipeline in the <i>bizdoc</i> variable.</p>

Step	Description
6	<p>Trading Networks searches its processing rules again to find the appropriate rule to use to deliver the outbound EDI document. You need to create a processing rule for the EDI document that does one of the following:</p> <ul style="list-style-type: none">■ Uses the Execute a Service processing action that invokes a service that you created to deliver the EDI document. <p>-OR-</p> <ul style="list-style-type: none">■ Uses the Deliver Document By processing action to:<ul style="list-style-type: none">■ Send the EDI document to a VAN. For more information, see “Outbound Processing: Delivering Documents to VANs” below.■ Batch the EDI document for delivery. For more information, see “Batching Outbound EDI Documents” on page 51.

For more information about the following subjects, see Chapter 16, "Forming EDI Documents to Send Outbound When Using Trading Networks", in the *webMethods EDI Module User's Guide*:

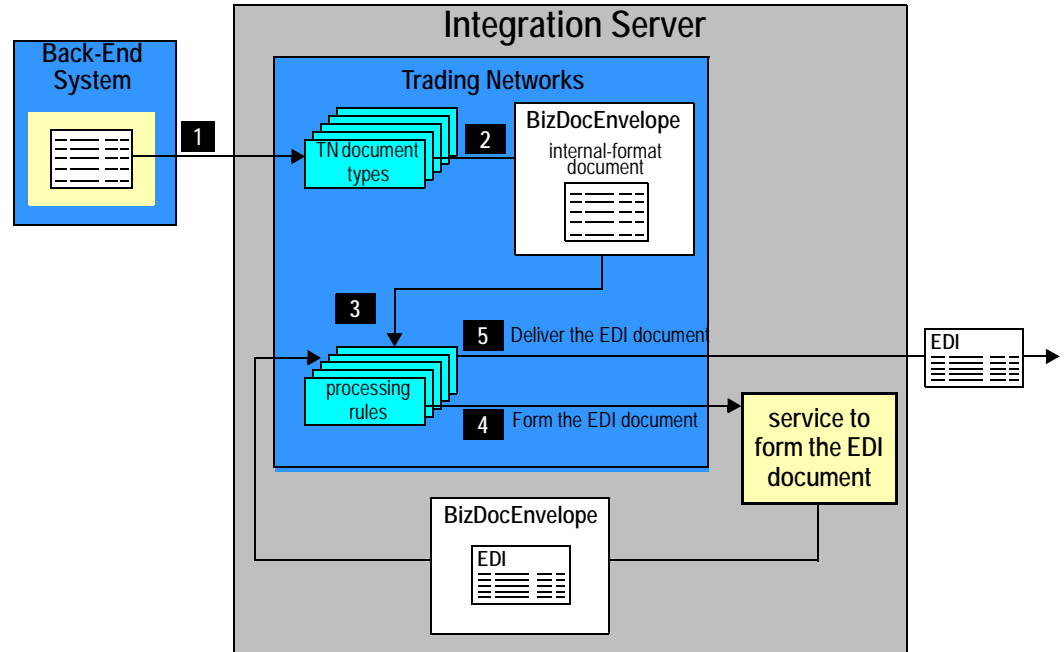
- The logic to use to submit an outbound EDI document to Trading Networks
- How to set up Trading Networks to deliver the outbound EDI document

Routing the Outbound EDI Document to Trading Networks to Deliver It

The following diagram illustrates the process of receiving an internal-format document and using a processing rule to invoke a service that you create to:

- Form the outbound EDI document based on the internal-format document.
- Route the outbound EDI document back to Trading Networks processing rules to select another processing rule to deliver the outbound EDI document.

Routing the outbound EDI document to Trading Networks to deliver it



Step	Description
1	A back-end system or client sends an internal-format document to the Integration Server invoking the <code>wm.tn:receive</code> service to send the document to Trading Networks.
2	Trading Networks matches the internal-format document to its TN document types. You need to create a TN document type that will match your internal-format. For more information about creating TN document types, see the <i>webMethods Trading Networks User's Guide</i> . After selecting the appropriate TN document type, Trading Networks forms the BizDocEnvelope that contains the internal-format document as the content and places the BizDocEnvelope in the pipeline in the <i>bizdoc</i> variable.
3	Trading Networks searches its processing rules to find the appropriate rule to use to process the internal-format document. You should create a processing rule that uses the Execute a Service processing action to invoke a service that you create to form the EDI document. For more information about creating this service, see “Forming EDI Documents to Send Outbound” on page 21 .

Step	Description
4	<p>The service you create to form the EDI document executes. After forming the EDI document, your service creates a BizDocEnvelope that contains the EDI document and places it in the pipeline in the <i>bizdoc</i> variable (overwriting the previous <i>bizdoc</i> variable).</p> <p>Your service then invokes the <code>wm.tn.route:routeBizDoc</code> service to send the BizDocEnvelope that contains the EDI document back into Trading Networks to select a different processing rule.</p>
5	<p>Trading Networks searches its processing rules again to find the appropriate rule to use to deliver the EDI document. You need to create a processing rule for the EDI document that does one of the following:</p> <ul style="list-style-type: none">■ Uses the Execute a Service processing action that invokes a service that you created to deliver the EDI document. <p>-OR-</p> <ul style="list-style-type: none">■ Uses the Deliver Document By processing action to:<ul style="list-style-type: none">■ Send the EDI document to a VAN. For more information, see “Outbound Processing: Delivering Documents to VANs” below.■ Batch the EDI document for delivery. For more information, see “Batching Outbound EDI Documents” on page 51.

For more information about the following subjects, see Chapter 16, "Forming EDI Documents to Send Outbound When Using Trading Networks", in the *webMethods EDI Module User's Guide*:

- The logic to use to route an outbound EDI document through Trading Networks
- How to set up Trading Networks to deliver the outbound EDI document

Working with VANs

Many EDI-based business partners use Value Added Networks (VANs) as their primary EDI document exchange engine. You can use the EDI Module with Trading Networks to connect with these VANs. The EDI Module provides built-in services that enable VAN connectivity, which allows you to access VANs to retrieve EDI documents from the VAN, deliver EDI documents to the VAN, and obtain reports about EDI documents.

webMethods has tested and certified the EDI Module to connect with the GXS, ICC.NET, and MCI VANs. If you need to connect to another VAN, you might need to customize the provided service to suit the specific VAN connectivity.



Note: The GXS VAN was formerly known as the GEIS VAN.

The EDI Module supports PGP (Pretty Good Privacy). The EDI Module optionally can PGP-encrypt and sign documents bound for VANs that support version 3 PGP-encryption, as well as decrypt and verify PGP-encrypted documents from VANs. The ICC.NET supports PGP encryption. For general information about PGP, see <http://www.pgp.net/pgpnet/pgp-faq/>.



Note: PGP-encryption support is deprecated; it will not be available in a future release of the EDI Module.

PGP-encryption is supported *only* as part of the VAN connectivity to ICC.net. PGP-encryption is *not* generically supported across the webMethods components.

Inbound Processing: Retrieving Documents from VANs

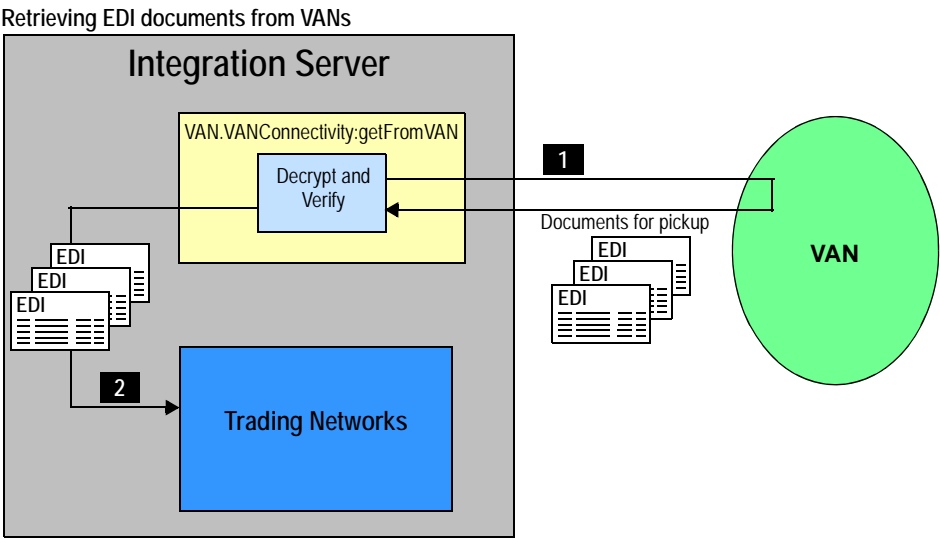
The EDI Module provides the `VAN.VANConnectivity:getFromVAN` service to retrieve documents waiting on a VAN. You can use the Server Administrator to create a scheduled task to have the Integration Server invoke this service at times you schedule.

You can set the inputs to the service to indicate other optional actions you want the `VAN.VANConnectivity:getFromVAN` service to take in addition to picking up the waiting documents. The optional actions you can request are:

- Decrypt and verify PGP-encrypted documents from VANs.
- Submit the picked up EDI documents to Trading Networks to have the EDI Module process the EDI document.
- Retrieve VAN-generated reports while connected to the VAN. The reports that are available depend on the VAN and can change at any time. Contact VANs directly for timely and accurate information.



Note: You can also pick up documents from a VAN when you deliver documents to a VAN. This is described below in [“Outbound Processing: Delivering Documents to VANs” on page 48](#).



Step	Description
1	Invoke the <code>VAN.VANConnectivity:getFromVAN</code> service. The service connects to the VAN and returns the EDI documents waiting for pickup.
2	Optionally, you can have the <code>VAN.VANConnectivity:getFromVAN</code> service submit the the EDI documents it picks up to Trading Networks. The EDI Module processes the returned EDI documents like it would any inbound EDI documents. For more information, see “Processing Inbound EDI Documents with Trading Networks” on page 25.
If you do not submit the picked up documents to Trading Networks, you can create your own logic to process the EDI documents returned from the VAN.	

For more information about setting up to retrieve inbound EDI documents from a VAN, see Chapter 18, "Retrieving and Delivering EDI Documents from and to VANs", in the *webMethods EDI Module User's Guide*.

Outbound Processing: Delivering Documents to VANs

The EDI Module provides the `VAN.VANConnectivity:putToVAN` service to deliver outbound EDI documents to a VAN. When you install the EDI Module, this service is registered as a Trading Networks registered delivery service and assigned the name `VANFTP` in Trading Networks.

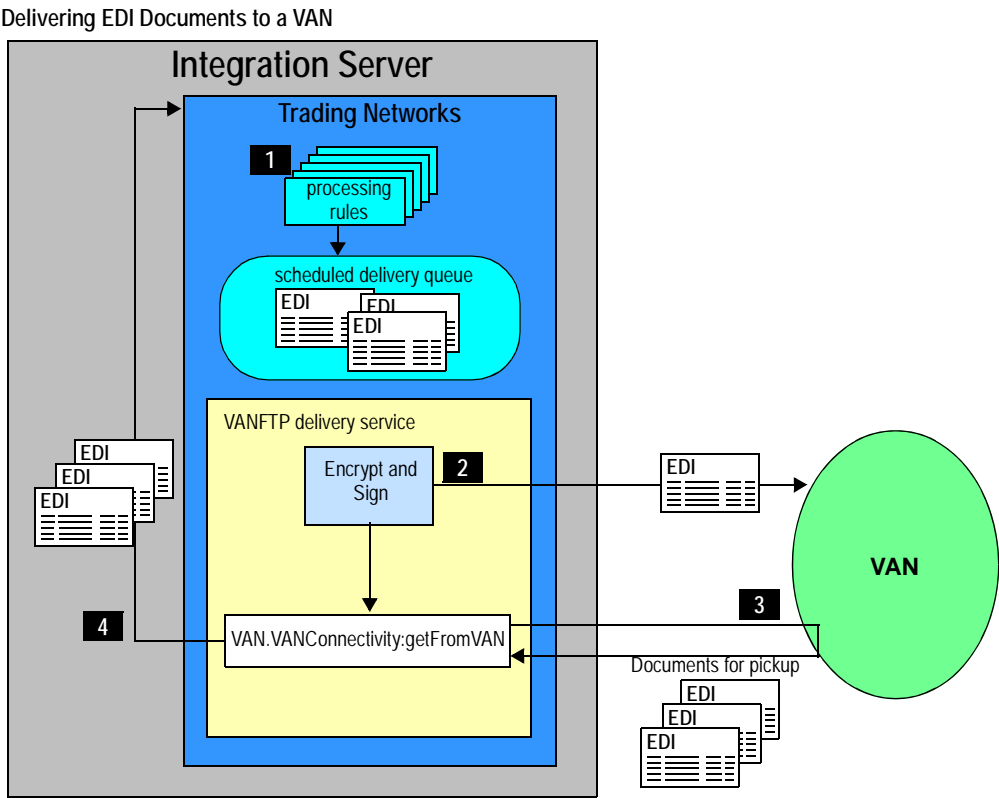
To use this service, you define a public scheduled delivery queue in Trading Networks. When you define the queue, you associate the queue with the `VANFTP` delivery service and specify a schedule for when Trading Networks is to deliver the documents in the queue. You can define as many queues as you need that use the `VANFTP` delivery service. For

example, you might define two queues if you want to deliver documents to two different VANs.

To get EDI documents into the queue, you define a Trading Networks processing rule that uses the **Deliver Document By** processing action. When you set the **Deliver Document By** processing action, you use **Scheduled Delivery** and identify the name of the queue that uses the VANFTP delivery service. When the schedule that you associated with the queue indicates, Trading Networks invokes the VANFTP delivery service to deliver the documents that are in the queue to the VAN.

When you define the public scheduled delivery queue, you can set the inputs to the service to indicate other optional actions you want the VANFTP delivery service to take in addition to delivering the outbound EDI documents. The optional actions you can request are:

- PGP-encrypt and sign the documents before sending them to the VAN.
- Retrieve VAN-generated reports while connected to the VAN. The reports that are available depend on the VAN and can change at any time. Contact VANs directly for timely and accurate information.
- Retrieve any inbound documents that are waiting on the VAN. The picked up EDI documents are automatically submitted to Trading Networks to have the EDI Module process the EDI documents. (Note that in this situation when you pick up EDI documents after delivery of EDI documents to a VAN, the EDI Module supports retrieving only non-PGP encrypted documents.)



Step	Description
1	Trading Networks uses its processing rules to determine how to deliver an EDI document. For more information about how to form an EDI document and send it to Trading Networks for delivery, see “Forming EDI Documents and Sending Them Outbound” on page 39. To deliver EDI documents to a VAN, you create a processing rule that uses the Deliver Document By processing action to deliver a document to a scheduled delivery queue associated with the VANFTP delivery service. For more information about scheduled delivery and defining queues for scheduled delivery, see the <i>webMethods Trading Networks User’s Guide</i> .
2	When the schedule that is associated with the queue indicates, Trading Networks invokes the VANFTP delivery service to deliver the EDI documents in the queue to the VAN. Based on input variables that you set for the service, the service can PGP encrypt and sign documents before sending them to the VAN.

Step	Description
3	Optionally, the VANFTP delivery service can invoke the built-in service <code>VAN.VANConnectivity:getFromVAN</code> service, which the EDI Module provides, to retrieve any documents that might be waiting on the VAN for pick up. You specify whether you want documents picked up when you set the input variables of the VANFTP delivery service. Note that if you want to pick up documents that are PGP-encrypted, you should use the <code>VAN.VANConnectivity:getFromVAN</code> service as described in “Inbound Processing: Retrieving Documents from VANs” on page 47.
4	When the VANFTP delivery service invokes the <code>VAN.VANConnectivity:getFromVAN</code> service, it specifies that the EDI documents it picks up should be submitted to Trading Networks. The EDI Module processes the returned EDI documents like it would any inbound EDI documents. For more information, see “Processing Inbound EDI Documents with Trading Networks” on page 25.

For more information about the following subjects, see Chapter 18, "Retrieving and Delivering EDI Documents from and to VANs", in the *webMethods EDI Module User's Guide*:

- How to define the scheduled delivery queue for outbound EDI documents that are to be sent to a VAN
- How to define processing rules that place outbound EDI documents into a queue

Batching Outbound EDI Documents

EDI systems typically and historically work on batch documents. You can use the EDI Module with Trading Networks to batch EDI documents for delivery rather than delivering EDI documents to the systems in real time as the documents are received.

Batching offers a more flexible and affordable approach to EDI document exchange and provides the following benefits:

- Enables documents to be grouped and sent at scheduled times that are more appropriate to organizational requirements.
- Increases system performance, requiring fewer communication connections and less time spent on authenticating envelopes that are sent individually.
- Makes working with legacy systems easier because legacy systems are batch-oriented.

To batch the documents, the EDI Module provides the `wm.b2b.edi tn.batch:batchProcess` service. When you install the EDI Module, this service is registered as a Trading Networks delivery service and assigned the name EDI Batch in Trading Networks. The `batchProcess` service combines EDI documents into a single document and adds group-level and interchange-level headers and trailers to the document.

To use the batchProcess service, you define public scheduled delivery queues in Trading Networks. When you define a queue, you associate the queue with the batchProcess service and specify a schedule for when Trading Networks is to invoke the batchProcess service to act on the documents in the queue.

To get the EDI documents into the queue so that they can be batched into a batch EDI document, you define a Trading Networks processing rule that uses the **Deliver Document By** processing action. When you set the **Deliver Document By** processing action, you use **Scheduled Delivery** and identify the name of the queue that uses the batchProcess delivery service. When the schedule that you associated with the queue indicates, Trading Networks invokes the batchProcess service to combine the EDI documents in the queue into a batch EDI document.

The batchProcess service uses its input variables and information in EDITPAs when creating the combined EDI document. You can set the input variables and EDITPA variables to:

- Control the creation of the UNA segment for UN/EDIFACT interchanges (or the BAT segment for TRADACOMS transmissions) for the batch process.
- Set default interchange or TRADACOMS transmission header values.

When batching EDI documents that were placed in the queue, the EDI Module can batch the documents into one of the following:

- A single output batch EDI document with multiple interchange or TRADACOMS transmission envelopes.
- Multiple output batch EDI documents, each containing only a single interchange or TRADACOMS transmission envelope.

For more information about how the EDI Module creates the output batch EDI document, see [“Creating the Batched EDI Documents” on page 53](#).

For more information about how to set up EDI batching, see Chapter 17, “Batching EDI Documents”, in the *webMethods EDI Module User’s Guide*.



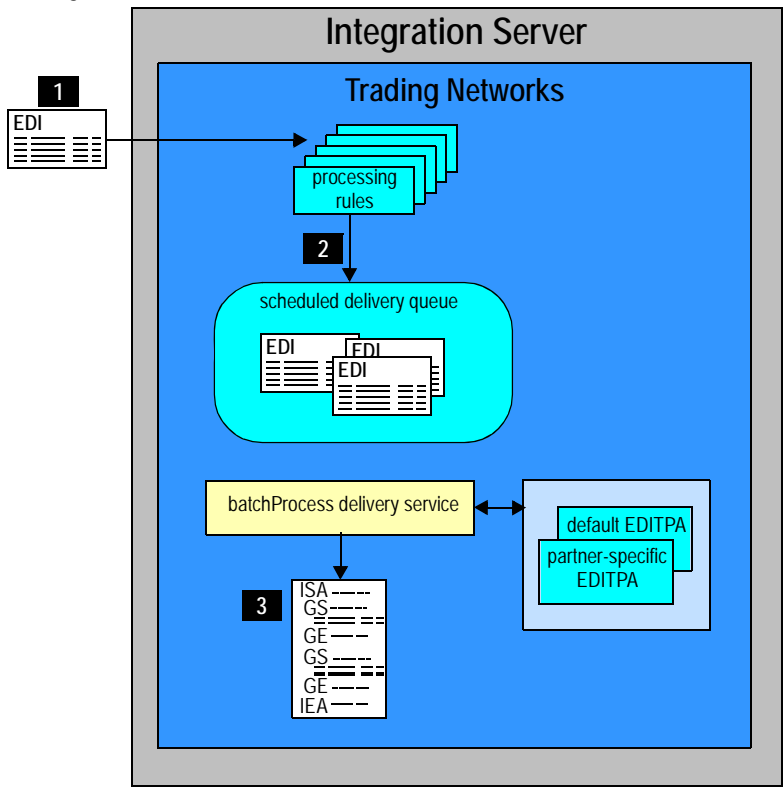
Note: This section describes the batching feature of version 6.1 and greater of the EDI Module. For information about batching in previous releases, see Appendix B, “Using the 6.0.1 Version of the Batching Feature”, in the *webMethods EDI Module User’s Guide*.

Creating the Batched EDI Documents

To set up to form batched EDI documents, you define scheduled delivery queues and processing rules in Trading Networks.

The following diagram illustrates how to form a batched EDI document. For more information, see the table after the diagram.

Forming the batched EDI document



Step	Description
1	EDI document is sent to Trading Networks. For more information about how to form an EDI document and send it to Trading Networks for delivery, see “Forming EDI Documents and Sending Them Outbound” on page 39 .
2	Trading Networks uses its processing rules to determine how to process the EDI document. Trading Networks selects a processing rule that you create that uses the Deliver Document By processing action to deliver a document to a scheduled delivery queue associated with the batchProcess service.
For more information about scheduled delivery and defining queues for scheduled delivery, see the <i>webMethods Trading Networks User’s Guide</i> .	

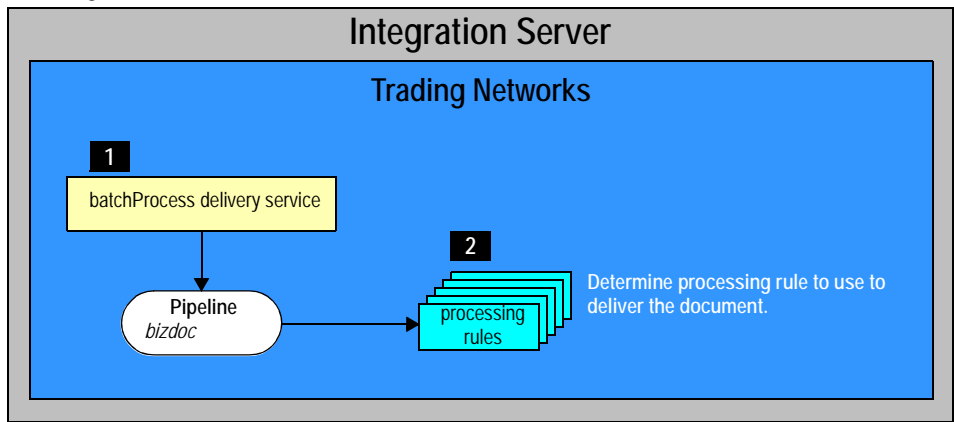
Step	Description
3	<p>When the schedule that is associated with the queue indicates, Trading Networks invokes the batchProcess service to combine the EDI documents in the queue into the output batch EDI document(s).</p> <p>The final EDI document is ready for delivery. For more information about how the EDI Module processes the document so it can be delivered, see “Delivering the Batched Document” on page 54.</p>

- For more information about the following subjects, see Chapter 17, "Batching EDI Documents", in the *webMethods EDI Module User's Guide*:
- How the batchProcess delivery service combines documents in the queue to form the batch EDI document
 - Variables that you specify to control how the batchProcess delivery service forms the batched EDI document
 - How to define the scheduled delivery queues and processing rules to use for batch processing

Delivering the Batched Document

After the batchProcess service forms the final batched EDI document, it routes the document back to Trading Networks processing rules for delivery. The following diagram illustrates the process for delivering a batched EDI document. For more information, see the table after the diagram.

Delivering a batch EDI document



Step	Description
1	The batchProcess service forms the final outbound EDI document as described in “Creating the Batched EDI Documents” on page 53 . The batchProcess service then creates a BizDocEnvelope for the final outbound EDI document and places it in the pipeline in the <i>bizdoc</i> variable. It then routes the BizDocEnvelope to the Trading Networks processing rules.
2	After forming the BizDocEnvelope, Trading Networks determines the processing rule to use to deliver the outbound batch EDI document. You create the processing rule to define how you want to deliver the document. For example, you can invoke a service that you create to deliver the batch EDI document, or you can deliver the batch EDI document to a VAN as described in “Outbound Processing: Delivering Documents to VANs” on page 48 .

For more information about defining processing rules to deliver outbound batch EDI documents, see Chapter 17, "Batching EDI Documents", in the *webMethods EDI Module User's Guide*:

EDI Documents in Business Processes

■ Overview	58
■ How You Define the Business Process	58
■ Conversation IDs for EDI Documents	59
■ How EDI Documents are Passed to a Business Process	60

Overview

As an alternative to using processing rule actions, or in addition to using processing rule actions, you can define a *business process* (also called a *conversation*) that describes the steps required to process ANSI X12 or UN/EDIFACT EDI documents. In the business process you can include steps:

- To send or wait for acknowledgements or response documents
- That require human interaction, which you can implement using webMethods Workflow

To use a business process for EDI documents, you must use Trading Networks.



Note: You cannot process TRADACOMS documents in a business process.

How You Define the Business Process

You define the actions that take place in a business process by using webMethods Modeler to design a process model. A *process model* is a diagram that shows the steps in the business process. You set properties for each step to further define information about the actions to take for each step. For example, you can set properties that identify a service to invoke for a step.

webMethods Modeler is a design-time tool only. Before the process model can be executed, you must create run-time elements for a process model. This is called *generating* the process model. When you generate the process model, webMethods Modeler generates triggers, flow services, etc. based on the steps in your process model and saves these run-time elements in the Integration Server namespace.

At run time the *process run time*, which is a facility of the Integration Server, manages the execution of business processes. The process run time executes the business process by using the appropriate run-time elements that were generated from a process model.

Typically, a business process starts based on the arrival of a document. It is the responsibility of the process run time to determine the actions to take for a specific document. The process run time determines the process model to use for the document and defines a new instance of the process model to govern the actions to take for the business process. When a subsequent document for the business process arrives, it is the process run time that determines the correct running instance of a process model and rejoins the business process by passing it the document that just arrived.

The way the process run time determines the documents that belong to a single instance of a business process is through the conversation ID. All documents in the same instance of a business process share an identifier called a *conversation ID*. So when the process run time receives a document, it determines whether it has a running business process that uses the conversation ID. If it does, the process run time passes the document to the

running business process to rejoin the running business process. If there is no running business processes that uses that conversation ID, the process run time searches for a process model that has the first step that waits for the document, and if found, starts a new instance of the process model.

As the process run time manages the execution of a business process, it logs its progress and status to the process audit log database. You can view the progress and status using webMethods Monitor.

For more information about:

- How to create process models, see the *webMethods Modeler User's Guide*.
- How to design the process models, view the sample process models provided in the *webMethods6\IntegrationServer\packages\WmEDIforTN\config\Models* directory. You can import them into webMethods Modeler to view them.
- How to monitor running business processes, see the webMethods Monitor documentation.

Conversation IDs for EDI Documents

Before an EDI document can be used in a business process, it must have a conversation ID.

The original EDI document is split into Transaction, Group, and Interchange documents as described in [“Processing Inbound EDI Documents with Trading Networks” on page 25](#). To process these documents using a business process, the document (Transaction, Group, and Interchange) must have a conversation ID.



Note: You are not required to process all three types of documents using a business process. Use the documents that work best to create your solution.

- **For a Transaction document**, you must provide the EDI Module with information that it uses to form the conversation ID. The information that you provide is an *instance ID query*. The instance ID query is a query that the EDI Module can perform against the Transaction document to retrieve a value for the conversation ID. The instance ID query is specific to the type of transaction set contained in the Transaction document. For example, you might define the instance ID query *ST/BEG/BEG03* for an X12 4010 850 transaction set. Whenever the EDI recognizer creates a Transaction document that contains an X12 4010 850 transaction set, it will use the instance ID query you specify to obtain the value to use for the conversation ID for the

Transaction document. If the EDI recognizer creates a Transaction for which there is no instance ID query, the conversation ID is not set.



Note: If you do not need to process the Transaction document, do not create an instance ID query for the transaction set, and do not create a process model that uses the document.

- For a Group or Interchange document, the EDI Module *always* assigns conversation IDs. No instance ID query is required. For a Group document, the EDI Module sets the value of the conversation ID to the group control number. For an Interchange document, the EDI Module sets the value of the conversation ID to the interchange control number.



Note: The Group and Interchange documents will always have a conversation ID. If you do not need to process either the Group or Interchange document in a business process, do not create a process model that uses the document. If the process run time is unable to locate a matching process model, it does not perform processing for the document.

For more information about how to define instance ID queries to set conversation IDs for Transaction documents, see Chapter 20, "Including Documents in a Business Process", in the *webMethods EDI Module User's Guide*.

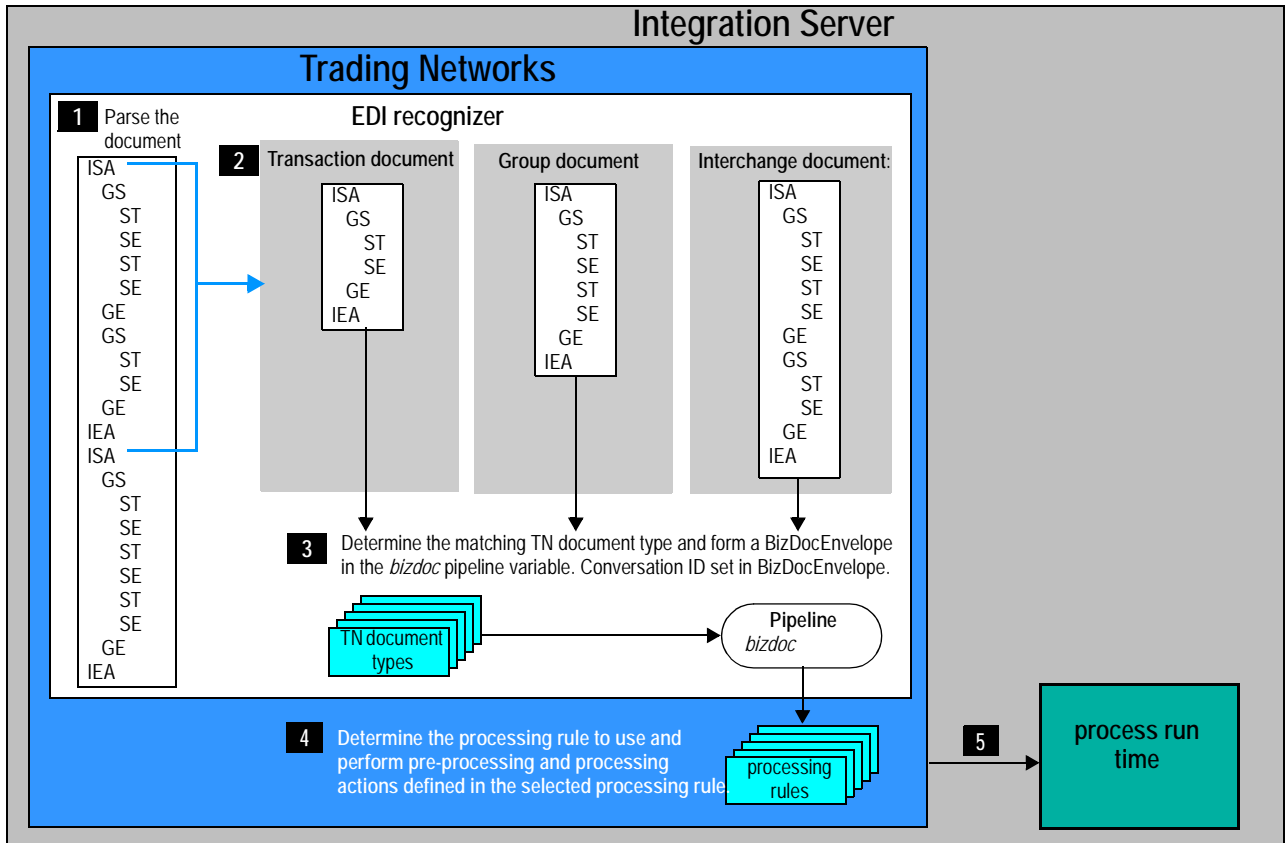
How EDI Documents are Passed to a Business Process

For an EDI document to be used in a business process, the document must be sent to the process run time. The way EDI documents get sent to the process run time is by Trading Networks.

First Trading Networks does its own processing (document recognition and performing the actions defined by a processing rule). Then, if the Trading Networks system attribute ConversationID has a value, Trading Networks passes the document to the process run time.

The following diagram illustrates the processing to have an EDI document sent to the process run time for processing. For more information, see the table after the diagram.

Processing an EDI document using a business process



Step	Description
1	When Trading Networks receives an EDI document, it passes the EDI document to the EDI recognizer. The EDI recognizer parses the inbound EDI document.
2	For each interchange segment, the EDI recognizer retrieves the appropriate EDITPA to determine the split option to use for the EDI document. The EDI recognizer splits the EDI document into the appropriate documents (i.e., Transaction, Group, and Interchange documents). For more information about how EDI documents are split, see “Processing Inbound EDI Documents with Trading Networks” on page 25.

Step	Description
3	The EDI recognizer submits the newly formed documents (Transaction, Group, or Interchange) to Trading Networks for processing. The EDI recognizer uses the TN document types to determine the type of document. After recognizing the type of document using TN document types, the EDI recognizer forms a BizDocEnvelope for the EDI document and sets the ConversationID system attribute in the BizDocEnvelope. For more information about the value used for the Conversation ID, see “Conversation IDs for EDI Documents” on page 59 .
4	After forming the BizDocEnvelope, Trading Networks determines the processing rule to use to process the document and executes the processing rule. When you have a document that you plan to process using a business process, you will typically either 1) set up the processing rule to ignore the document -or- 2) not define a processing rule for the document causing Trading Networks to select the Default processing rule, which ignores the document.
5	Because the ConversationID system attribute contains a value, Trading Networks passes the document to the process run time. The process run time either 1) starts a new business process based on a process model that you have designed -or- 2) determines the running business process that the document is to join.

Glossary

activity log

A log that webMethods Trading Networks (Trading Networks) maintains in its database to record activity that occurs within the Trading Networks system.

Batch document

A document that the EDI Module creates from splitting the transmission of a TRADACOMS EDI document. A batch document contains a single batch along with its files. After creating a Batch document the EDI Module sends the document to Trading Networks for processing.

bizdoc

The name of the variable in the pipeline that contains the *BizDocEnvelope*.

BizDocEnvelope

A BizDocEnvelope represents a routable Trading Networks transaction. It contains a document that Trading Networks is processing and includes additional information that Trading Networks requires for routing and processing the document. It is in the pipeline in the *bizdoc* variable and conforms to the IS document type `wm.tn.rec:BizDocEnvelope`.

business process

A multi-step interaction among participating systems, people, and trading partners. A business process can be fully automated (involve only interaction among computer systems) or include varying degrees of human interaction (for example, review and approval steps). It may be brief or long-running. Some business processes transpire over days or weeks.



Note: TRADACOMS documents cannot be used in business processes.

control number

A number in the header of an EDI document that is used for validation and ordering of documents exchanged between trading partners. Control numbers

can be used to detect duplicate, missing or out of sequence documents. See also [duplicate control number](#) and [out-of-sequence control number](#).



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

control number cap

An EDI Module setting that you can configure to specify the highest number allowed for a control number. The EDI Module maintains the next expected control number for a document in the [EDIControlNumber table](#). To calculate the next expected control number, the EDI Module adds the [control number increment](#) to the current control number. If the result exceeds the control number cap, the EDI Module sets the next expected control number to the [control number minimum](#).

control number increment

An EDI Module setting that you can configure to specify the increment to use when calculating the next expected control number. Typically, the value is one. To calculate the next expected control number, the EDI Module adds the control number increment to the current control number.

control number minimum

An EDI Module setting that you can configure to specify the lowest number that a control number can be. The EDI Module maintains the next expected control number for a document in the [EDIControlNumber table](#). To calculate the next expected control number, the EDI Module adds the [control number increment](#) to the current control number. If the result exceeds the [control number cap](#), the EDI Module sets the next expected control number to the control number minimum.

control number window

An EDI Module setting that you can configure to identify a range of numbers that the EDI Module uses when determining whether an invalid control number is a *duplicate control number* or an *out-of-sequence control number*.

- Typically the EDI Module determines that a control number is a *duplicate control number* if the control number is less than the expected control number. This is not true when the expected control number is close to the *control number minimum* because an already used control number might be larger than the control number. The *control number window* indicates a range of numbers that are greater than the expected control number for which a control number should be considered duplicate rather than a out-of-sequence.

Example

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

- Typically, the EDI Module determines that a control number is out-of-sequence if the control number is greater than the expected control number. This is not true as the control number approaches the *control number cap* because incrementing the *control number cap* causes the control number to return to the *control number minimum* (a lower number). The *control number window* indicates a range of numbers that are lower than the *control number cap* for which a control number should be considered out-of-sequence rather than a duplicate.

Example

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

conversation

A specific case of a [business process](#) that involves a series of related documents being exchanged by two or more trading partners. All documents from a specific trading partner contain the same conversation ID. You model a conversation by creating a [process model](#) using webMethods Modeler.



Note: Conversations are not applicable to TRADACOMS documents.

conversation ID

A webMethods Trading Networks' system document attribute that identifies a value within a document that is common to all documents that are part of the same [business process](#) (also called a [conversation](#)).



Note: Conversations IDs are not applicable to TRADACOMS documents.

delivery task

Information that Trading Networks maintains for a document that is scheduled for delivery. For example, when you use a scheduled delivery queue to schedule an EDI document for delivery to a VAN, Trading Networks creates a delivery task for the EDI document.

document ID

A Trading Networks' system attribute that identifies a document. This is typically a unique value that distinguishes a document from other versions of the same document. The webMethods EDI Module forms the document ID using EDI control numbers.

document type

See [IS document type](#) or [TN document type](#).

duplicate control number

Generally a [control number](#) is considered a duplicate if it is less than the expected control number. For example, you received an EDI document that has control number 6, but the expected control number is 8. See [control number window](#) for information about determining a duplicate control number when the expected control number is close to the [control number minimum](#).

EDIControlNumber table

An EDI Module-specific table in the Trading Networks database that the EDI Module uses to maintain information about control numbers.

EDI ID qualifier

A code within a header of an EDI document that indicates the type of value that is used to identify the sender for receiver. For example, if the value for a sender is a D-U-N-S number, the EDI ID qualifier is the code '01' that indicates a D-U-N-S number. For all EDI ID qualifiers, see the EDI Standards documentation for your EDI standard and version. Note that when using the EDI Module with Trading Networks, the EDI ID qualifier corresponds to a Trading Networks [external ID type](#).

EDITPA

A *Trading Partner Agreement (TPA)* that contains EDI Module-specific settings. You can have partner-specific EDITPAs to define specific settings for a pair of trading partners. There is also a default EDITPA that the EDI Module uses when a partner-specific EDITPA does not exist or when a value in the partner-specific EDITPA is not specified.

EDITTRACKING table

An EDI Module-specific table in the Trading Networks database that the EDI Module uses to maintain information about EDI documents and their corresponding functional acknowledgements. The EDI Module uses the information in the table for FA reconciliation.

extended fields

Fields within a profile that are not provided with Trading Networks out-of-the-box. Users of Trading Networks can define extended profile fields to extend the profile to meet their needs.

external ID

The value of an *external ID type*. For example, if an external ID type is a D-U-N-S number, the external ID is the actual D-U-N-S number.

external ID type

A type of identifier that trading partners use in documents. For example, a user might use a D-U-N-S number. The external ID type corresponds to the EDI ID qualifier in an EDI document.

File document

A document that the EDI Module creates from splitting the transmission of a TRADACOMS EDI document. A file document contains multiple messages. After creating a File document the EDI Module sends the document to Trading Networks for processing.

file polling

A facility of the Integration Server where you define a File Polling listening port. When you define the File Polling listening port, you define a directory and associate a content type and service with the directory. The Integration Server monitors the directory that you define. When files arrive in the directory, the Integration Server invokes the service that you associate with the directory and passes the file to the service with the content type that you associate with the directory. For more information about File Polling, see information about configuring ports in the *webMethods Integration Server Administrator's Guide*. You can create clients for EDI documents that use File Polling.

flat file dictionary

A collection of record definitions, field definitions, and composite definitions that can be used in multiple flat file schemas.

flat file schema

The blueprint that contains the instructions for parsing or creating a flat file. This blueprint details the structure of the document, including delimiters, records, and loops.

A flat file schema also acts as the model against which you validate an inbound flat file. Flat file schemas are elements in the Integration Server's namespace.

functional acknowledgement (FA)

A type of EDI transaction set that acknowledges the receipt as well as the structural and syntactical validity of an EDI document.

Group document

A document that the EDI Module creates from splitting the interchange of an ANSI X12 or UN/EDIFACT EDI document. A group document contains a single group along with its transaction sets. After creating a Group document the EDI Module sends the document to Trading Networks for processing.

IData object

The collection of name/value pairs on which a service operates. An IData object can contain any number of elements of any valid Java objects, including additional IData objects and IDataCodable objects.

instance ID query

A query that the EDI Module performs against a Transaction document to retrieve a value for a conversation ID. An instance ID query is specific to the type of transaction set contained in the Transaction document. For example, you might define the instance ID query `ST/BEG/BEG03` for an X12 4010 850 transaction set.

Interchange document

A document that the EDI Module creates from splitting the interchange of an ANSI X12 or UN/EDIFACT EDI document. An interchange document contains a single interchange along with its groups and transaction sets. After creating an Interchange document the EDI Module sends the document to Trading Networks for processing.

internal-format document

A document is in a format that a back-end system or internal application uses (as opposed to a standard EDI format).

IS document type

An element in the Integration Server's namespace that contains a set of fields used to define the structure and type of data in an IS document (IData object).

out-of-sequence control number

A [control number](#) that indicates that there might be missing EDI documents. That is, EDI documents that you should have already received have not yet arrived. Generally a control number is considered out-of-sequence if it is greater than the expected control number. For example, you received an EDI document that has 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. See [control number window](#) for information about determining an out-of-sequence control number when the expected control number is approaching to the [control number cap](#).

pipeline

The general term used to refer to the data structure in which input and output values are maintained for a flow service at run time. The pipeline starts with the input to the flow

service and collects inputs and outputs from subsequent services in the flow. When a service in the flow executes, it has access to all data in the pipeline.

processing rule

A webMethods Trading Networks object that contains set of actions that determine how Trading Networks is to process a document and criteria that indicates when to select a processing rule for an incoming document.webMethods Modeler.

process model

Diagrams that illustrate and define the actions to perform for a *business process* or *conversation*. You create process models using webMethods Modeler.

process run time

A facility of the Integration Server that manages the execution of processes (or *conversations*). You model a process (or conversation) using webMethods Modeler.

profile

A webMethods Trading Networks object that contains a summary of information about a corporation that is part of a trading network. A profile contains standard fields that webMethods defines and extended fields that are site defined.

receiver ID

The value in the header of an EDI document that identifies the receiver of the document, for example, a D-U-N-S number.

receiver qualifier

The *EDI ID qualifier* associated with a *receiver ID*.

SEF file

A file in *Standard Exchange Format*. A SEF file contains all of the metadata for the EDI standard. In other words, all of the transaction sets and segments for a particular version of EDI standard.

sender ID

The value in the header of an EDI document that identifies the sender of the document, for example, a D-U-N-S number.

sender qualifier

The *EDI ID qualifier* associated with a *sender ID*.

Standard Exchange Format

A guideline standard placed in public domain by Foresight Corporation (www.foresightcorp.com).

TN document type

A webMethods Trading Networks object that defines how Trading Networks is to recognize a document and specifies initial actions to take on a recognized document.

Trading Partner Agreement (TPA)

A webMethods Trading Networks object that you can use to tailor how documents are exchanged between two trading partners.

Transaction document

A document that the EDI Module creates from splitting the interchange of an ANSI X12 or UN/EDIFACT EDI document. A transaction document contains a single transaction set. After creating a Transaction document the EDI Module sends the document to Trading Networks for processing.

Transmission document

A document that the EDI Module creates from splitting the transmission of a TRADACOMS EDI document. A transmission document contains a single transmission along with its batch and files. After creating a Transmission document the EDI Module sends the document to Trading Networks for processing.

tspace

Local hard disk drive space that you define and that the EDI Module uses to temporarily store documents that are considered large rather than keeping the documents in memory. You define the location of tspace using the *watt.server.tspace.location* property in the *webMethods6\IntegrationServer\config\server.cnf* file.

Value Added Network (VAN)

A private network provider that facilitates EDI by providing support for functions, such as, moving data from one company to another, EDI translation, encryption, secure e-mail, and management reporting.

Index

A

- acknowledgement, FA
 - automatically generating 31
 - defined 19, 68
- acknowledgements, business process 58
- actions
 - pre-processing, *See* pre-processing actions
 - processing, *See* processing actions
 - VAN.VANConnectivity:getFromVAN service
 - inbound 47
 - outbound 49
- activity log, defined 63
- ANSI X12 document format 14
- architecture of webMethods EDI Module 10

B

- back-end system
 - internal-format document from 39
 - sending internal-format document to IS 41, 42, 45
- BAT segment for TRADACOMS transmissions 52
- batch documents 54
 - BAT segment for TRADACOMS transmissions 52
 - BizDocEnvelope for final batch document 55
 - defined 29, 63
 - Deliver Document By processing action to deliver 44, 46
 - diagram of how batch document formed 53
 - EDI Batch delivery service in Trading Networks 51
 - outbound EDI document 39, 40, 51
 - processing rules 53, 55
 - UNA segment for UN/EDIFACT interchanges 52
 - wm.b2b.editn.batch:batchProcess service 51
- Batch split option
 - defined 32
- batchProcess service 51
 - See also* wm.b2b.editn.batch:batchProcess service
 - combine EDI documents in queue into single EDI document 54
 - create BizDocEnvelope for final batch document 55

- Deliver Document By processing action 53
 - routes batched document back to Trading Networks 54
 - scheduled delivery queue 53
- bizdoc variable 41, 43, 45
 - BizDocEnvelope, *See* BizDocEnvelope
 - defined 35, 63
 - overwriting previous bizdoc variable 46
- BizDocEnvelope 35
 - Conversation ID system attribute 62
 - defined 63
 - formed by
 - EDI recognizer 62
 - in bizdoc pipeline variable, *See* bizdoc variable 35
 - internal-format document content 41, 43, 45
 - outbound
 - batch EDI document 55, 62
 - EDI document content 43, 46
 - wm.tn.route:routeBizDoc service 46
- built-in services
 - add interchange and group envelopes to EDI document 22
 - as building block to create service to from EDI document 21
 - diagram of for inbound processing 16
 - EDI Module-provided 16
 - to enable VAN connectivity 46
 - to generate a functional acknowledgement (FA) 19
 - validation
 - EDI document 22
 - internal document 22
 - VAN.VANConnectivity:getFromVAN service 51
 - See also* services
- business process 11
 - See also* conversations
 - Conversation ID system attribute 62
 - defined 58, 63
 - processing rules 62
 - types of EDI documents 59

C

- Check for Duplicate Document pre-processing action 36
- clients
 - EDI 25
 - for inbound processing 15
 - with Trading Networks 25
 - sending internal format document to IS 41, 42, 45
 - to send EDI documents 15
 - with Trading Networks 25
- combine EDI documents, batch 52
- connecting to VANs 46
- control number cap, defined 64
- control number increment, define 64
- control number minimum, defined 64
- control number window, defined 65
- control numbers
 - defined 63
 - duplicate, defined 66
 - out-of-sequence, defined 68
 - validation of 31
- conventions used in this document 5
- conversation
 - See also*, business model
 - defined 58, 66
 - EDI documents 11
 - ID, *See* conversationID
- conversation ID
 - defined 66
 - EDI documents 59
 - Group document 60
 - in business process 58
 - Interchange document 60
 - Transaction document 59
- ConversationID system attribute
 - and process run time 62
 - in business process 62
 - set by EDI recognizer 62
 - Trading Networks system attribute 60
- convert
 - EDI document from IData object to String or InputStream 22
 - EDI documents using flat file schemas 18

D

- database, *See* Trading Networks database
- decrypting PGP-encrypted documents from VANs 47
- default
 - EDITPA 38
 - interchange header values 52
 - processing rule 62
- Deliver Document By processing action
 - batch document 44, 46
 - in queue into single EDI document 52
 - deliver document to
 - queue using VANFTP delivery service 49
 - scheduled delivery queue 53
 - outbound document to VAN 44, 46, 50
- delivering
 - add logic to service to deliver document 39
 - batch outbound EDI documents 39, 40, 51, 54
 - documents to a VAN 50
 - EDI document directly from service that forms it 40
 - outbound EDI document
 - in queue to VAN 50
 - routing to TN processing rules 40, 44
 - routing to TN to deliver 44
 - submitting to TN to deliver 42
 - to a VAN 39, 40
 - using Trading Networks 39
- delivering final batch document 54
- delivery task, defined 66
- diagram of
 - Client sends EDI document to Integration Server 15
 - Client sends EDI document to Trading Networks 26
 - Delivering documents to a VAN 50
 - Delivering EDI Documents
 - batch documents 54
 - directly from the service that forms it 41
 - to a VAN 50
 - EDI Module
 - and Integration Server architecture 10
 - and the webMethods Architecture 10
 - Forming the batched EDI document 53
 - Further processing of Transaction, Group, and Interchange documents 34
 - Generating Functional Acknowledgements (FAs) 20

- Inbound EDI document processing 14
 - when using Trading Networks 25
- Outbound EDI document processing 21
- Processing EDI document using a business process 61
- Recognizing the original EDI document and beginning processing 28, 29
- Retrieving EDI documents from VANs 48
- Service for
 - forming outbound EDI documents 21
 - processing inbound EDI documents 16
- Submitting the outbound EDI document to Trading Networks to deliver it 42
- document ID, defined 66
- document recognition
 - submit outbound EDI document to 39
 - Trading Networks and EDI outbound documents 42
 - wm.tn.doc.sml:routeXML service 43
- document types
 - See* TN document types
 - See also* IS document types
- documentation
 - additional 5
 - conventions used 5
 - feedback 5
- documents
 - batch, *See* batch documents
 - built-in service to validate structure 22
 - business process 58
 - decrypting PGP-encrypted from VANs 47
 - delivering to a VAN 50
 - file, *See* File documents
 - group, *See* Group documents
 - interchange, *See* Interchange documents
 - map internal to EDI 22
 - PGP-encrypted 47, 49, 50
 - retrieving documents
 - waiting on a VAN 47, 49, 51
 - while connected to a VAN 49
 - sign PGP-encrypted to send to VAN 49, 50
 - transaction, *See* Transaction documents
 - transmission, *See* Transmission documents
 - verifying PGP-encrypted from VANs 47
- duplicate control number, definition 66
- E**
 - EANCOM document format 14
 - EDI Batch delivery service 51
 - EDI client, *See* clients
 - EDI content
 - handler 15, 25
 - type 15, 25
 - EDI documents
 - See also* outbound EDI documents
 - batch, *See* batch documents
 - BizDocEnvelope 35
 - built-in service to validate structure 22
 - business processes 11
 - Check for Duplicate Document pre-processing action 36
 - combine by batching 52
 - conversation ID 59
 - conversations 11
 - convert from IData object to String or InputStream 22
 - delivering from service that forms it 40
 - exchange engine, VANs 46
 - flat file schema 18
 - forming outbound 39
 - in business process, types of 59
 - inbound processing 14
 - with Trading Networks 25
 - map data from internal documents to 22
 - methods to deliver 39
 - outbound processing 21
 - with Trading Networks 39
 - picked up and submitted to Trading Networks 47
 - pre-processing actions 36
 - processing actions 37
 - processing rules 24, 36
 - to form outbound EDI document 39
 - profiles 24
 - public
 - queues 24
 - scheduled queues 49
 - routing to Trading Networks to deliver 44
 - Save Document to Database pre-processing action 36
 - saving original to Trading Networks database 32
 - sent to process run time 60
 - services for inbound processing 16

- standard format 14
 - submitting to Trading Networks to deliver 42
 - TN document types 24
 - trading partner agreements (TPAs) 24
 - Validate Structure pre-processing action 36
 - VANs 46
 - EDI ID qualifier, defined 66
 - EDI Module
 - architecture of 10
 - batch EDI documents for delivery 51
 - functionality 8
 - overview 8, 24
 - packages of 10
 - samples package 10
 - tools 8
 - Trading Networks objects 24
 - using without other webMethods components 14
 - with Integration Server 8
 - with Trading Networks 9
 - with webMethods Monitor 9
 - EDI recognizer 28
 - BizDocEnvelope 35
 - diagram of initial processing 28, 29
 - group envelope 34
 - interchange
 - envelope 34
 - sender/receiver pair information 30
 - parsing EDI document 61
 - processing
 - after split option 34
 - non-standard 30
 - standard 30
 - resubmitting document to Trading Networks 35
 - saving EDI document to Trading Networks database 32
 - setting Conversation ID system attribute 62
 - split option variable within EDITPA 32
 - splitting EDI document 43, 61
 - submitting EDI document 62
 - TN document types 35, 62
 - EDI Trading Partner Agreement (EDITPA), *See* EDITPA
 - EDIControlNumber table, defined 66
 - edidata variable 15, 25
 - EDIINT AS1 (SMTP) 11, 25
 - EDIINT AS2 (HTTP or HTTPS) 25
 - EDIINT AS2 (HTTP) 11
 - EDIINT AS3 (FTPS (FTP over SSL)) 25
 - EDIINT AS3 (FTPS) 11
 - EDITPA
 - batch EDI documents 52
 - batchProcess delivery service 52
 - default
 - defined 38
 - setting to save original EDI document to TN database 32
 - defined 38, 67
 - EDI recognizer 29
 - partner-specific, defined 38
 - split option variable 32
 - EDITRACKING table, defined 67
 - encryption, PGP documents to send to VANs 49, 50
 - envelope, validation 29
 - Execute a Service processing action 37, 39
 - Group split option 37
 - Interchange split option 38
 - internal format document to outbound EDI document 41, 43, 45
 - invoke service to deliver EDI document 44
 - logic to add for
 - Group split option 37
 - Interchange split option 38
 - Transaction split option 37
 - to deliver outbound EDI document 46
 - Transaction split option 37
 - extended profile fields, defined 67
 - external ID
 - defined 67
 - type, defined 67
- ## F
- FA (functional acknowledgement)
 - automatically generating 31
 - defined 19, 68
 - FA reconciliation
 - and Trading Networks 9
 - Group split option 43
 - File documents, defined 29
 - File Polling transport 15, 25

file polling, defined 67
 File split option, defined 32
 flat file dictionary, defined 67
 flat file schema
 built-in service to validate against 22
 defined 18, 67
 for EDI documents 18
 for internal-format (back-end system) documents 19
 FTP transport 15, 25
 functional acknowledgement (FA)
 automatically generating 31
 defined 19, 68
 diagram of generating FAs 20

G

GEIS VANs 47
 generating the process model 58
 group control number 60
 Group documents
 conversation ID 60
 create processing rule for
 Group split option 37
 Transaction split option 37
 defined 29, 68
 diagram of processing 34
 group control number 60
 interchange control number 60
 processing
 after Group split option 35
 after Transaction split option 35
 non-standard 30
 standard 30
 group envelope
 built-in service to add 22
 Transaction document 34
 group headers and trailers, and batchProcess service 51
 Group split option
 defined 32
 Execute a Service processing action 37
 FA reconciliation and outbound EDI document delivery 43
 for EDI outbound documents submitted back to TN document
 recognition 43
 processing afterwards 35
 grouping documents, batch documents 51
 GXS VANs 47

H

headers
 access of in Trading Networks database 39
 default interchange values 52
 HTTP transport 15, 25
 HTTPS transport 15, 25

I

ICC.NET VANs 47
 IData object
 and flat file schema 18
 built-in service to convert outbound documents 22
 built-in service to validate structure 22
 defined 68
 inbound processing
 clients to send EDI documents 15
 with Trading Networks 25
 EDI documents 14
 with Trading Networks 25
 EDI recognizer 28
 Execute a Service action 37
 retrieving documents from VANs 47
 inputs
 batchProcess service 52
 VANFTP delivery service 49, 50, 51
 InputStream
 built-in service to convert to IData object 22
 converting to IData object 18
 installing
 EDI Module packages 10
 TN document types for EDI documents 24, 35
 VANFTP devliery service 48
 wm.btb.editn.batch:batchProcess service 51
 WmEDIforTN package 28
 instance ID query, defined 59, 68
 Integration Server
 and EDI Module architecture 10
 and EDI Module, diagram 10
 defined 10
 with EDI Module 8
 interchange control number 60

Interchange documents

- conversation ID 60
- create processing rule for
 - Group split option 37
 - Interchange split option 38
 - Transaction split option 37
- defined 29, 68
- diagram of processing 34
- processing
 - after Group split option 35
 - after Interchange split option 35
 - after Transaction split option 35
 - standard 30

interchange envelope

- built-in service to add 22
- Transaction document 34
- validation 29

interchange sender/receiver pair information 30

Interchange split option

- defined 32
- Execute a Service processing action 38
- for EDI outbound document submitted back to TN document recognition 43
- processing afterwards 35

internal-format document

- defined 39, 68
- from back-end system 39, 41, 42, 45
- processing rules 41, 43, 45
- TN document types 41, 43, 45
- to form outbound EDI document 39, 40, 42, 44

invoking

- batchProcess service 52, 54
- built-in service to generate an FA 19, 20
- VAN.VANConnectivity:getFromVAN service 48, 51
- VANFTP delivery service 49, 50
- wm.tn.doc.xml:routeXml service 43
- wm.tn.route:routeBizDoc service 46
- wm.tn.receive service 25, 41, 42, 45

IS document type, defined 68

L

legacy systems, and batching documents 51

logic

- to add to deliver document to service that forms outbound EDI document 39
- to add to service for Execute a Service processing action
 - Group split option 37
 - Interchange split option 38
 - Transaction split option 37
- to process EDI documents returned from VANs 48

M

map data from internal document to EDI document 22

MCI VANs 47

MDN, receipt exchange 11

N

non-standard processing of EDI documents using Trading Networks 30

non-standard processing, WmEDIforTN package 30

O

ODETTE, document format 14

outbound EDI documents

- based on internal-format document 40, 42, 44
- batch, *See* batch documents
- defining TN document types for 39
- Deliver Document By processing action to deliver batch documents 44, 46
- delivering
 - batch documents 54
 - to VAN 39, 40
- forming from internal-format document 39
- in scheduled delivery queue 50
- not saved in Trading Networks database 39
- parsing by EDI recognizer 61
- processing rules 44, 46, 50
 - for batch 55
- split option for EDI outbound documents submitted back to TN document recognition 43
- splitting by EDI recognizer 61

outbound processing
 delivering documents to VANs 48
 diagram of 21
 EDI documents 21
 with Trading Networks 39
 Execute a Service action 39
 out-of-sequence control number, defined 68
 override default pre-processing actions 36

P

packages
 EDI Module 10, 11
 WmEDI 11
 WmEDIforTN 11, 28
 WmEDIsamples 10
 parsing EDI document 61
 partner-specific EDITPA 38
 PGP-encrypted documents
 from VANs 47
 to VANs 49, 50
 pipeline, defined 68
 pre-processing actions
 Check for Duplicate Document 36
 defined in TN document types for EDI documents 36
 Save Document to Database 36
 Validate Structure 36
 Verify Digital Signature 37
 process model
 defined 58, 69
 for business processes 11
 generating 58
 Group document 60
 Interchange document 60
 Transaction document 59
 webMethods Modeler, desinging with 58
 process run time
 ConversationID 62
 defined 58, 69
 EDI document sent to 60

processing
 actions, *See* processing actions
 after
 Group split option 35
 Interchange split option 35
 Transaction split option 34
 EDITPA 29
 inbound EDI documents 14
 with Trading Networks 25
 non-standard processing for EDI documents 30
 outbound EDI documents 21
 with Trading Networks 39
 rules, *See* processing rules
 standard processing for EDI documents 30
 transaction set 17
 processing actions
 Deliver Document By action, *See* Deliver Document By
 processing action
 EDI documents 37
 Group split option 37
 Interchange split option 38
 Transaction split option 37
 processing rules 24, 36
 business process 62
 defined 69
 Deliver Document By action, *See* Deliver Document By
 processing action
 Execute a Service action 37
 Execute a Service action, *See* Execute a Service processing
 action
 inbound EDI documents 36
 internal-format document 41, 43, 45
 outbound EDI document
 batch 55
 to deliver 44, 46
 to form 39
 processing EDI document for batching 53
 routing EDI documents to 40, 44
 service
 to form outbound EDI document and delivers it 40
 to route outbound EDI document back to Trading Networks
 to deliver it 44
 to submit outbound EDI document back to Trading Net-
 works to deliver it 42
 to deliver outbound EDI document 50

profiles 24
 defined 69
program code conventions in this document 5
public queues 24
 See also public scheduled delivery queues
public scheduled delivery queue
 batchProcess delivery service 52
 inputs to service 49
 VANFTP 48

Q

queues
 public, *See* public queues
 public scheduled delivery, *See* public scheduled delivery queues 48
 scheduled delivery 49, 52
 to get EDI documents into public scheduled queue 49
 VANFTP delivery service 48

R

receiver
 ID, defined 69
 qualifier, defined 69
registered Trading Networks delivery service
 EDI Batch 51
 VANFTP 48
response documents, business process 58
resubmitting document to Trading Networks for processing 35
retrieving
 inbound documents waiting on VAN 49, 51
 VAN-generated reports 47, 49
routing outbound EDI document to
 Trading Networks processing rules 40, 44
 Trading Networks to deliver 44
routing to Trading Networks, batched EDI document for delivery 54

S

S/MIME, version 2 11
samples, EDI Module package 10
Save Document to Database pre-processing action 36
saving
 document to Trading Networks database 36
 EDI document to Trading Networks database 32

scheduled delivery queue 49, 52
 Deliver Document By processing action to send document to queue 53
 document associated with
 batchProcess service 53
 VANFTP delivery service 50
 to hold documents to batch 53
scheduling
 batch documents 51
 batchProcess service 52
 VAN.VANConnectivity:getFromVan service 47
 VANFTP delivery service 48
schemas, flat file, *See* flat file schema
SEF file, defined 19, 69
sender ID, defined 69
sender qualifier, defined 69
sending
 batched documents back to Trading Networks 54
 outbound EDI documents 39
 to TN document recognition 39
services
 add logic to deliver document 39
 batchProcess service 54
 delivery 48, 51
 EDI document directly from the service that forms it 40
 EDI Batch delivery service in Trading Networks 51
 Execute a Service processing action 37
 inbound processing of EDI documents 16
 outbound EDI document
 from internal-format document 39
 to form 21, 41, 43, 46
 to invoke logic to deliver 41
 to route to Trading Networks to deliver 44
 to submit to Trading Networks to deliver 42
 when using Trading Networks 39, 40
 VAN.VANConnectivity:getFromVAN service 47, 48, 51
 VANFTP delivery service in Trading Networks 48, 50
 wm.b2b.edi:tn.batch:batchProcess service 51
 wm.tn.doc.wml:routeXML service 43
 wm.tn.route:routeBizDoc service 46
 wm.tn.receive service 25, 41, 42, 45
sign documents, send to VANs 49, 50
Signature system attribute 37
SignedBody system attribute 37

- split option variable 32
 - Batch, *See* Batch split option 32
 - EDITPA 38
 - File, *See* File split option 32
 - Group, *See* Group split option 32
 - in business process 61
 - Interchange, *See* Interchange split option 32
 - processing after 34
 - Transaction, *See* Transaction split option 32
- splitting EDI document 61
- Standard Exchange Format, defined 69
- standard processing of EDI documents using Trading Networks 30
- String format
 - built-in service to convert to IData object 22
 - converting to IData object 18
- submitting to Trading Networks
 - EDI document, in business process 62
 - outbound EDI document
 - document recognition 39
 - for delivery 42
 - picked up EDI document for delivery 47

T

- TN document types 24
 - Check for Duplicate Document not performed 36
 - defined 69
 - defining for outbound document 39
 - EDI recognizer 62
 - for EDI documents 35
 - internal-format document 41, 43, 45
 - pre-processing actions 36
 - saving EDI documents to database 36
- TPAs (Trading Partner Agreements)
 - See also* EDITPAs
 - defined 69
- TRADACOMS
 - transmissions and BAT segment 52
- Trading Networks
 - and EDI Module architecture 10
 - batch EDI documents for delivery 51
 - batchProcess service 54
 - BizDocEnvelope 35
 - document recognition, submitting EDI documents to 39

- EDI
 - Batch delivery service 51
 - clients 25
 - recognizer 28
 - inbound processing of EDI documents 25
 - objects used in EDI Module 24
 - outbound processing of EDI documents 39
 - processing actions for EDI documents 37
 - registered delivery service 48, 51
 - routing batched document back to 54
 - routing to deliver outbound EDI document 44
 - sending EDI documents to process run time 60
 - service to form outbound EDI document 39
 - submitting outbound EDI documents
 - back to TN document recognition 42
 - for delivery 42
 - using EDI Module with TN to connect to VANs 46
 - VAN.VANConnectivity:getFromVAN service 47, 48, 51
 - VANFTP delivery service 48
 - with EDI Module 9
 - wm.tn:receive service 25
- Trading Networks database
 - access header information 39
 - Save Document to Database pre-processing action 36
 - saving original EDI document to 32
- trading partner agreements (TPAs) 24
 - See* EDITPAs
- Transaction documents
 - conversation ID 59
 - create processing rule for Transaction split option 37
 - defined 29, 67, 70
 - diagram of processing 34
 - group envelope 34
 - instance ID query 59
 - interchange envelope 34
 - processing
 - after Transaction split option 34
 - diagram of 34
 - non-standard 30
 - standard 30
- transaction set
 - functional acknowledgement (FA) 19
 - processing within EDI document 17

Transaction split option
 defined 32
 Execute a Service processing action 37
 processing afterwards 34
Transmission documents
 defined 29, 70
transports that clients use 25
 to send EDI documents 15
troubleshooting information 5
tspace, defined 70
typographical conventions in this document 5

U

UCS document format 14
UN/EDIFACT
 document format 14
 interchanges and UNA segment 52
UNA segment for UN/EDIFACT interchanges 52
unknown sender and receiver set in default EDITPA 38

V

Validate Structure pre-processing action 36
validation of
 control numbers in inbound documents 31
 envelope 29
Value Added Network (VAN)
 defined 70
 See VANS
VAN.VANConnectivity:getFromVAN service
 delivering outbound EDI documents to VANS 48
 invoking 48, 51
 optional actions
 inbound processing 47
 outbound processing 49
 retrieving documents waiting on VANS 47
 submitting EDI document it picks up to Trading Networks 48, 51
VANFTP delivery service 48
 inputs 49, 50, 51
 scheduled delivery queue 50
VANS
 and Trading Networks 9
 Deliver Document By processing action to send outbound document to VAN 44, 46, 50
 deliver outbound EDI document to 39, 40
 diagram of retrieving EDI documents from VANS 48

 inbound processing 47
 logic to process returned EDI documents 48
 outbound processing 48
 working with 46
variables, edidata, *See* edidata variable
Verify Digital Signature pre-processing action 37
verifying PGP-encrypted documents from VANS 47
VICS document format 14

W

webMethods EDI Module, *See* EDI Module
webMethods EDIINT Module, description 11
webMethods Integration Server
 See Integration Server
webMethods Modeler
 and EDI Module architecture 11
 designing process model 58
 with EDI Module 9
webMethods Monitor
 and EDI Module architecture 11
 viewing progress and status of business process 59
 with EDI Module 9
webMethods Trading Networks, *See* Trading Networks
webMethods Workflow, business process requiring human interaction 58
wm.b2b.edi:tn.batch:batchProcess service
 See batchProcess service
wm.tn.doc.xml:routeXml service 43
wm.tn.route:routeBizDoc service 46
wm.tn.receive service 25, 41, 42, 45
WmEDI package
 and EDI Module architecture 10
 defined 11
 using without other webMethods components 14
WmEDIforTN package
 and EDI Module 10
 defined 11
 EDI recognizer 28
 inbound processing 28
 non-standard processing of EDI documents 30
WmEDIsamples package 10