



webMethods EDIINT Module

User's Guide

VERSION 6.5.2

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

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

Acrobat and Adobe are registered trademarks, and Reader is a trademark of Adobe Systems Incorporated. Amdocs is a registered trademark, and ClarifyCRM is a trademark of Amdocs. Ariba is a registered trademark of Ariba, Inc. BEA, BEA WebLogic Server, Jolt, and Tuxedo are registered trademarks, and BEA WebLogic Platform is a trademark of BEA Systems, Inc. Action Request System, BMC Software, PATROL, and Remedy are registered trademarks of BMC Software, Inc. BroadVision is a registered trademark of BroadVision, Inc. ChemeStandards and CIDX are trademarks of Chemical Industry Data Exchange. Unicenter is a registered trademark of Computer Associates International, Inc. PopChart is a registered trademark of Corda Technologies, Inc. Kenan and Arbor are registered trademarks of CSG Systems, Inc. Data Connection and SNAP-IX are registered trademarks of Data Connection Corporation. DataDirect, DataDirect Connect, and 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 © 2006 by webMethods, Inc. All rights reserved, including the right of reproduction in whole or in part in any form.

Contents

- About This Guide** 7
 - Document Conventions 7
 - Additional Information 7

- Chapter 1. Concepts** 9
 - What Is EDIINT? 10
 - webMethods EDIINT Module Introduction 10
 - webMethods EDIINT Module Support for EDIINT 11
 - EDIINT Interoperability Tests 11
 - The EDIINT Module and the webMethods Components 12
 - Features Provided for EDIINT Processing 13
 - Inbound EDIINT Processing with AS1 and AS2 13
 - EDIINT Client 14
 - Processing Inbound EDIINT Documents 14
 - Processing Inbound EDIINT MDNs 17
 - Outbound EDIINT Processing with AS1 and AS2 18
 - Sending Outbound EDIINT Documents 19
 - Sending Outbound EDIINT MDNs 20
 - Run-Time Processing with EDIINT AS3 23
 - Run-Time Processing For Host Partners 24
 - Host Partner Sending an AS3 Message to a Client Partner 24
 - Host Partner Retrieving an AS3 Message or MDN From a Client Partner 24
 - Run-Time Processing For Client Partners 24
 - Client Partner Sending an AS3 Message To the Host Partner 24
 - Client Partner Retrieving an AS3 Message or MDN From the Host Partner 24
 - Using a Business Process to Send Outbound EDIINT Documents 25

- Chapter 2. Before You Can Transport Documents Using EDIINT** 27
 - Overview 28
 - Including EDIINT Information in Profiles 28
 - Corporate Tab of the Profile 29
 - Behavior of External ID Matching 29
 - Updating Your Existing Profiles to Use EDIINT AS1, EDIINT AS2, and EDIINT AS3 . . . 29
 - Using the EDIINT ID Match Option 30
 - Delivery Method Tab of the Profile 31
 - Extended Fields Tab of the Profile 32
 - Security Tab of the Profile 34

Configuring SMTP Settings to Enable EDIINT Message Exchange	35
Configuring Inbound EDIINT SMTP Settings	35
Configuring Outbound EDIINT SMTP Settings	36
Configuring Your System To Support EDIINT AS3	37
Creating Directories For Uploading/Downloading	37
Creating Trading Partner Agreements (TPAs) For AS3 Support	38
Configuring Whether Trading Networks Is To Process Payloads	46
Trading Networks Objects Provided for EDIINT	48
TN Document Types	49
Document Attributes	49
Extended Fields	50
Processing Rules	50
Services Invoked by Processing Rules	52
Chapter 3. Creating a Client to Submit a Document Using EDIINT	53
Introduction	54
Content Types to Use	54
Setting the Input Variables for the wm.EDIINT:send Service	54
Chapter 4. Processing Inbound EDIINT Documents and MDNs	55
Processing Inbound EDIINT Documents	56
Before You Can Process Inbound EDIINT Documents	56
Example of an EDIINT Document Posted by HTTP	57
Creating Outbound MDNs that Acknowledge Receipt of an EDIINT Document	57
Setting Up to Process Payloads From EDIINT Documents	58
How the S/MIME Type Profile Field Affects Processing Payloads	59
Processing Inbound EDIINT MDNs	60
Before You Can Process Inbound EDIINT MDNs	60
Example of an EDIINT MDN Posted by HTTP	61
Chapter 5. Using EDIINT to Deliver Outbound Documents	63
Before You Can Deliver Outbound EDIINT Documents	64
Setting the S/MIME Type of the Outbound EDIINT Document	64
Using the wm.EDIINT:send Service to Send EDIINT Documents	64
Chapter 6. Viewing and Managing Information about EDIINT Documents and MDNs ...	67
Using Trading Networks Console to View Information	68
Viewing Related Documents	70
Resubmitting EDIINT Outbound Transactions	71
Chapter 7. webMethods EDIINT Module Services	73
wm.EDIINT	74

Appendix A. Glossary	81
Index	85

About This Guide

This guide is for users of the webMethods EDIINT Module. It provides an overview of the webMethods EDIINT Module and its features.

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:

- **Sample services.** webMethods provides sample services and documentation in the WmEDIsample package, which is located in the [webMethods Knowledge Base](#). The sample services in this package have been certified, meaning that they have been tested by webMethods.
- **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](#).

Concepts

■ What Is EDIINT?	10
■ webMethods EDIINT Module Introduction	10
■ webMethods EDIINT Module Support for EDIINT	11
■ EDIINT Interoperability Tests	11
■ The EDIINT Module and the webMethods Components	12
■ Features Provided for EDIINT Processing	13
■ Inbound EDIINT Processing with AS1 and AS2	13
■ Outbound EDIINT Processing with AS1 and AS2	18
■ Run-Time Processing with EDIINT AS3	23
■ Using a Business Process to Send Outbound EDIINT Documents	25

What Is EDIINT?

EDIINT stands for “Electronic Data Interchange-Internet Integration,” or “EDI over the Internet.” EDIINT is a standard defined by the Internet Engineering Task Force (IETF) that defines a protocol for using the Internet to securely exchange business documents (EDI, XML, or other). The three versions of the EDIINT standard include:

- EDIINT AS1 (EDIINT Applicability Statement 1) that uses SMTP (e-mail) to transport documents
- EDIINT AS2 (EDIINT Applicability Statement 2) that uses HTTP (or HTTP/S) to transport documents
- EDIINT AS3 (EDIINT Applicability Statement 3) that uses FTPS (FTP over SSL) to transport documents

All versions support digital signatures, encryption, and signed receipts. The EDIINT standard specifies how to exchange business documents over the Internet in a secure, reliable, non-repudiable way. It does not specify how to validate or process the business documents that are transported.

webMethods EDIINT Module Introduction

The webMethods EDIINT Module (EDIINT Module) adds support for the EDIINT exchange protocol. Documents using the EDIINT protocol are processed through Trading Networks. As a result, if you want to use the EDIINT protocol, you must use Trading Networks.

The EDIINT Module supports EDIINT AS1 (SMTP), EDIINT AS2 (HTTP), and EDIINT AS3 (FTPS) 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.

You can use EDIINT to transport both EDI and non-EDI formatted (e.g., XML or custom format) documents.

- When you use the EDIINT transport for EDI documents, you must also have the webMethods EDI Module installed. The EDIINT Module passes EDI documents to Trading Networks, which in turn allows the EDI Module to process the EDI document using the functions of the EDI Module packages (WmEDI and WmEDIforTN).
- When you use the EDIINT transport non-EDI documents, the EDIINT Module passes the documents to Trading Networks, and Trading Networks processes based on logic you define in Trading Networks.

webMethods EDIINT Module Support for EDIINT

The EDIINT Module supports the following EDIINT features:

- Securely exchanges business documents using EDIINT AS1, EDIINT AS2, and EDIINT AS3.
- Exclusively uses the S/MIME version 2 cryptographic format to package, compress, encrypt, and provide a digital signature to outbound data, as well as to unpack, decrypt, and verify the authenticity of inbound data.



Note: The S/MIME (Secure/Multipurpose Internet Mail Extensions) standard specifies formats and procedures for providing the cryptographic security services of message authentication, integrity, non-repudiation of origin, and confidentiality.

- Uses the SHA-1 hash algorithm to sign outbound messages, and verifies inbound messages that were signed with either SHA-1 or MD5.
- Enables you to set encryption types and key lengths for each of your trading partners using the extended fields in the partner's Trading Networks' profile. The choices include: Triple DES, DES, RC2 (40 bits), RC2 (64 bits), and RC2 (128 bits).
- Provides the standard outbound encryption permutations (signed, encrypted, signed and encrypted, or plain) at the send-service level.
- Can send receipts of received messages back to the sender as well as receive message receipts. EDIINT receipts are known as MDNs (message disposition notifications). The EDIINT Module can send and receive synchronous or asynchronous, signed or unsigned MDNs.

EDIINT Interoperability Tests

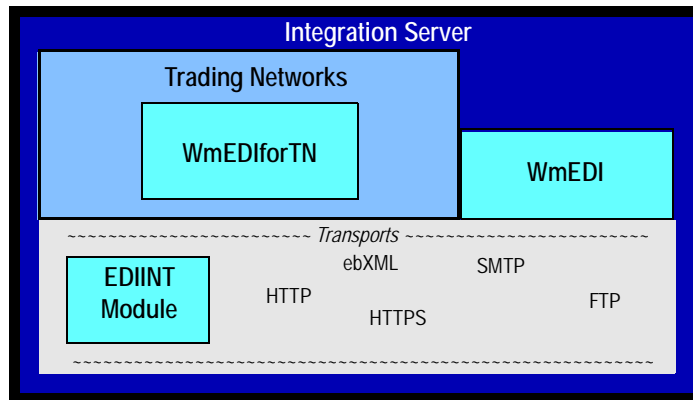
The webMethods EDIINT Module successfully completed the eBusinessReady interoperability testing for both EDIINT AS1, EDIINT AS2, and EDIINT AS3. The eBusinessReady interoperability testing is an industry-neutral software testing program under joint partnership of the Uniform Code Council, Inc. (UCC) and Drummond Group, Inc. (DGI). For more information about the interoperability tests, refer to <http://www.eBusinessReady.org>.

webMethods has participated in all past EDIINT AS2 interoperability tests hosted by the Drummond Group.

The EDIINT Module and the webMethods Components

When you install the webMethods EDIINT Module, the WmEDIINT package is installed into the Integration Server. The following diagram illustrates how the EDIINT Module fits into the webMethods architecture. For more information, see the text after the diagram.

EDIINT Module and the webMethods Architecture



- Integration Server is the underlying foundation of webMethods.
- 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. Trading Networks is required to use the EDIINT Module. For more information about Trading Networks, see the *webMethods Trading Networks Concepts Guide* and the *webMethods Trading Networks User's Guide*.
- EDI Module is comprised of the following two packages:
 - The WmEDI package is the basic functionality that provides support for the EDI standard.
 - 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.
- EDIINT Module contains the support for the EDIINT exchange protocol as described above in "[webMethods EDIINT Module Introduction](#)" on page 10.

Features Provided for EDIINT Processing

After installing the webMethods EDIINT Module, webMethods provides the following to support EDIINT processing:

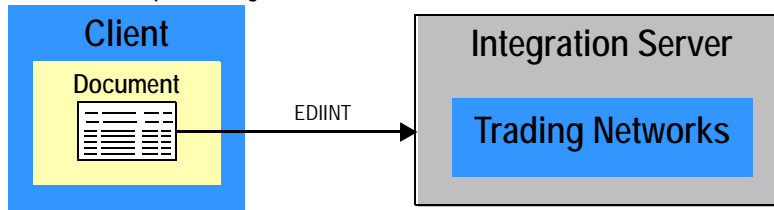
- Content handlers to recognize inbound EDIINT documents and MDNs.
- Built-in services that you use to send outbound EDIINT documents and MDNs and that you use to receive inbound EDIINT documents and MDNs.
- TN document types that are automatically installed into Trading Networks and allow Trading Networks to recognize inbound EDIINT documents and MDNs; the TN document types are set up so all EDIINT documents and MDNs are automatically saved to the Trading Networks database.
- Extended profile fields that are automatically installed into Trading Networks and that are available in Trading Networks profiles for you to supply partner-specific information needed for EDIINT transport-level processing.
- Processing rules that are automatically installed into Trading Networks and are used to perform the EDIINT transport-level processing.
- Delivery services that are automatically registered with Trading Networks and used for delivering EDIINT documents and MDNs.

For more information about how the EDIINT Module uses the items listed above for EDIINT inbound and outbound processing, see [“Inbound EDIINT Processing with AS1 and AS2” on page 13](#) and [“Outbound EDIINT Processing with AS1 and AS2” on page 18](#).

Inbound EDIINT Processing with AS1 and AS2

For inbound processing, a client sends a document to the Integration Server using the EDIINT exchange protocol. The document is processed in the Integration Server using services provided with the EDIINT Module and by using Trading Networks.

Inbound EDIINT processing



EDIINT Client

Use the EDIINT Module to create the client to send documents using EDIINT. If you are *not* using webMethods' software on the client side, see documentation for the EDIINT software that you are using.

When the client sends the EDIINT document to the Integration Server, it must associate the inbound document with a content type that the EDIINT Module recognizes, for example, `multipart/signed`. When the Integration Server receives a document that has an EDIINT content type, it passes the document to the appropriate EDIINT content handler, which was installed when you installed the EDIINT Module.

The EDIINT content handler passes the document to the service the client specifies. To use the EDIINT exchange protocol, the client must specify the `wm.EDIINT:receive` service. The `wm.EDIINT:receive` service is a built-in service provided with the EDIINT Module. For more information about processing the EDIINT document, see [“Processing Inbound EDIINT Documents”](#) below.

Processing Inbound EDIINT Documents

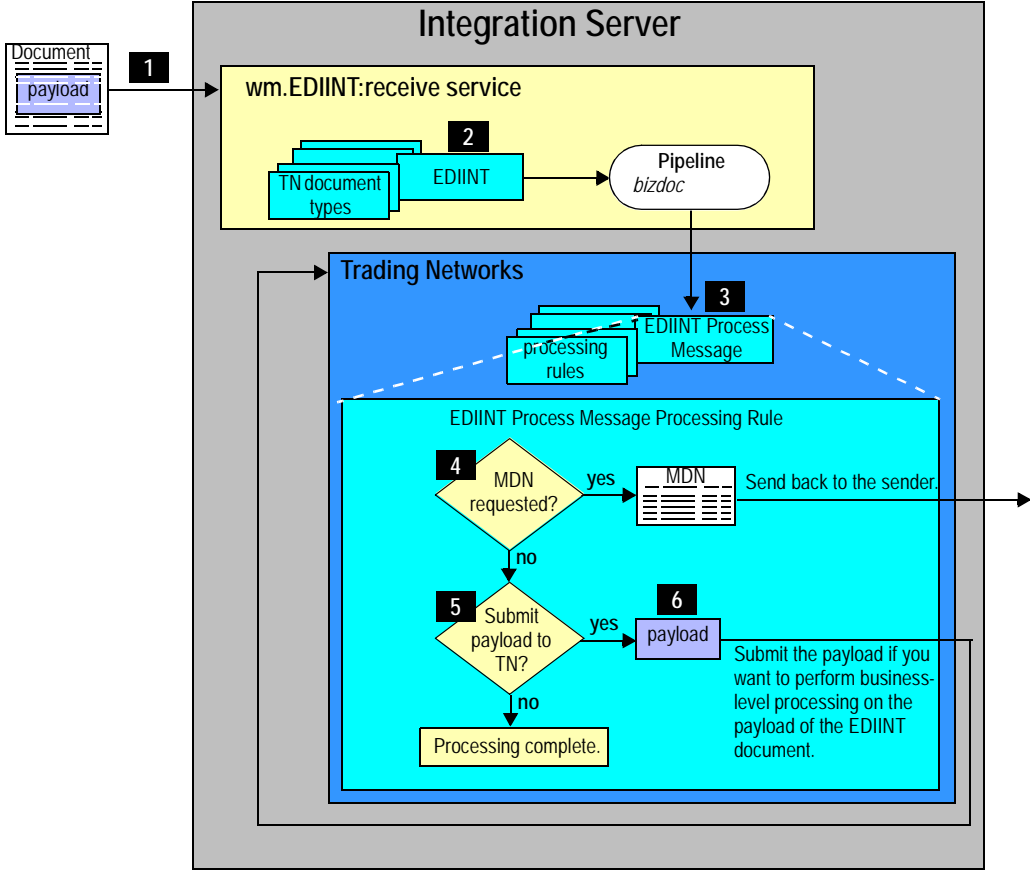
The EDIINT Module provides all logic required to perform the transport-level processing for inbound EDIINT documents. You can use Trading Networks to add business-level processing for the payloads of your EDIINT documents.



Note: For information about how the EDIINT Module processes inbound MDNs, see [“Processing Inbound EDIINT MDNs”](#) on page 17.

The following diagram illustrates how the EDIINT Module performs transport-level processing for an inbound EDIINT document. Additionally, as illustrated in the diagram, if you want to perform business-level processing, you can submit the payload of the EDIINT document to Trading Networks for further processing. For more information, see the table after the diagram.

Processing an inbound EDIINT document



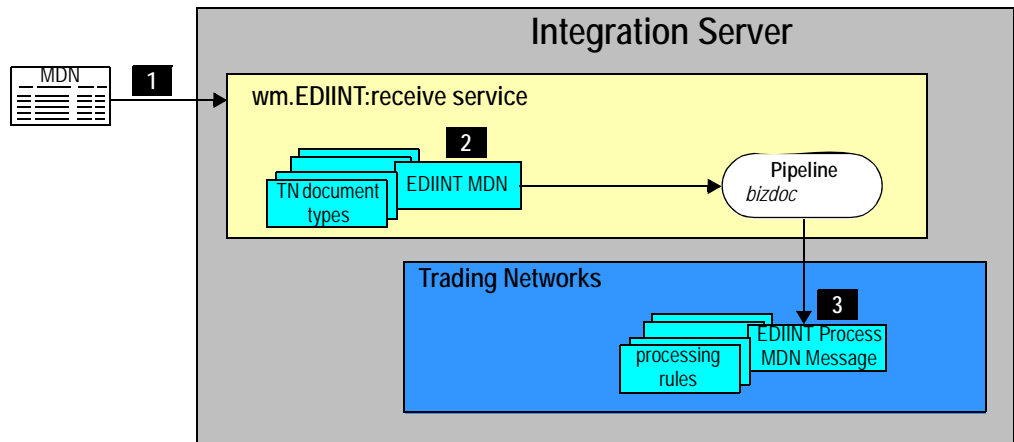
Step	Description
1	The <code>wm.EDIINT:receive</code> service accepts the inbound document.
2	<p>The <code>wm.EDIINT:receive</code> service uses the TN document types to determine the type of document. The document matches the EDIINT TN document type that is installed into Trading Networks when you install the EDIINT Module.</p> <p>After determining the TN document type, the <code>wm.EDIINT:receive</code> service forms a <code>BizDocEnvelope</code> for the inbound document and places it in the pipeline in the <code>bizdoc</code> variable. A <code>BizDocEnvelope</code> contains the original document and includes additional information that Trading Networks requires for routing and processing the document. One piece of information that Trading Networks can use in the selection of a processing rule is the user status. The EDIINT recognizer sets the user status to <code>ProcessMsg</code>.</p> <p>After forming the <code>BizDocEnvelope</code>, the <code>wm.EDIINT:receive</code> service sends <code>BizDocEnvelope</code> to Trading Networks for processing.</p>
3	<p>Trading Networks determines the processing rule to use for the document. For inbound EDIINT documents, Trading Networks uses the EDIINT Process Message processing rule that is installed into Trading Networks when you install the EDIINT Module. It selects this processing rule because the TN document type is EDIINT and the user status is <code>ProcessMsg</code>.</p> <p>This processing rule performs the Execute a Service action to invoke the <code>wm.EDIINT.rules:processMsg</code> service. The service processes the message by opening the MIME or S/MIME package and then decrypting and/or verifying the signature of the message.</p>
The remaining steps represent actions specified in the EDIINT Process Message processing rule.	
4	<p>The <code>wm.EDIINT.rules:processMsg</code> service determines whether the sender of the EDIINT document requested an MDN.</p> <ul style="list-style-type: none"> ■ If the sender did request an MDN, the <code>wm.EDIINT.rules:processMsg</code> service determines whether the sender has requested a signed or an unsigned MDN. The service creates the appropriate type of MDN and sends the MDN back to Trading Networks for delivery. Trading Networks can deliver the MDN synchronously or asynchronously. For more information about how Trading Networks delivers the MDN, see “Sending Outbound EDIINT MDNs” on page 20. After sending the document to Trading Networks, continue with the next step. ■ If the sender did not request an MDN, continue with the next step.

Step	Description
5	<p>The <code>wm.EDIINT.rules:processMsg</code> service invokes the <code>wm.EDIINT.rules:processPayload</code> service, which determines whether you want to send the payload of the EDIINT document to Trading Networks for processing.</p> <p>You define whether you want the EDIINT Module to send EDIINT payloads to Trading Networks for processing when you configure the EDIINT Module.</p>
6	<p>If the payload is to be sent to Trading Networks for processing, submit the payload. You send the payload of the document for processing if you want to perform business-level logic on payload. The payload can be either an EDI document or a non-EDI document (e.g., an XML document).</p> <ul style="list-style-type: none"> EDI documents. If the payload is an EDI document, the EDI Module processes the document. When the payload is an EDI document, you must set up the EDI Module and Trading Networks to process the EDI document. For example, use the EDI Module to install TN document types for the EDI documents and create a processing rule to process the EDI document. For more information about how the EDI Module processes EDI documents, see information about using the EDI Module with Trading Networks in the <i>webMethods EDI Module Concepts Guide</i>. Non-EDI documents. If the payload is not an EDI document, you must define the TN document types that Trading Networks can use to recognize the payload and the processing rule that Trading Networks should use to process the document.

Processing Inbound EDIINT MDNs

The following diagram illustrates how the EDIINT Module processes inbound MDNs. For more information, see the table after the diagram.

Processing inbound MDNs



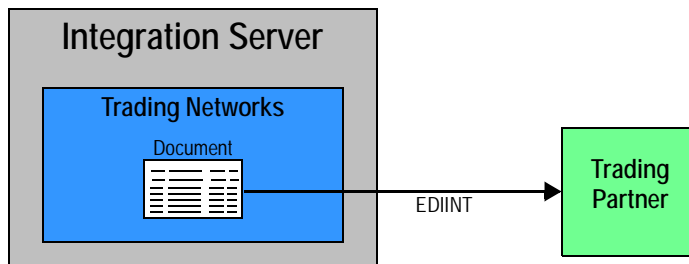
Step	Description
1	The wm.EDIINT:receive service accepts the inbound MDN.
2	<p>The wm.EDIINT:receive service uses the TN document types to determine the type of document. The MDN matches the EDIINT MDN TN document type that is installed into Trading Networks when you install the EDIINT Module.</p> <p>After determining the TN document type, the The wm.EDIINT:receive service forms a BizDocEnvelope for the inbound MDN and places it in the pipeline in the <i>bizdoc</i> variable. A BizDocEnvelope contains the MDN and includes additional information that Trading Networks requires for routing and processing the document. One piece of information that Trading Networks can use in the selection of a processing rule is the user status. The EDIINT recognizer sets the user status to ProcessMDNMsg.</p> <p>After forming the BizDocEnvelope, the wm.EDIINT:receive service sends BizDocEnvelope to Trading Networks for processing.</p>
3	<p>Trading Networks determines the processing rule to use for the MDN. For inbound MDNs, Trading Networks uses the EDIINT Process MDN Message processing rule that is installed into Trading Networks when you install the EDIINT Module. Trading Networks selects this processing rule because the TN document type is EDIINT MDN and the user status is ProcessMDNMsg.</p> <p>This processing rule performs the Execute a Service action to invoke the wm.EDIINT.rules:processMDN service, which processes the EDIINT MDN.</p>

Outbound EDIINT Processing with AS1 and AS2

The documents you want to send using EDIINT can be EDI documents or a non-EDI documents. The EDIINT standard specifies requirements for how to “package” a document for transport and how to transport the document.

To package the document and transport it, you use services provided with the EDIINT Module along with Trading Networks.

Outbound EDIINT processing



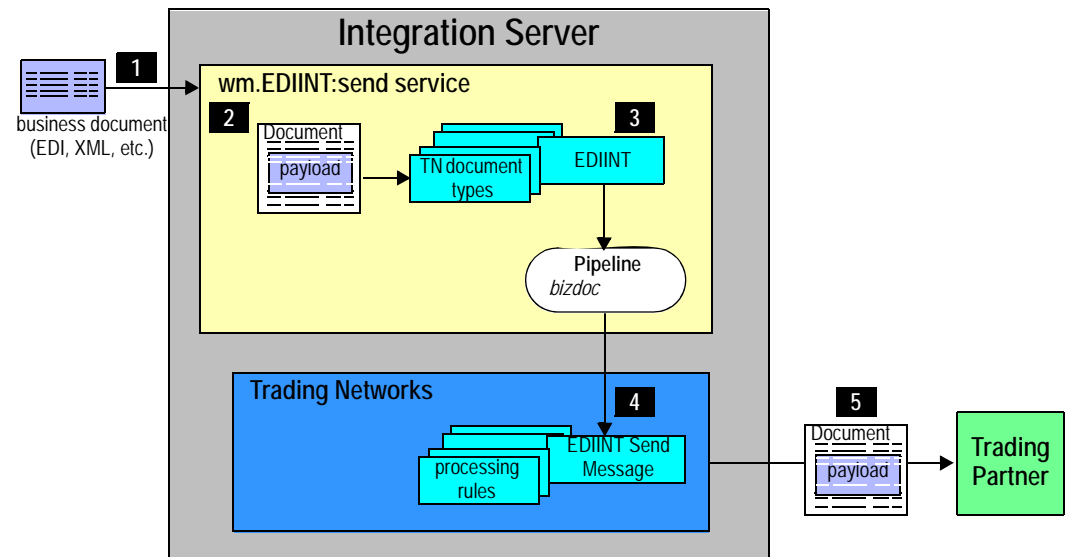
Sending Outbound EDIINT Documents

The EDIINT Module provides all logic required to perform the transport-level processing for sending outbound EDIINT documents.

Note: For information about how to send outbound MDNs, see [“Sending Outbound EDIINT MDNs”](#) on page 20.

The following diagram illustrates how to use the EDIINT Module to send an outbound EDIINT document. For more information, see the table after the diagram.

Sending outbound EDIINT documents



Step	Description
1	You invoke the <code>wm.EDIINT:send</code> service to send a document to the EDIINT Module to be packaged as an EDIINT document and delivered to the receiving trading partner. The inputs of <code>wm.EDIINT:send</code> include the level of encryption to use, whether to compress the data, whether you are requesting an MDN and if so what kind, and how to deliver the document.
2	Using the input information that you provide, the <code>wm.EDIINT:send</code> service creates the EDIINT document, that is, an EDIINT MIME or S/MIME message.

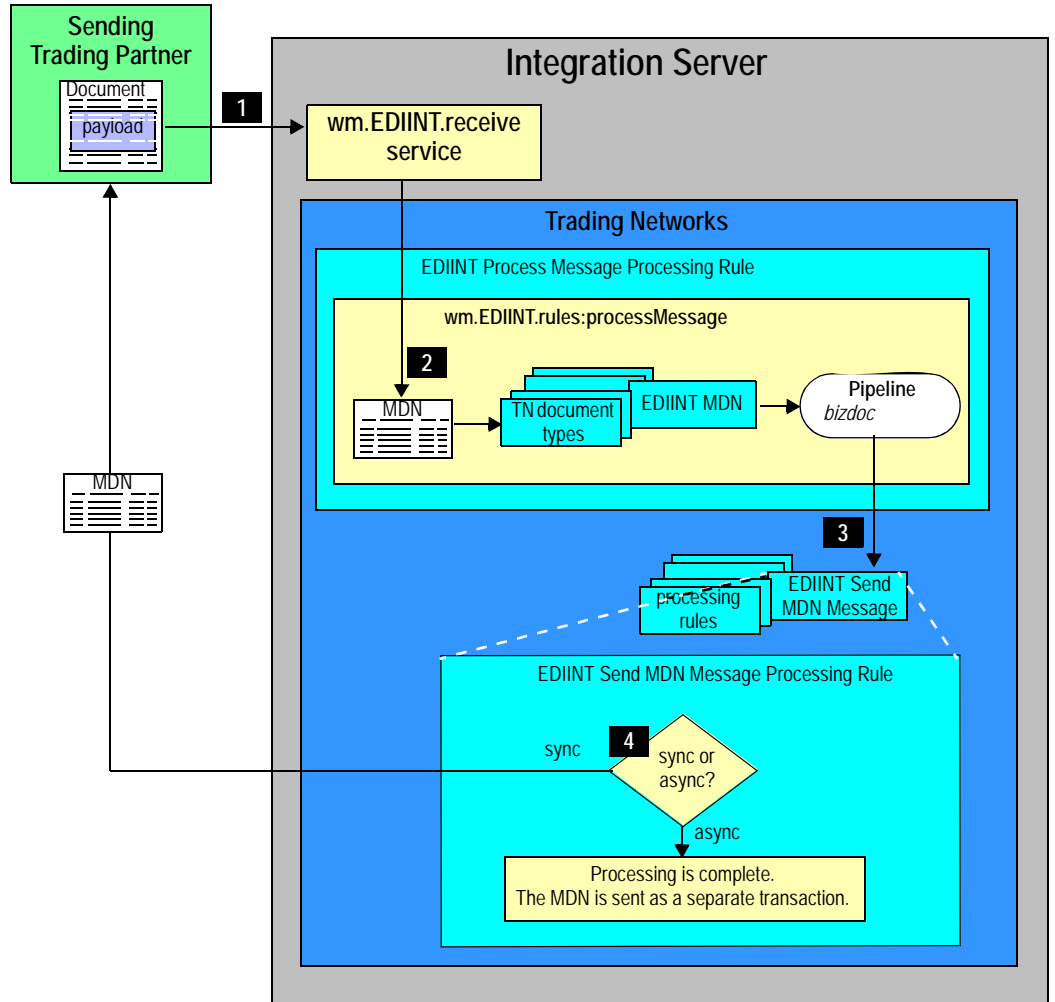
Step	Description
3	<p>The <code>wm.EDIINT:send</code> service uses the TN document types to determine the type of document. The document matches the EDIINT TN document type that is installed into Trading Networks when you install the EDIINT Module.</p> <p>After determining the TN document type, the <code>wm.EDIINT:send</code> forms a <code>BizDocEnvelope</code> for the inbound document and places it in the pipeline in the <code>bizdoc</code> variable. A <code>BizDocEnvelope</code> contains the original document and includes additional information that Trading Networks requires for routing and processing the document. One piece of information that Trading Networks can use in the selection of a processing rule is the user status. The EDIINT recognizer sets the user status to <code>SendMsg</code>.</p> <p>After forming the <code>BizDocEnvelope</code>, the <code>wm.EDIINT:send</code> service sends <code>BizDocEnvelope</code> to Trading Networks for processing.</p>
4	<p>Trading Networks determines the processing rule to use for the document. For outbound EDIINT documents, Trading Networks uses the EDIINT Send Message processing rule that is installed into Trading Networks when you install the EDIINT Module. Trading Networks selects this processing rule because the TN document type is EDIINT and the user status is <code>SendMsg</code>.</p> <p>This processing rule performs the <code>Execute a Service</code> action to invoke the <code>wm.EDIINT.rules:sendMsg</code> service.</p>
5	<p>The <code>wm.EDIINT.rules:sendMsg</code> service delivers the document by invoking the <code>wm.EDIINT.delivery:deliveryDocument</code> service. The <code>wm.EDIINT.delivery:deliveryDocument</code> service delivers the document to the appropriate trading partner.</p>

Sending Outbound EDIINT MDNs

The EDIINT Module automatically sends an outbound MDN when it receives an inbound EDIINT document that requests an MDN. Based on how the sender of the inbound EDIINT document requests the MDN to be sent, the EDIINT Module can send the MDN either synchronously or asynchronously. When the EDIINT Module sends an MDN synchronously, it sends the MDN using the same HTTP connection as that of the inbound EDIINT document. Otherwise, it sends the MDN as a separate transaction.

The following diagram illustrates how the EDIINT Module sends an outbound MDN. For more information, see the table after the diagram.

Sending an outbound MDN



Step	Description
1	The sender sends an EDIINT document to the <code>wm.EDIINT:receive</code> service with a request for an MDN.
2	<p>The <code>wm.EDIINT:receive</code> service accepts the inbound EDIINT document. The <code>wm.EDIINT:receive</code> service passes the EDIINT document to Trading Networks. Trading Networks processes the document using the EDIINT Process Message processing rule. For details about inbound processing, see “Processing Inbound EDIINT Documents” on page 14.</p> <p>The EDIINT Process Message processing rule invokes the <code>wm.EDIINT.rules:processMsg</code> service to process the inbound EDIINT document. Because an MDN is requested, the <code>wm.EDIINT.rules:processMsg</code> service creates the MDN. The <code>wm.EDIINT.rules:processMsg</code> service performs document recognition on the MDN using the Trading Networks TN document types. The MDN matches the EDIINT MDN TN document type. After determining the TN document type, Trading Networks forms a <code>BizDocEnvelope</code> for the MDN and places it in the pipeline in the <code>bizdoc</code> variable. The Trading Networks user status for the MDN is set to <code>SendMDNMsg</code>. The <code>wm.EDIINT.rules:processMsg</code> service then passes the <code>BizDocEnvelope</code> into standard Trading Networks processing.</p>
3	<p>Trading Networks determines the processing rule to use for the MDN. For inbound MDNs, Trading Networks uses the EDIINT Send MDN Message processing rule that is installed into Trading Networks when you install the EDIINT Module. Trading Networks selects this processing rule because the TN document type is EDIINT MDN and the user status is <code>SendMDNMsg</code>.</p> <p>The EDIINT Send MDN Message processing rule performs the Execute a Service action to invoke the <code>wm.EDIINT.rules:sendMDN</code> service.</p>
4	<p>The <code>wm.EDIINT.rules:sendMDN</code> service determines what type of MDN the sender has requested (synchronous or asynchronous) and then sends the MDN accordingly.</p> <ul style="list-style-type: none"> <li data-bbox="448 1261 1349 1358">■ If the sender requested a synchronous MDN, the <code>wm.EDIINT.rules:sendMDN</code> service returns a synchronous MDN to the sender using the same HTTP connection. <li data-bbox="448 1381 1379 1477">■ If the sender requested an asynchronous MDN, the <code>wm.EDIINT.rules:sendMDN</code> service invokes <code>wm.EDIINT.delivery:deliveryDocument</code> service to send an asynchronous MDN as a separate transaction.

Run-Time Processing with EDIINT AS3

To exchange AS3 messages with a trading partner, you use an FTP server that is located either on your system or on your trading partner's system. You use just one FTP server. The partner with the FTP server is referred to as the *host partner*.

The partner who accesses the host partner's FTP server is referred to as the *client partner*. To retrieve the AS3 messages or files, the client partner needs to log in remotely as an FTP client.

To enable the partners to exchange AS3 messages, the host partner must provide the client partner with a particular set of specifications known as a choreography. The AS3 term *choreography* refers to the actions that occur between a client and an FTP server, and the FTP commands that enable those actions to occur. The choreography describes the means for delivering, retrieving, and deleting AS3 messages. It includes information on how an upload is communicated to the server as finished and available for a trading partner to download, such as renaming the file extension. In addition, it states whether the partner who downloads the message must send a delete command to clean up the file, or whether the message is removed through other means within the server.

There are three categories of actions:

- Server logon actions (secure or un-secure)

The EDIINT Module utilizes the secure FTP support provided by the Integration Server. Using this support, clients connect to remote FTP servers using Secure Sockets Layer (SSL).

- Document upload and download actions.
- MDN upload and download actions.

Each partner must specify the choreography information in a Trading Partner Agreement (TPA), as described in [“Creating Trading Partner Agreements \(TPAs\) For AS3 Support” on page 38](#).

Run-Time Processing For Host Partners

If you are a host partner (the partner with the FTP server on your Integration Server), you will send and retrieve AS3 messages as follows:

Host Partner Sending an AS3 Message to a Client Partner

- 1 The host partner invokes the `wm.EDIINT:send` service, which creates an AS3 message and uploads it to the `userFtpRoot\userhome\AS3\inbox` directory on the FTP server.
- 2 The client partner logs in to the host partner's FTP server and retrieves the message from the host partner's `userFtpRoot\userhome\AS3\inbox` directory. The client partner can optionally delete the message.

Host Partner Retrieving an AS3 Message or MDN From a Client Partner

- 1 The client partner logs in to the host partner's FTP server and uploads the AS3 message or MDN to the host partner's `userFtpRoot\userhome\AS3\outbox` directory.
- 2 EDIINT submits the AS3 message or MDN to Trading Networks to be processed.
- 3 EDIINT places an MDN response message (if the TPA specifies it) in the host partner's `userFtpRoot\userhome\AS3\inbox` directory.

Run-Time Processing For Client Partners

If you are a client partner (the partner accessing a remote FTP server on an Integration Server), you will send and retrieve AS3 messages as follows:

Client Partner Sending an AS3 Message To the Host Partner

- 1 The client partner invokes the `wm.EDIINT:send` service, which creates an AS3 message and uploads it to the host partner's FTP server.
- 2 The host partner processes the AS3 message and *puts* an MDN (if the TPA specifies it) on the host partner's FTP server.

Client Partner Retrieving an AS3 Message or MDN From the Host Partner

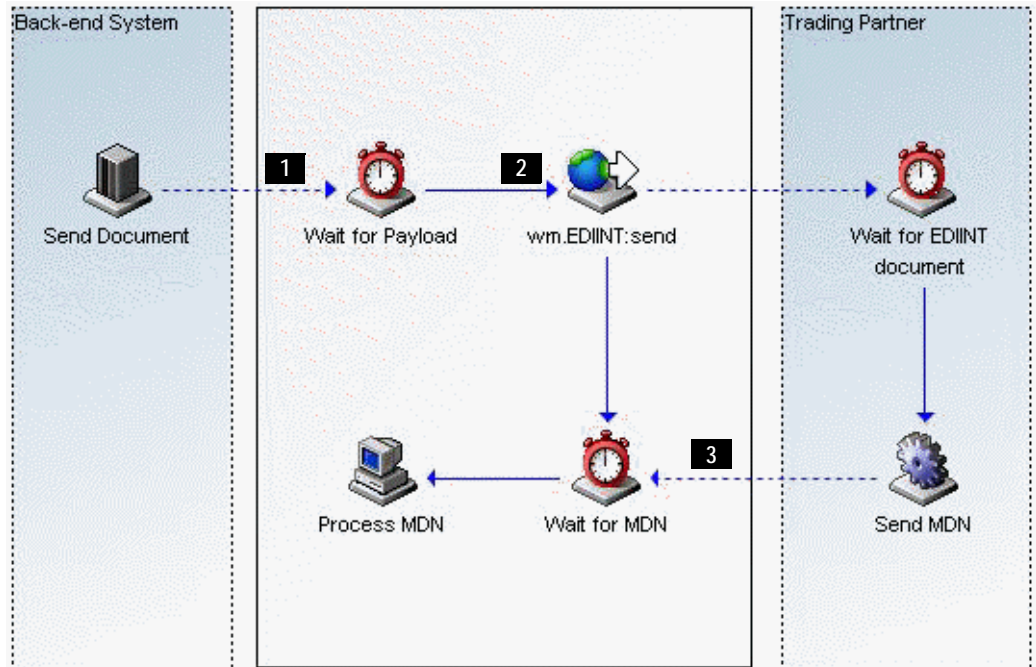
- 1 The client partner invokes the `wm.EDIINT:retrieveAS3Message` service, which logs in to the host partner's FTP server and downloads the AS3 message or MDN. EDIINT deletes the message if the TPA specifies it.
- 2 EDIINT submits the AS3 message or MDN to Trading Networks to be processed.
- 3 If the TPA requires an MDN response message, EDIINT logs in to the FTP server and uploads the MDN to the host partner's FTP server.

For information about configuring your system to support EDIINT AS3, see [“Configuring Your System To Support EDIINT AS3” on page 37](#).

Using a Business Process to Send Outbound EDIINT Documents

You can design a process model that waits for a document that you want to send using the EDIINT transport. You can assign a conversation ID to the outbound EDIINT document. When its corresponding MDN is returned, the EDIINT Module assigns the MDN the same conversation ID, so the MDN can rejoin the conversation.

Sample process model that waits sends an EDIINT document and receives the corresponding MDN



Step	Description
1	The business process waits for a document, for example, from a back-end system. This is the document that you want to send using EDIINT.
2	The next step in the business process is to form an EDIINT document with the back-end system document as the payload, and send the EDIINT document to the trading partner. The step invokes the <code>wm.EDIINT:send</code> service to package the back-end system document as the payload of an EDIINT document. The <code>ConversationID</code> input parameter to the <code>wm.EDIINT:send</code> service is set to define the value to use for the conversation ID. It should be the same conversation ID that the back-end system document used.
3	The trading partner responds with an MDN. The EDIINT Module sets the conversation ID of the MDN to the value specified for the <code>ConversationID</code> input parameter in the preceding step. As a result, the MDN rejoins the correct business process.

Before You Can Transport Documents Using EDIINT

■ Overview	28
■ Including EDIINT Information in Profiles	28
■ Configuring SMTP Settings to Enable EDIINT Message Exchange	35
■ Configuring Your System To Support EDIINT AS3	37
■ Configuring Whether Trading Networks Is To Process Payloads	46

Overview

Before you can transport EDI or non-EDI documents using the EDIINT transport, perform the tasks listed below:

- Ensure that the partner profiles for senders and receivers contain the information that the EDIINT Module requires, as described in [“Including EDIINT Information in Profiles” on page 28](#).
- If you want to use EDIINT AS1 (SMTP), or if you want to use EDIINT AS2 and have partners send MDNs via SMTP, configure the SMTP settings to enable the EDIINT Module to send and receive EDIINT documents via SMTP, as described in [“Configuring SMTP Settings to Enable EDIINT Message Exchange” on page 35](#).
- If you want to use EDIINT AS3 (FTPS), configure your system as described in [“Configuring Your System To Support EDIINT AS3” on page 37](#).



Note: If you use HTTP, HTTPS, or FTPS, see the *webMethods Integration Server Administrator’s Guide* or the *webMethods Integration Server Administrator’s Online Help* for instructions about how to add a port. No EDIINT-specific settings are required to add an HTTP, HTTPS, or FTPS port.

- If you want the EDIINT Module to send the payload of the document to Trading Networks for further processing, configure the EDIINT Module appropriately, as described in [“Configuring Whether Trading Networks Is To Process Payloads” on page 46](#).

Including EDIINT Information in Profiles

To use the EDIINT transport when exchanging documents between partners (senders and receivers), the partners’ profiles in Trading Networks must contain EDIINT information that the EDIINT Module requires.

To modify profiles to include EDIINT information

For steps to create or modify profiles, see the chapter about partner profiles in the *webMethods Trading Networks User’s Guide*.

The following sections identify the specific information that you supply on each tab of the profile to use the EDIINT transport. The sections do *not* include all profile tabs or all fields on the tabs. See the *webMethods Trading Networks User’s Guide* for descriptions of profile fields that are *not* listed in the sections below.

Corporate Tab of the Profile

Provide values for the following fields on the **Corporate** tab:

Corporate Tab field	Description
External ID Type and Value	<p>Select all versions of the EDIINT standard that the partner's corporation uses: EDIINT AS1, EDIINT AS2, and/or EDIINT AS3. You may also select any number of other external ID types, such as User Defined.</p> <p>Then, for each selected external ID type, specify a value to identify the partner. You may assign the same external ID value for multiple external ID types within the same profile. For example, if the partner uses EDIINT AS2 and EDIINT AS3, you can specify the same identification value for both types.</p> <p>The external ID type and its value correspond to the values in the EDIINT document headers "From" and "To" (for AS1), "AS2-From" and "AS2-To" (for AS2), and "AS3-From" and "AS3-To" (for AS3).</p>

Behavior of External ID Matching

The external ID types EDIINT AS1, EDIINT AS2, and EDIINT AS3 are new as of version 6.5.2 of the webMethods EDIINT Module. If you used a version of the EDIINT Module *prior* to version 6.5.2 and you want to use your existing profiles, you have two options:

- Update your existing profiles so they use the external ID types EDIINT AS1, EDIINT AS2, or EDIINT AS3 instead of the existing external ID types. For details, see ["Updating Your Existing Profiles to Use EDIINT AS1, EDIINT AS2, and EDIINT AS3"](#) on page 29.
- OR-
- If you want to continue using the external ID types of your existing profiles without having to change your existing profiles, turn off the new EDIINT ID Match option to make the EDIINT Module behave as it did prior to version 6.5.2. For details, see ["Using the EDIINT ID Match Option"](#) on page 30.



Important! You *must* either update your existing profiles or turn off the EDIINT ID Match option. Otherwise, you may get unpredictable results when trying to match external IDs.

Updating Your Existing Profiles to Use EDIINT AS1, EDIINT AS2, and EDIINT AS3

Prior to version 6.5.2, the EDIINT Module only looked at the external ID value in order to find a matching profile when sending or receiving documents; it ignored the external ID *type*. Thus, each external ID value was required to be unique; a value could not be duplicated within a single profile or in any other profile.

With version 6.5.2, the EDIINT Module looks at both the external ID value *and* the external ID type in order to find a matching profile. This means you may assign the same external ID value for multiple external ID types within the same profile or in any other profile (or the values may be unique).

By default, when you send an EDIINT document the `wm.EDIINT:send` service tries to match the document's "To" header (for example, `AS2-To: 987654321`) with the external ID type and value defined in a partner profile (for example, `EDIINT AS2` and `987654321`).

Similarly, when you receive an EDIINT document the `wm.EDIINT:receive` service tries to match the value of its sender ID input parameter (which specifies both type and value) with a partner profile's external ID type and value.



Important! If you use the `EDIINT AS1`, `EDIINT AS2`, or `EDIINT AS3` types in your profiles, make sure the `EDIINT ID Match` option is selected (this is the default). If you turn off this option, you may get unpredictable results when trying to match external IDs. For details, see ["Using the EDIINT ID Match Option" on page 30](#).

Using the EDIINT ID Match Option

If you do *not* want to use the external ID types `EDIINT AS1`, `EDIINT AS2`, or `EDIINT AS3` in your profiles, turn *off* the `EDIINT ID Match` option. Otherwise, leave this option turned on.

If you fail to set this option appropriately, you may get unpredictable results when trying to match external IDs. For example, if you define a profile with three external ID types that have identical values as follows:

External ID type	Value
EDIINT AS2	987654321
EDIINT AS3	987654321
User Defined	987654321

and you send an EDIINT document with the following headers:

```
AS2-From: 123456789
AS2-To: 987654321
```

the `EDIINT ID Match` option controls the external ID matching as follows:

When EDIINT ID Match is ...	The matching external ID ...
On	Is <code>EDIINT AS2</code> .
Off	Could be either any of the three IDs; you cannot predict which one will match.

 To set the EDIINT ID Match option

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDIINT**.

The Server Administrator opens a new browser window to display the EDIINT Module home page.

- 3 In the navigation panel of the EDIINT Module home page, click **Configuration**.
- 4 Selecting or clearing the EDIINT ID Match check box causes the following behavior:

If EDIINT ID Match option is ...	Then the EDI Module tries to match ...
On (default)	Both the external ID value and the external ID type.
Off	Only the external ID value.

Delivery Method Tab of the Profile

On the **Delivery Method** tab, specify the delivery methods that EDIINT requires. The `wm.EDIINT:send` service uses the information that you specify on this tab to send outbound EDIINT messages and MDNs.

If you are using...	Description
EDIINT AS1	Specify at least one of the following delivery methods: 1) Primary E-mail, 2) Secondary E-mail.
	Note: You must define the delivery method in both the sender and receiver profiles. The e-mail address in the sender's profile is used for the "From" address and the e-mail address in the receiver's profile is used for the delivery e-mail address ("To").
EDIINT AS2	Specify at least one of the following delivery methods: 1) Primary HTTP, 2) Secondary HTTP, 3) Primary HTTPS, 4) Secondary HTTPS.



Note: EDIINT AS3 is an available delivery method as well. For details, see ["Configuring Your System To Support EDIINT AS3" on page 37](#).

Extended Fields Tab of the Profile

The following table lists the **Extended Fields** tab fields you should supply for EDIINT.

Extended Fields Tab field	Description
AS1MDNURL	<p>If you are using EDIINT AS1, use this field to specify the e-mail address that is to accept inbound AS1 EDIINT MDNs. For example, <code>receiver@company.com</code>.</p> <p>An inbound AS1 MDN is by definition asynchronous because it is not returned using the same connection as that of the originally sent document.</p>
AS2MDNURL	<p>If you are using EDIINT AS2, specify the URL that is to accept inbound AS2 EDIINT MDNs.</p> <ul style="list-style-type: none"> ■ To accept MDNs via HTTP, specify a URL that includes the <code>/invoke/</code> element to invoke the <code>wm.EDIINT:receive</code> service. For example (where <code>host:port</code> would be an actual host and port number): <pre>http://host:port/invoke/wm.EDIINT/receive</pre> ■ To accept AS2 MDNs via SMTP, specify a URI similar to the following: <pre>mailto:receiver @company.com</pre> <p>An inbound AS2 MDN could be synchronous (HTTP only) or asynchronous (SMTP or HTTP). A synchronous MDN is returned using the same HTTP connection as that of the originally sent document.</p>
FTPUserName	<p>If you are using EDIINT AS3 and you are the hosting partner (that is, your partner will access your FTP server), specify your partner's user name so that EDIINT can place AS3 messages and MDNs in the <code>userFtpRoot\username\AS3\inbox</code> directory. Your partner will download the messages and MDNs from this directory. For more information about the inbox directory, see “Creating Directories For Uploading/Downloading” on page 37.</p>
Encryption Algorithm	<p>Specify the encryption option to use for outbound EDIINT messages.</p> <p>The default value is Triple DES. The choices include TripleDES, DES, RC2 40 (40 bits), RC2 64 (64 bits), and RC2 128 (128 bits).</p>

Extended Fields Tab field	Description
S/MIME Type	<p>The S/MIME type to use for payloads sent and received by the trading partner. You can specify one of the following values:</p> <ul style="list-style-type: none">■ <code>plain</code>, which means payloads are neither signed nor encrypted.■ <code>signed</code>, which means payloads are signed.■ <code>encrypted</code>, which means payloads are encrypted.■ <code>signedAndEncrypted</code>, which means payloads are signed and encrypted. This is the default. <p>For more information about how the S/MIME Type extended profile field is used during inbound processing, see “How the S/MIME Type Profile Field Affects Processing Payloads” on page 59 in Chapter 4, “Processing Inbound EDIINT Documents and MDNs”. For more information about how the S/MIME Type extended profile field is used during outbound processing, see “Setting the S/MIME Type of the Outbound EDIINT Document” on page 64 in Chapter 5, “Using EDIINT to Deliver Outbound Documents”.</p>

Security Tab of the Profile

The following table lists the information that you should supply for EDIINT on the Security tab of the profile.

Subtabs on the Security Tab	Description
Sign/Verify	<p>If you want the EDIINT Module to be able to digitally sign outbound EDIINT documents, specify the certificates for your corporation along with your private key on the Sign/Verify tab of the profile.</p> <ul style="list-style-type: none"> ■ You can set up a default signing certificate information in the Enterprise profile. ■ If you need to use a specific certificate to sign outbound documents for a particular partner, specify the certificate information on the Sign/Verify tab of that partner's profile. <p>If you expect to receive from a partner, an EDIINT document with a digital signature, and you want the EDIINT Module to be able to verify the digital signature, specify verify certificates on the Sign/Verify tab in the partner's profile. You specify the certificates for your partner's corporation.</p>
Decrypt/Encrypt	<p>If you expect to receive encrypted EDIINT documents from partners, specify the certificates for your corporation along with your private key that you need to use to decrypt the documents on the Decrypt/Encrypt tab of the profile.</p> <ul style="list-style-type: none"> ■ You can set up default decrypting certificate information in the Enterprise profile. ■ If you need to use a specific certificate to decrypt inbound documents from a particular partner, specify the certificate information on the Decrypt/Encrypt tab of that partner's profile. <p>If you want the EDIINT Module to be able to encrypt outbound EDIINT documents for a partner, specify encrypt certificates on the Decrypt/Encrypt tab in the partner's profile. You specify the certificates for your partner's corporation.</p>

Configuring SMTP Settings to Enable EDIINT Message Exchange

To allow the EDIINT Module to receive and send documents using SMTP, you must configure inbound (for receiving) and outbound (for sending) SMTP settings. You must configure these settings if:

- You want to use EDIINT AS1.
- You want to use EDIINT AS2 and have partners send MDNs via SMTP.



Important! Check your mailbox settings for message size limitations that could adversely affect your ability to receive or send large EDIINT documents.

Configuring Inbound EDIINT SMTP Settings

To receive an inbound EDIINT message or MDN via SMTP, use these general guidelines.



To configure your system to be able to receive inbound EDIINT documents via SMTP

- 1 Set up an e-mail account with an e-mail service provider that supports either the POP3 or IMAP protocols.
- 2 From the Server Administrator, add a `webMethods/Email` port that corresponds to the e-mail host that you established in the previous step. For instructions about how to add a port, see either the *webMethods Integration Server Administrator's Guide* or the *webMethods Integration Server Administrator's Online Help*.

The following table lists information you should specify when adding the port:

In this section of the screen...	For this field...	Specify...
Package	Package Name	Either <code>WmEDIINT</code> or <code>WmRoot</code> . <ul style="list-style-type: none"> ■ Specify <code>EDIINT</code> if you want the port disabled when the <code>WmEDIINT</code> package is disabled. ■ Specify <code>WmRoot</code> if you want the port available whenever the server is running.
Server Information	all fields	Information about the e-mail host that you established in the previous step.

<u>In this section of the screen...</u>	<u>For this field...</u>	<u>Specify...</u>
Security	Require authorization within message	No
	Run services as user	A user account with administrator authority, e.g., Administrator.
Message Processing	Global Service	<code>wm.EDIINT:receive</code>
	Default service	leave blank
	Invoke service for each part of multipart message	No
	Include email headers when passing message to content handler	Yes



Note: The EDIINT Module does not use the fields in the Message Processing section of the screen that are not listed in the table above. Leave the fields set to their default values.

- 3 Enable the port.
- 4 Edit the port's Access Mode and click Set Access Mode to Allow by Default.

Configuring Outbound EDIINT SMTP Settings

To send outbound EDIINT documents or MDNs via SMTP, you must configure the name of the SMTP server you want to use for outbound EDIINT documents. You only need to perform this procedure if you want to use EDIINT AS1.



To enable outbound EDIINT SMTP transport

- 1 Open the Server Administrator if it is not already open.
- 2 In the Solutions menu of the navigation panel, click EDIINT. The Server Administrator opens a new browser window to display the EDIINT Module home page.
- 3 In the navigation panel of the EDIINT Module home page, click Configuration.
- 4 In the SMTP Server field, type the name of your SMTP server.
- 5 Click Save Changes.



Note: You can also make this change by directly editing the `wm.EDIINT.SMTPHost` property in the `properties.cnf` file in the following location:

```
webMethods6\IntegrationServer\packages\WmEDIINT\config
```

After you make changes to the `properties.cnf` file, you must restart the server.

Configuring Your System To Support EDIINT AS3

Configuring your system to support EDIINT AS3 involves:

- [“Creating Directories For Uploading/Downloading” on page 37](#)
- [“Creating Trading Partner Agreements \(TPAs\) For AS3 Support” on page 38](#)

Creating Directories For Uploading/Downloading

The host partner must create the following directories in the Integration Server, under the `userFtpRoot` directory, which is the default FTP root directory that the Integration Server creates at startup.

```
userFtpRoot\userhome\AS3
```

where *userhome* is the user’s FTP home directory.



Note: To rename the default `userFtpRoot` directory, use the Integration Server configuration parameter `watt.server.userFtpRoot`. For details, see the *webMethods Integration Server Administrator’s Guide*.

Then the host partner must create the following directories under the AS3 directory:

- `userFtpRoot\userhome\AS3\inbox` — The directory from which the host partner will download AS3 messages.
- `userFtpRoot\userhome\AS3\outbox` — The directory to which the host partner will upload AS3 messages.

Creating Trading Partner Agreements (TPAs) For AS3 Support

A *Trading Partner Agreement (TPA)* in Trading Networks is a set of parameters that you can use to govern how documents are exchanged between two trading partners. One partner fulfills the sender role during document exchange, and the other partner fulfills the receiver role. Both the sender and receiver in a TPA must be a partner in your Trading Networks system that has an existing profile.

Each TPA must specify a unique combination of the following:

- A partner that represents the originator of a send or retrieve operation
- The partner of the originator
- The Agreement ID (the type of the TPA). Use the predefined Agreement ID named EDIINTAS3 to support AS3 message exchange.

You might have multiple TPAs for a pair of trading partners. For example, if PartnerA is the originator of a send or retrieve operation, you would define the following TPA:

TPA field	Value
Sender	PartnerA
Receiver	PartnerB
Agreement ID (type of TPA)	EDIINTAS3

Conversely, if PartnerB is the originator of a send or retrieve operation, you would define the following TPA:

TPA field	Value
Sender	PartnerB
Receiver	PartnerA
Agreement ID (type of TPA)	EDIINTAS3

You define TPAs on the Agreement Details screen of the Trading Networks Console, as described below.

To define a TPA

- 1 Start the Trading Networks Console. If you need procedures for this step, see the *webMethods Trading Networks User's Guide*.
- 2 Select **View** ► **Agreements**. Trading Networks displays the Agreements screen.

- 3 Perform *one* of the following procedures to create a new TPA.



Method to create a TPA	Procedure
<p>New—The TPA fields are empty.</p>	<p>Select Agreements ▶ New. Trading Networks displays the Agreement Details screen to allow you to fill in the information for the TPA.</p>
<p>Duplicate—The TPA fields are filled with values from the TPA agreements that you duplicate. You can update any or all fields from the duplicated TPA agreement.</p> <p>You can use this function to create a template TPA.</p>	<ol style="list-style-type: none"> a Click the row containing the name of the TPA agreement that you want to duplicate. b Select Agreements ▶ Duplicate. Trading Networks displays the Agreement Details screen.

- 4 Fill in the following fields on the Agreement Details screen for the TPA you want to create:



Note: The **Sender**, **Receiver**, and **Agreement ID** fields must be unique for each TPA. The values of these three fields together uniquely identify a TPA. After you create a TPA, you *cannot* change or update these fields of the TPA.

For this TPA field...	Specify
<p>Sender</p>	<p>Name of the trading partner that has the originator role in the transaction that the TPA will govern. Type in the name of the partner or click the Select... button to select the partner from the Partner Selection Dialog. This list includes your own profile (Enterprise).</p> <p>For EDI, to create a template that you will duplicate to create other TPAs, you can use the default value of Unknown.</p>
<p>Receiver</p>	<p>Name of the trading partner that receives the send or retrieve operations from Sender. Type in the name of the partner or click the Select... button to select the partner from the Partner Selection Dialog. This list includes your own profile (Enterprise).</p> <p>For EDI, to create a template that you will duplicate to create other TPAs, you can use the default value of Unknown.</p>

For this TPA field...	Specify
Agreement ID	<p>Specify the Agreement ID <code>EDIINTAS3</code>, which indicates that the type of agreement between the two partners is an AS3 agreement using EDIINT.</p> <hr/> <p>Note: You will not be able to continue creating a TPA unless you specify the Agreement ID.</p> <hr/>
IS Document Type	<p>Specify the IS document type <code>wm.EDIINT.TPA:EDIINTAS3</code>. This IS document type defines the application-specific TPA data. The TPA data is used to govern the exchange of documents between the two partners.</p> <p>Alternatively, click Find IS Document Type  to browse the IS document types and select it. Trading Networks displays the data tree input values of the selected IS document type in the bottom panel of the Agreement Details screen. The right side of this panel displays the input values (TPA data inputs) to variables of the IS document type.</p>
Control Number	Accept the default value zero (0) or leave blank; EDIINT does not use this field.
Data Status	Whether you want to be able to modify the values of the TPA data of the IS document type. The data status is only applicable when the agreement status is Agreed.
Export Service	Leave blank; EDIINT does not use this field.
Initialization Service	<p>Specify the service <code>wm.EDIINT.TPA:initService</code>. This service populates the inputs to the variables in the IS document type <code>wm.EDIINT.TPA:EDIINTAS3</code> with default values.</p> <p>Type in the name of the initialization service located on the server or click Find Service  to browse the services and select the one you want to use.</p>
Description	Optionally, specify a description for the TPA in the Description field. Specify 1-1024 characters. There is no restriction to the characters that you can use.

For more information about creating TPAs, see Chapter 10, "Trading Partner Agreements (TPAs)" of the *webMethods Trading Networks User's Guide*.

- Click the **Set Inputs** icon and provide values for the following upload input parameters for the document type `wm.EDIINT.TPA:EDIINTAS3`:

Upload Parameter	Description
AS3FTPServerLocation	<p>Indicates whether to upload/download AS3 messages to a remote or local FTP server.</p> <ul style="list-style-type: none"> ■ remote — Upload/download AS3 messages to/from the remote FTP server in the FTPUpload and FTPDownload fields. ■ local — Upload/download AS3 messages to/from the local FTP server. No other TPA fields will be used.
uploadService	Always select the <code>wm.EDIINT.delivery.defaultFTPUpload</code> service, which uploads AS3 files.
serverhost	Name or IP address of the FTP server.
serverport	Port number on which the FTP server listens for requests. Default: 21.
dataport	Optional. Listener port number of the data transfer channel, for example, 3345.
username	FTP user on the remote FTP server.
password	Password for the user specified in <code>username</code> .
account	Optional. The user name for an account on the FTP server. Specify a value if your FTP host requires account information. The account is defined in the FTP protocol to further identify the user and password specified in <code>username</code> and <code>password</code> .
transfertype	<p>Specifies the type of the FTP data transfer mode:</p> <ul style="list-style-type: none"> ■ active — Default. Active FTP data transfer mode. ■ passive — Passive FTP data transfer mode.
encoding	Optional. Default character set for encoding data transferred during this session. Specify an IANA-registered character set, for example, <code>ISO-8859-1</code> . If you do not set <code>encoding</code> , the default JVM encoding is used.
timeout	Optional. Time (measured in seconds) to wait for a response from the FTP server before timing out and terminating the request. The default is to wait forever.
secureFTP	Type of the remote FTP server to connect to.

Upload Parameter	Description
securedata	<p>Specifies whether to protect the FTP data channel.</p> <p><code>true</code> — Protect the FTP data channel.</p> <p><code>false</code> — Do not protect the FTP data channel.</p>
auth	<p>Authentication/security mechanism:</p> <ul style="list-style-type: none"> ■ SSL ■ TLS ■ TLS-P
dirpath	<p>Optional. The directory path to which AS3 messages are uploaded. If you do not specify a directory path, the current directory will be used.</p>
fileExtension	<p>Optional. The file extension to be assigned to the uploaded AS3 message file, for example, <code>msg</code>.</p> <p>AS3 message file names are generated using the following naming convention:</p> <p><code>MMddhhmmSSss</code></p> <p>where <code>MM</code> is month, <code>dd</code> is day, <code>hh</code> is hour, <code>mm</code> is minutes, and <code>SSSS</code> is seconds. For example, a generated file name with the extension <code>msg</code> might be <code>122012002222.msg</code>.</p>
renameTo	<p>Optional. After uploading the AS3 message file, move it to a specified directory, and optionally rename their file extension.</p> <p>For example, if your FTP <code>put</code> command places the AS3 file in the <code>tmp</code> directory, and you want to move it to the <code>outbox</code> directory after uploading it, specify this command in the <code>renameTo</code> field:</p> <p><code>/outbox/*</code></p> <p>The wildcard character <code>*</code> is a placeholder for the file name.</p> <p>If you specified a file extension in the <code>fileExtension</code> field (for example, <code>msg</code>), you would specify this command in the <code>renameTo</code> field:</p> <p><code>/outbox/*.msg</code></p> <p>Optionally, you can rename the file extension in this field as well.</p>
MDNDirpath	<p>Optional. The directory to which MDNs are uploaded.</p>

Upload Parameter	Description
MDNFileExtension	<p>Optional. The file extension to be assigned to the uploaded MDN file, for example, <code>mdn</code>.</p> <p>MDN file names are generated using the following naming convention:</p> <p><code>MMddhhmmSSss</code></p> <p>where <code>MM</code> is month, <code>dd</code> is day, <code>hh</code> is hour, <code>mm</code> is minutes, and <code>SSss</code> is seconds. For example, a generated file name with the extension <code>mdn</code> might be <code>122012002222.mdn</code>.</p>
MDNRenameTo	<p>Optional. After uploading the MDN files, move it to a specified directory, and optionally rename its file extension.</p> <p>For example, if your FTP <code>put</code> command places the MDN file in the <code>tmp</code> directory, and you want to move it to the outbox directory after uploading it, specify this command in the <code>renameTo</code> field:</p> <p><code>/outbox/*</code></p> <p>The wildcard character <code>*</code> is a placeholder for the file name.</p> <p>If you specified a file extension in the <code>fileExtension</code> field (for example, <code>mdn</code>), you would specify this command in the <code>renameTo</code> field:</p> <p><code>/outbox/*.mdn</code></p> <p>Optionally, you can rename the file extension in this field as well.</p>
storeUnique	Optional. EDIINT assigns a unique file name after the file is uploaded.
transfermode	<p>Specifies the type of the FTP file transfer mode:</p> <ul style="list-style-type: none"> ■ binary (required for AS3) ■ ascii

- 6 Scroll down and provide values for the following download input parameters for `wm.EDIINT.TPA:EDIINTAS3`:

Download Parameter	Description
downloadService	Always select the <code>wm.EDIINT.delivery.defaultFTPDownload</code> service, which downloads AS3 files.
serverhost	Name or IP address of the FTP server.

Download Parameter	Description
serverport	Port number of the FTP server. Default: 21.
dataport	Optional. Listener port number of the data transfer channel, for example, 3345.
username	FTP user on the remote FTP server.
password	Password for the user specified in <code>username</code> .
account	Optional. The user name for an account on the FTP server. Specify a value if your FTP host requires account information. The account is defined in the FTP protocol to further identify the user and password specified in <code>username</code> and <code>password</code> .
transfertype	Specifies the type of the FTP data transfer mode: <ul style="list-style-type: none"> ■ <code>active</code> — Default. Active FTP data transfer mode. ■ <code>passive</code> — Passive FTP data transfer mode.
encoding	Optional. Default character set for encoding data transferred during this session. Specify an IANA-registered character set, for example, <code>ISO-8859-1</code> . If you do not set <code>encoding</code> , the default JVM encoding is used.
timeout	Optional. Time (measured in seconds) to wait for a response from the FTP server before timing out and terminating the request. The default is to wait forever.
secureFTP	Type of the remote FTP server to connect to.
securedata	Specifies whether to protect the FTP data channel. <ul style="list-style-type: none"> <code>true</code> — Protect the FTP data channel. <code>false</code> — Do not protect the FTP data channel.
auth	Authentication/security mechanism: <ul style="list-style-type: none"> ■ <code>SSL</code> ■ <code>TLS</code> ■ <code>TLS-P</code>
dirpath	Optional. The directory path to which AS3 messages are downloaded. If you do not specify a directory path, the current directory will be used.
filenamepattern	Optional. AS3 message file pattern, for example, <code>*.msg</code> .

Download Parameter	Description
MDNDirpath	Optional. The directory to which MDNs are downloaded. You can specify either the path relative to dirpath or the absolute path.
MDNFilenamepattern	Optional. MDN message file pattern, for example, *.mdn.
deleteFile	Indicates whether the file is to be deleted after downloading it.
transfermode	Specifies the type of the FTP file transfer mode: <ul style="list-style-type: none"> <li data-bbox="705 575 1048 608">■ binary (required for AS3) <li data-bbox="705 624 810 654">■ ascii

7 Click OK to create the TPA.

Configuring Whether Trading Networks Is To Process Payloads


In addition to having the EDIINT Module perform transport-level processing for an entire inbound EDIINT document, you can perform further processing on the payload of the document by having the EDIINT Module send the payload to Trading Networks for separate processing. As installed, the EDIINT Module is configured to submit the payload to Trading Networks.

When the EDIINT Module is configured to send the payload to Trading Networks, the EDIINT Module submits the payload after it completes transport-level processing.

Use the EDI Module to process payloads that have one of the following content types: 1) application/edi-X12, 2) application/EDIFACT, 3) application/XML, or 4) application/edi-consent (which you can use to submit TRADACOMS payloads). If you want to have the EDIINT Module submit payloads that have a different content type, you must provide your own service to process the payload and submit it to Trading Networks.

To enable or disable Trading Networks payload processing

- 1 Open the Server Administrator if it is not already open.
- 2 In the **Solutions** menu of the navigation panel, click **EDIINT**. The Server Administrator opens a new browser window to display the EDIINT Module home page.
- 3 In the navigation panel of the EDIINT Module home page, click **Configuration**.
- 4 Enable or disable the EDIINT Module to or from submitting payloads to Trading Networks by doing one of the following:
 - To enable, select the **Submit payload to TN** check box.
 - To disable, clear the **Submit payload to TN** check box.

 **Note:** You can also make this change by directly editing the `wm.EDIINT.submitPayload` property in the `properties.cnf` file in the following location:

```
webMethods6\IntegrationServer\packages\WmEDIINT\config
```

Specify either `true` (to enable) or `false` (to disable) the submitting of payloads to Trading Networks. After you make changes to the `properties.cnf` file, you must restart the server.

5 In the **User Process Payload Service** field, specify a service that you created to process payloads and submit them to Trading Networks. You *only* need to specify a service if:

- You selected the **Submit payload to TN** check box.

-AND-

- The content types of the inbound payloads are *not* one of the following:
 1) application/edi-X12, 2) application/EDIFACT, 3) application/XML, or 4) application/edi-consent.

The service you specify in the **User Process Payload Service** field must accept the following input variables:

Input Variable	Description
<i>stream</i>	InputStream The payload.
<i>contentType</i>	String The content type of the payload.
<i>EDIINTbizdoc</i>	Document The bizdoc that contains the original EDIINT message. For the structure of <i>EDIINTbizdoc</i> , see the <i>wm.tn.rec:BizDocEnvelope</i> service in the <i>webMethods Trading Networks Built-in Services Reference</i> .

The EDIINT Module ignores your settings in the following situations:

Conditions	Behavior of the EDI Module
<ul style="list-style-type: none"> ■ You select the Submit payload to TN check box. ■ You specify a service in the User Process Payload Service field. 	<p>The EDI Module ignores the service you specify in the User Process Payload Service field.</p>
<p>-AND-</p> <ul style="list-style-type: none"> ■ The inbound payload has a content type that <i>is</i> one of: 1) application/edi-X12, 2) application/EDIFACT, 3) application/XML, or 4) application/edi-consent 	<p>When you select the Submit payload to TN check box, the EDIINT Module <i>always</i> uses the EDI Module to process payloads that have one of the following content types: 1) application/edi-X12, 2) application/EDIFACT, 3) application/XML, or 4) application/edi-consent</p>

Conditions	Behavior of the EDI Module
<ul style="list-style-type: none"> ■ You select the Submit payload to TN check box. ■ You do <i>not</i> specify a service in the User Process Payload Service field. <p style="text-align: center;">-AND-</p> <ul style="list-style-type: none"> ■ The content type of the inbound payload is <i>not</i> one of: <ol style="list-style-type: none"> 1) application/edi-X12, 2) application/EDIFACT, 3) application/XML, or 4) application/edi-consent 	<p>The EDIINT Module ignores the check in the Submit payload to TN check box.</p> <p>The EDIINT Module <i>cannot</i> submit a payload with an unsupported content type to Trading Networks. You must provide a service that submits payloads with unsupported content types.</p>
<ul style="list-style-type: none"> ■ You clear the Submit payload to TN check box. <p style="text-align: center;">-AND-</p> <ul style="list-style-type: none"> ■ You specify a service in the User Process Payload Service field. 	<p>The EDIINT Module ignores the service you specify in the User Process Payload Service field.</p> <p>The EDIINT Module only invokes a service you specify to submit payloads to Trading Networks when you select the Submit payload to TN check box.</p>

6 Click **Save Changes**.



Note: For information about how to set up the EDI Module and Trading Networks to process EDI documents, see the *webMethods EDI Module User's Guide*.

Trading Networks Objects Provided for EDIINT

When you install the EDIINT Module, Trading Networks objects (i.e., TN document types, document attributes, extended profile fields, and processing rules) are installed in Trading Networks for you. This section describes the Trading Networks objects provided with the EDIINT Module.

The information in this section about the Trading Networks objects is for reference *only*. You should *not* alter the definitions of any of the Trading Networks objects.

TN Document Types

The following table describes the TN document types provided for EDIINT processing and the document attributes associated with each. For more information about the document attributes, see [“Document Attributes”](#) below.

TN document type name	Description	Document attributes associated with the TN document type
EDIINT	Trading Networks matches all EDIINT documents to this TN document type.	<ul style="list-style-type: none"> ■ EDIINT Message Type ■ EDIINT Message ID ■ EDIINT Message Digest ■ EDIINT Delivery URL
EDIINT MDN	Trading Networks matches all EDIINT MDNs to this TN document type.	<ul style="list-style-type: none"> ■ EDIINT Message Type ■ EDIINT Message ID ■ EDIINT Delivery URL ■ EDIINT MDN Original Message ID ■ EDIINT MDN Received MIC ■ EDIINT MDN Disposition

Document Attributes

The following table describes the document attributes provided for EDIINT processing. The TN document types described in [“TN Document Types”](#) above extract these attributes from the EDIINT documents and MDNs.

Attribute name	Description	Extracted from
EDIINT Message Type	The protocol that the EDIINT document uses: AS1, AS2, or AS3.	<ul style="list-style-type: none"> ■ EDIINT documents ■ EDIINT MDNs
EDIINT Message ID	The value of the EDIINT Message-ID header, which is also used for the value of the Trading Networks Document ID system attribute.	<ul style="list-style-type: none"> ■ EDIINT documents ■ EDIINT MDNs
EDIINT Message Digest	The message digest calculated for the EDIINT document.	<ul style="list-style-type: none"> ■ EDIINT documents
EDIINT Delivery URL	The destination URL or IP address from the EDIINT document.	<ul style="list-style-type: none"> ■ EDIINT documents ■ EDIINT MDNs

Attribute name	Description	Extracted from
EDIINT MDN Original Message ID	The value of the EDIINT Message-ID header from the original EDIINT document for which the MDN is a receipt.	■ EDIINT MDNs
EDIINT MDN Received MIC	The message digest calculated for the EDIINT MDN.	■ EDIINT MDNs
EDIINT MDN Disposition	The results from processing the original EDIINT document for which the MDN is a receipt.	■ EDIINT MDNs

Extended Fields

When you install the EDIINT Module, the EDIINT field group is added to the Trading Networks profiles. The EDIINT field group contains then following extended profile fields that are used for EDIINT processing. For more information about these extended profile fields, see [“Including EDIINT Information in Profiles” on page 28](#).

Extended profile field name	Description
AS1MDNURL	The e-mail address that is to accept inbound AS1 MDNs.
AS2MDNURL	The URL that is to accept inbound AS2 MDNs.
Encryption Algorithm	The level of encryption to apply to outbound EDIINT messages being sent to the partner.
S/MIME Type	The S/MIME type to use for payloads sent and received by the trading partner.

Processing Rules

The EDIINT Module provides processing rules for EDIINT *transport-level* processing. If you want to do *business-level* processing on the payload of the EDIINT document, you need to:

- Configure EDIINT Module to submit the payload to Trading Networks. For instructions, see [“Configuring Whether Trading Networks Is To Process Payloads” on page 46](#).
- Create your own processing rules to process the payloads. For information about creating processing rules, see the *webMethods Trading Networks User’s Guide*.

The following table describes the processing rules provided for EDIINT transport-level processing:

Processing rule name	Description
EDIINT Process Message	Trading Networks invokes this processing rule for inbound EDIINT documents. The EDIINT Process Message processing rule invokes <code>wm.EDIINT.rules:processMsg</code> service to process the inbound EDIINT document. For more information about the <code>wm.EDIINT.rules:processMsg</code> service, see “Services Invoked by Processing Rules” on page 52 .
EDIINT Process MDN Message	Trading Networks invokes this processing rule for inbound EDIINT MDNs. The EDIINT Process MDN Message processing rule invokes <code>wm.EDIINT.rules:processMDN</code> service to process an inbound MDN message. For more information about the <code>wm.EDIINT.rules:processMDN</code> service, see “Services Invoked by Processing Rules” on page 52 .
EDIINT Send Message	Trading Networks invokes this processing rule for outbound EDIINT documents. The EDIINT Send Message processing rule invokes the <code>wm.EDIINT.rules:sendMsg</code> service to initiate the sending of an outbound EDIINT document. For more information about the <code>wm.EDIINT.rules:processMDN</code> service, see “Services Invoked by Processing Rules” on page 52 .
EDIINT Send MDN Message	Trading Networks invokes this processing rule for outbound EDIINT MDNs. The EDIINT Send MDN Message processing rule invokes <code>wm.EDIINT.rules:sendMDN</code> service to initiate the sending of an outbound EDIINT MDN message. For more information about the <code>wm.EDIINT.rules:processMDN</code> service, see “Services Invoked by Processing Rules” on page 52 .



Important! You should *not* modify or customize these processing rules in any way.

Services Invoked by Processing Rules

The following table describes the services that are invoked by the processing rules described in “[Processing Rules](#)” above. All services are located in the `wm.EDIINT.rules` folder except for `deliveryDocument`, which is located in the `wm.EDIINT.delivery` folder.

Service	Description	Invoked by this processing rule
<code>processMsg</code>	Processes an inbound EDIINT document.	EDIINT Process Message
<code>processPayload</code>	Processes the payload of an EDIINT document. If you configure the EDIINT Module to process payloads, the <code>wm.EDIINT.rules:processMsg</code> service invokes this service to submit the payload to Trading Networks. For more information about configuring EDIINT Module to process payloads, see “ Configuring Whether Trading Networks Is To Process Payloads ” on page 46.	EDIINT Process Message
<code>processMDN</code>	Processes an inbound EDIINT MDN.	EDIINT Process MDN Message
<code>sendMsg</code>	Initiates the sending of an outbound EDIINT document.	EDIINT Send Message
<code>sendMDN</code>	Initiates the sending of an outbound EDIINT MDN document.	EDIINT Send MDN Message
<code>deliveryDocument</code>	Sends an outbound EDIINT document or MDN. The <code>wm.EDIINT.rules:sendMDN</code> and the <code>wm.EDIINT.rules:sendMsg</code> services invoke this service.	EDIINT Send Message -AND- EDIINT Send MDN Message

Creating a Client to Submit a Document Using EDIINT

- Introduction 54
- Content Types to Use 54
- Setting the Input Variables for the wm.EDIINT:send Service 54

Introduction

The EDIINT standard requires the documents you send using the EDIINT transport be “packaged” in a specific way. If your client is running on the Integration Server that has the webMethods EDIINT Module installed, the client should invoke the `wm.EDIINT:send` service to package the document correctly and send it. If you are not using webMethods software for the client, see the documentation for the EDIINT software you are using to determine how to correctly package and send documents. This chapter describes how to create a client using the `wm.EDIINT:send` service of the EDIINT Module.



Note: You can use the `wm.EDIINT:send` service to send both EDI documents and non-EDI documents.

Content Types to Use

You can use any of the EDIINT content types listed below. These content types are for both EDIINT documents and MDNs.

- `multipart/signed`
- `multipart/report`
- `message/disposition-notification`
- `application/pkcs7-signature`
- `application/pkcs7-mime`
- `application/edi-X12`
- `application/EDIFACT`
- `application/XML`
- `application/edi-consent`

Setting the Input Variables for the `wm.EDIINT:send` Service

The client should invoke the `wm.EDIINT:send` service to correctly package a document for EDIINT transport. For the list of input variables that the client should set for the `wm.EDIINT:send` service, see [Chapter 7, “webMethods EDIINT Module Services”](#).

Processing Inbound EDIINT Documents and MDNs

- Processing Inbound EDIINT Documents 56
- Processing Inbound EDIINT MDNs 60

Processing Inbound EDIINT Documents

The webMethods EDIINT Module (EDIINT Module) provides the `wM.EDIINT:receive` service as the entry point for inbound EDIINT documents. That is, clients that send EDIINT documents must invoke the `wM.EDIINT:receive` service. For more about creating a client that sends EDIINT documents, see [Chapter 3, “Creating a Client to Submit a Document Using EDIINT”](#).

For EDIINT transport-level processing, you do *not* need to create or customize services. The EDIINT Module provides all the logic needed to perform transport-level processing. If you want to perform business-level processing on the payload of the EDIINT document, you will need to add your own logic to perform the business-level processing.

Before You Can Process Inbound EDIINT Documents


- Define a profile for the sender and receiver of the EDIINT document. For instruction about how to create profiles, see the chapter about creating profiles in the *webMethods Trading Networks User’s Guide*. For information about adding EDIINT information to profiles, see [“Including EDIINT Information in Profiles” on page 28](#).
- If you want to process the payloads of the EDIINT documents, configure the EDIINT Module to submit the payloads to Trading Networks. For instructions, see [“Configuring Whether Trading Networks Is To Process Payloads” on page 46](#).

To deliver the outbound EDIINT MDN, Trading Networks executes the EDIINT Send MDN Message processing rule, which in turn invokes the `wm.EDIINT.rules:sendMDN` service. The `sendMDN` service determines whether the sender requested a synchronous or asynchronous MDN.

- If the sender requested a synchronous MDN, the `sendMDN` service returns the MDN to the sender using the same HTTP connection.
- If the sender requested an asynchronous MDN, the `sendMDN` service invokes the `wm.EDIINT.rules:deliveryDocument` service to send the MDN as a separate transaction.

Setting Up to Process Payloads From EDIINT Documents

Do the following to perform business-level logic on the payloads of EDIINT documents.

 To perform business-level processing on the payloads of EDIINT documents

- 1 Configure the EDIINT Module to process payloads. For instructions, see [“Configuring Whether Trading Networks Is To Process Payloads” on page 46](#).
- 2 Set the value of the S/MIME Type extended field in the senders’ profiles to specify the S/MIME type you expect inbound EDIINT documents to use. The EDIINT Module uses the S/MIME type you specify to determine whether to process the payload. For more information, see [“How the S/MIME Type Profile Field Affects Processing Payloads” on page 59](#). For more information about setting up profiles for EDIINT, see [“Including EDIINT Information in Profiles” on page 28](#).
- 3 Set up Trading Networks to process the payload and perform the business-level logic.
 - If the payload is an EDI document, see the *webMethods EDI Module User’s Guide* for how to set up the EDI Module to process the EDI documents.
 - If the payload is *not* an EDI document, you must:
 - Define a TN document type for the payload. For instructions, see the chapter about TN document types in the *webMethods Trading Networks User’s Guide*.
 - Define a processing rule for the payload document. For instructions, see the chapter about processing rules in the *webMethods Trading Networks User’s Guide*.

How the S/MIME Type Profile Field Affects Processing Payloads

The EDIINT Module determines the S/MIME type used by the inbound EDIINT document, that is whether the inbound document is plain, signed, encrypted or signed and encrypted. The EDIINT Module then compares the S/MIME type of the inbound document with the value of the S/MIME Type extended profile field of the sender's profile. The table below describes the actions the EDIINT Module takes based on the outcome of the comparison.

Value of S/MIME Type extended profile field	S/MIME type of the inbound document	Action the EDIINT Module takes for the inbound document
plain	any value	Processes the payload.
signed	signed signed and encrypted	Processes the payload.
	plain encrypted	Logs an error message to the Trading Networks activity log and does <i>not</i> process the payload.
encrypted	encrypted signed and encrypted	Processes the payload.
	plain signed	Logs an error message to the Trading Networks activity log and does <i>not</i> process the payload.
signedAndEncrypted	signed and encrypted	Processes the payload.
	plain signed encrypted	Logs an error message to the Trading Networks activity log and does <i>not</i> process the payload.

Processing Inbound EDIINT MDNs

The EDIINT Module provides the `wM.EDIINT:receive` service as the entry point for inbound EDIINT MDNs. That is, clients that send EDIINT MDNs must invoke the `wM.EDIINT:receive` service. For more about creating a client that sends EDIINT documents, see [Chapter 3, “Creating a Client to Submit a Document Using EDIINT”](#).

You do *not* need to create or customize services to process inbound EDIINT MDNs. The EDIINT Module provides all the logic needed to perform the processing.

Before You Can Process Inbound EDIINT MDNs

Define a profile for the sender and receiver of the EDIINT MDN. For instruction about how to create profiles, see the chapter about creating profiles in the *webMethods Trading Networks User’s Guide*. For information about adding EDIINT information to profiles, see [“Including EDIINT Information in Profiles”](#) on page 28.

Example of an EDIINT MDN Posted by HTTP

```

AS2-From: 987654321
AS2-To: 123456789
Message-ID: <2038921766.1012252564086.JavaMail.zhenzhou@zhenzhou>Content-Type:
multipart/signed; protocol="application/pkcs7-signature"; micalg=SHA-1;
boundary="-----_Part_20_-1967424986.1012252564076"
-----=_Part_20_-1967424986.1012252564076Content-Type: multipart/report; Report-
Type=disposition-notification; boundary="-----_Part_19_568293921.1012252564056"
-----=_Part_19_568293921.1012252564056Content-Type: text/plainContent-Transfer-
Encoding: 7bit
MDN for -
  Message ID: <128678451.1012252560430.JavaMail.zhenzhou@zhenzhou>
  From: 123456789
  To: 987654321
  Received on: 2002-01-28 at 16:16:04 (EST)
  Status: processed
  Comment: This is not a guarantee that the message has been completely processed
or understood by the receiving translator
-----=_Part_19_568293921.1012252564056Content-Type: message/disposition-
notificationContent-Transfer-Encoding: 7bit
Reporting-UA: webMethods Integration ServerOriginal-Recipient: 987654321Final-
Recipient: 987654321Original-Message-ID:
<128678451.1012252560430.JavaMail.zhenzhou@zhenzhou>Received-content-MIC:
qZvJD2+2H/OAQYa3+uIZUIyNUaw=, SHA-1Disposition: automatic-action/MDN-sent-
automatically; processed
-----=_Part_19_568293921.1012252564056--
-----=_Part_20_-1967424986.1012252564076Content-Type: application/pkcs7-
signature; name=smime.p7sContent-Transfer-Encoding: base64Content-Disposition:
attachment; filename=smime.p7s
MIAGCSqGSIB3DQEHAQCAMIACAQExCzAJBgUrDgMCGGUAMIAGCSqGSIB3DQEHAQAAMYIBujCCAbYCAQEwYD
BbMQswCQYDVQQGEwJVUzEXMBUGA1UEChMod2ViTWV0aG9kcyBJbmMxZDZANBgNVBAAsTB1BEIEVESTEiMCAG
A1UEAxMZRUJRSU5UIHNhbXBsZSBSZW50aXZlciBDQkIBATAJBGUrDgMCGGUAOIGxMBGCSqGSIB3DQEJAz
ELBgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTAyMDEyODIxMTYwNFowIwYJKoZIhvcNAQkEMRYEFP0/
GE3KNoRkF6KUtngD0m40bUxEMFIGCSqGSIB3DQEJDzFFMEMwCgYIKoZIhvcNAwcwDgYIKoZIhvcNAwICAg
CAMA0GCCqGSIB3DQMCAGFAMA0GCCqGSIB3DQMCAGeOMA0GBSsOAwIHMA0GCSqGSIB3DQEBAQUABIGAJBb3
whwo+h0PsmEyPMXQHIpjFS5fa5w8PIipHQ9nfJV0TTbp5VTL4zT1E34vjESoktGBYmYnD+gTTe2aEB3PoI
qCym25Lv2MZuvcSVNoa2hS4hrCnDwmYNqbFyS1V2ZAqodgBELztd71eeIgnXLUL1/R65gFOJw72Wto0xi8Q
930AAAAAAAA=
-----=_Part_20_-1967424986.1012252564076--

```


Using EDIINT to Deliver Outbound Documents

- Before You Can Deliver Outbound EDIINT Documents 64
- Setting the S/MIME Type of the Outbound EDIINT Document 64
- Using the wM.EDIINT:send Service to Send EDIINT Documents 64

Before You Can Deliver Outbound EDIINT Documents

Define a profile for the sender and receiver of the EDIINT document. For instruction about how to create profiles, see the chapter about creating profiles in the *webMethods Trading Networks User's Guide*. For information about adding EDIINT information to profiles, see [“Including EDIINT Information in Profiles” on page 28 in Chapter 2, “Before You Can Transport Documents Using EDIINT”](#).

Setting the S/MIME Type of the Outbound EDIINT Document

You can define the S/MIME type that you want the webMethods EDIINT Module (EDIINT Module) to use for an outbound EDIINT document; that is, whether you want to send the outbound EDIINT message:

- Without signing or encrypting (`plain`)
- Signing only (`signed`)
- Encrypting only (`encrypt`)
- Signing and encrypting (`signedAndEncrypted`)

You specify the S/MIME type that you want the EDIINT Module to use by using one of the following:

- The `type` input variable to the `wm.EDIINT:send` service
- The S/MIME Type extended profile field and setting the `type` input variable to the `wm.EDIINT:send` service to `getFromProfile`. For more information about setting the S/MIME Type extended profile field, see [“Extended Fields Tab of the Profile” on page 32 in Chapter 2, “Before You Can Transport Documents Using EDIINT”](#).

Using the wm.EDIINT:send Service to Send EDIINT Documents

The EDIINT Module provides the `wm.EDIINT:send` service to send EDIINT documents. This service performs all the necessary EDIINT transport-level processing.

To use the `wm.EDIINT:send` service:

- Set the `wm.EDIINT:send` service input variables as necessary. You must create a `java.io.InputStream` from the EDI or XML data and pass it the input variable `data/stream`. For a description of key input variables, see [“Setting the Input Variables for the wm.EDIINT:send Service” on page 54 in Chapter 3, “Creating a Client to Submit a Document Using EDIINT”](#). For a complete description of this service, see [Chapter 7, “webMethods EDIINT Module Services”](#).
- Invoke the `wm.EDIINT:send` service from a service that you create.



Important! Do *not* invoke the wm.EDIINT:send service directly from the Trading Networks Execute a Service processing action. Doing so will produce processing errors. The service that you create to invoke the wm.EDIINT:send service can be invoked directly from the Execute a Service processing action.

Viewing and Managing Information about EDIINT Documents and MDNs

- Using Trading Networks Console to View Information 68
- Resubmitting EDIINT Outbound Transactions 71

Using Trading Networks Console to View Information

Because EDIINT documents and MDNs are processed through Trading Networks, the webMethods EDIINT Module (EDIINT Module) takes advantage of Trading Networks features. One of the features is to use the Trading Networks system attribute, **User Status**, to assign a user-defined status to a document. The EDIINT Module assigns statuses to EDIINT documents and MDNs as it processes the document.

You can view the **User Status** associated with a document from the **Transaction Analysis** screen of the Trading Networks Console. For instructions on how to view information about documents using the **Transaction Analysis** screen, see the *webMethods Trading Networks User's Guide*.

The following table describes the values of the **User Status** system attribute for EDIINT transport-level processing.

Type	User Status	Description
Inbound EDIINT document	ProcessMsg	The EDIINT document has been received and processing is starting.
	ProcessMsg PAYLOAD	The EDIINT document was successfully processed. The EDIINT Module is configured to process payloads and the <code>wm.EDIINT.rules:processPayload</code> service has been invoked.
	ProcessMsg ERROR	One of the following: <ul style="list-style-type: none"> ■ The EDIINT document contained an invalid sender ID or receiver ID. ■ The message could not be decrypted or verified.

Type	User Status	Description
Inbound EDIINT MDN	ProcessMDNMsg	The EDIINT MDN has been received and processing is starting.
	ProcessMDNMsg DONE	The MDN was processed.
	ProcessMDNMsg ERROR	One of the following: <ul style="list-style-type: none"> ■ The MDN contained an invalid sender ID or receiver ID. ■ The signature of the MDN could not be verified. ■ The MDN contained errors. ■ The MDN digest did not match that of the original EDIINT document.
	ProcessMDNMsg IGNORED	An identical MDN was previously received.
Outbound EDIINT document	SendMsg	Processing to send the EDIINT document has started.
	SendMsg DONE	One of the following: <ul style="list-style-type: none"> ■ The message was sent and an MDN was not requested. ■ The message was sent and an MDN was returned.
	SendMsg WAITMDN	The message was sent and an MDN was requested, but not yet received.
	SendMsg ERROR	One of the following: <ul style="list-style-type: none"> ■ The message could not be sent. ■ The returned MDN contained errors.
Outbound EDIINT MDN	SendMDNMsg	Processing to send the EDIINT MDN has started.
	SendMDNMsg DONE	The MDN was sent successfully.
	SendMDNMsg ERROR	The MDN could not be sent.



Viewing Related Documents

If you submit the payload of an inbound EDIINT document to Trading Networks for business-level processing, you can view information about the envelope(s) that the payload contains, using the **View Related Documents** option of the Trading Networks Console.

By default Trading Networks persists to the Trading Networks database any EDIINT document that is submitted to it. If the document, which is stored in the pipeline variable *bizdoc*, has *not* been persisted (or if the `wm.EDIINT.rules:receive` service fails to place the document into *bizdoc*), the following occurs depending on the content type of the payload:

Payload content type	If the bizdoc has <i>not</i> been persisted...
application/XML	No information about the payload is displayed.
application/edi-X12 application/EDIFACT application/edi-consent	Information about the main envelope is <i>not</i> displayed; information about each individual envelope is displayed if they have been persisted.
other	If you specified a user-defined service to handle another content type (as described in “Configuring Whether Trading Networks Is To Process Payloads” on page 46), the <code>webMethods EDIINT Module</code> passes the <i>bizdoc</i> to the user-defined service, along with the payload’s data stream and content type. The user-defined service must handle the display of payload information.

To view related documents

- 1 On the Trading Networks Console’s Transaction Analysis screen, display the Detail view. If you are currently displaying the Summary view, select **View Detail/Summary** to switch to the Detail view.
- 2 Click  to display the query panels if they are not already displayed.
- 3 Specify search criteria to find the inbound document you want to view and click the **Run Query**  button on the toolbar. For more information about creating queries on this screen, see Chapter 18, “Managing and Tracking Documents”, in the *webMethods Trading Networks User’s Guide*.
- 4 Click the row containing the inbound document for which you want to view related documents.
- 5 Select **View Related Documents**.



Resubmitting EDIINT Outbound Transactions


You can resubmit EDIINT outbound transactions. When you resubmit a document, Trading Networks performs the following processing:

- Uses the TN document type definitions to recognize the document
- Performs a processing rule lookup to determine the rule to use
- Performs the pre-processing and processing actions identified in the matching processing rule

Resubmit an outbound transaction if the document was not recognized when it was originally received. Create or modify a TN document type definition to recognize the document before processing the document again.

To resubmit an EDIINT outbound transaction

- 1 On the Trading Networks Console's Transaction Analysis screen, select **View ▶ Trading Partners**.
- 2 Use the **Select...**  button to select the partner with which you want to associate the document query. Select the partner from the **Partner Selection Dialog**.
- 3 Select the sender from the **Partner Selection Dialog**.
- 4 Select **View ▶ Transaction Analysis**.
- 5 Display the **Detail** view. If you are currently displaying the **Summary** view, select **Transactions ▶ View Detail/Summary** to switch to the **Detail** view.
- 6 If the query panels are not displayed, select **Transactions ▶ Show/Hide Query** to display the query panels.
- 7 Specify search criteria to find the outbound document and click the **Run Query**  button on the toolbar. For more information about creating queries on this screen, see Chapter 18, "Managing and Tracking Documents", in the *webMethods Trading Networks User's Guide*.

 **Note:** To find documents that were processed but encountered errors in the recognition process, specify the processing status **DONE W/ ERRORS** for the processing status in the basic criteria.

- 8 Click the row containing the outbound document that you want to resubmit.
- 9 Select **Resubmit**.

webMethods EDIINT Module Services

- [wm.EDIINT](#) 74

wm.EDIINT

Use the services in this folder for sending and receiving EDIINT (AS1, AS2, or AS3) messages and MDNs.

wm.EDIINT:receive

Receives inbound EDIINT (AS1/AS2/AS3) messages or MDNs and submits the message to Trading Networks to be unwrapped, decrypted, and signature authenticated.

The EDIINT content handler populates the inputs described below.

Input Parameters

<i>protocol</i>	String The EDIINT protocol to use.								
	<table> <thead> <tr> <th>Value of <i>protocol</i></th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>smtp</td> <td>AS1 message or MDN.</td> </tr> <tr> <td>http</td> <td>AS2 message or MDN.</td> </tr> <tr> <td>ftp</td> <td>AS3 message or MDN.</td> </tr> </tbody> </table>	Value of <i>protocol</i>	Meaning	smtp	AS1 message or MDN.	http	AS2 message or MDN.	ftp	AS3 message or MDN.
Value of <i>protocol</i>	Meaning								
smtp	AS1 message or MDN.								
http	AS2 message or MDN.								
ftp	AS3 message or MDN.								
<i>message-ID</i>	String The EDIINT message ID of the EDIINT message or MDN. This value becomes the Trading Networks system attribute, Document ID .								
<i>contentType</i>	String The content type of the EDIINT message or MDN.								
<i>stream</i>	Object The data InputStream representing the inbound EDIINT message or MDN.								
<i>AS2-From</i>	String The sender ID from the EDIINT AS2 message/MDN. This should match the sender's Trading Networks external ID.								
<i>AS2-To</i>	String The receiver ID from the EDIINT AS2 message/MDN. This should match the receiver's Trading Networks external ID.								
<i>AS3-From</i>	String The sender ID from the EDIINT AS3 message/MDN. This should match the sender's Trading Networks external ID.								
<i>AS3-To</i>	String The receiver ID from the EDIINT AS3 message/MDN. This should match the receiver's Trading Networks external ID.								
<i>From</i>	String The sender ID from the EDIINT AS1 message/MDN. This should match the sender's Trading Networks external ID.								
<i>To</i>	String The receiver ID from the EDIINT AS1 message/MDN. This should match the receiver's Trading Networks external ID.								
<i>ReceiptDeliveryOption</i>	String The address to which to send an asynchronous MDN, if requested.								

Output Parameters

None

Usage Notes

For information about how to use this service, see [Chapter 4, “Processing Inbound EDIINT Documents and MDNs”](#).

wm.EDIINT:retrieveAS3Message

Downloads AS3 messages or MDNs from a partner's remote FTP server and submits the message to Trading Networks to be unwrapped, decrypted, and to have its signature authenticated. The service locates the partner's remote FTP server using the values defined in IS document type `wm.EDIINT.TPA:EDIINTAS3TPA`. If the IS document type specifies that the retrieved file(s) are to be deleted, the service deletes the file(s).

Input Parameters

sender Document Optional. Identification of the partner from whom to retrieve the message.

Note: When *sender* is not specified, the service will access the remote FTP servers of *all* partners with whom you have a Trading Partner Agreement.

Variable in

<i>sender</i>	Meaning
---------------	---------

<i>id</i>	String The partner's external ID.
-----------	-----------------------------------

<i>idTypeDesc</i>	String Optional. The partner's external ID type. Default: AS3.
-------------------	--

receiver Document Identification of the partner who retrieves the message.

Variable in

<i>sender</i>	Meaning
---------------	---------

<i>id</i>	String The partner's external ID.
-----------	-----------------------------------

<i>idTypeDesc</i>	String Optional. The partner's external ID type. Default: AS3.
-------------------	--

Output Parameters

None

wm.EDIINT:send

Constructs an outbound EDIINT message according to the configuration of the input variables, and then submits the message to Trading Networks.

Input Parameters

<i>type</i>	<p>String The S/MIME type that you want to use for the outbound EDIINT message. Specify one of the following:</p> <table> <thead> <tr> <th><u>Value of <i>type</i></u></th> <th><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>plain</td> <td>You want the wm.EDIINT:send service to neither sign nor encrypt the outbound EDIINT message.</td> </tr> <tr> <td>signed</td> <td>You want the wm.EDIINT:send service to sign the outbound EDIINT message.</td> </tr> <tr> <td>encrypted</td> <td>You want the wm.EDIINT:send service to encrypt the outbound EDIINT message.</td> </tr> <tr> <td>signedAndEncrypted</td> <td>You want the wm.EDIINT:send service to sign and encrypt the outbound EDIINT message.</td> </tr> <tr> <td>getFromProfile</td> <td>You want the wm.EDIINT:send service to determine whether to sign and/or encrypt the outbound EDIINT message using the value of the S/MIME Type extended field from the receiver's profile.</td> </tr> </tbody> </table>	<u>Value of <i>type</i></u>	<u>Description</u>	plain	You want the wm.EDIINT:send service to neither sign nor encrypt the outbound EDIINT message.	signed	You want the wm.EDIINT:send service to sign the outbound EDIINT message.	encrypted	You want the wm.EDIINT:send service to encrypt the outbound EDIINT message.	signedAndEncrypted	You want the wm.EDIINT:send service to sign and encrypt the outbound EDIINT message.	getFromProfile	You want the wm.EDIINT:send service to determine whether to sign and/or encrypt the outbound EDIINT message using the value of the S/MIME Type extended field from the receiver's profile.
<u>Value of <i>type</i></u>	<u>Description</u>												
plain	You want the wm.EDIINT:send service to neither sign nor encrypt the outbound EDIINT message.												
signed	You want the wm.EDIINT:send service to sign the outbound EDIINT message.												
encrypted	You want the wm.EDIINT:send service to encrypt the outbound EDIINT message.												
signedAndEncrypted	You want the wm.EDIINT:send service to sign and encrypt the outbound EDIINT message.												
getFromProfile	You want the wm.EDIINT:send service to determine whether to sign and/or encrypt the outbound EDIINT message using the value of the S/MIME Type extended field from the receiver's profile.												
<i>compressed</i>	<p>String (optional) Whether you want the EDIINT message that you are sending to be compressed before it is signed and/or encrypted. Specify either <i>true</i> or <i>false</i>.</p> <table> <thead> <tr> <th><u>Value of <i>compressed</i></u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>true</td> <td>Compress outbound message before signing and encrypting.</td> </tr> <tr> <td>false</td> <td>Do <i>not</i> compress outbound message before signing and encrypting. This is the default.</td> </tr> </tbody> </table>	<u>Value of <i>compressed</i></u>	<u>Meaning</u>	true	Compress outbound message before signing and encrypting.	false	Do <i>not</i> compress outbound message before signing and encrypting. This is the default.						
<u>Value of <i>compressed</i></u>	<u>Meaning</u>												
true	Compress outbound message before signing and encrypting.												
false	Do <i>not</i> compress outbound message before signing and encrypting. This is the default.												

deliveryMethod

String The delivery method you want to use to send the EDIINT document. The `wm.EDIINT:send` service obtains the Trading Networks profile for the receiver (specified by the `receiverID` input variable) and delivers the EDIINT document to the receiver's system using the information specified for the delivery method you specify. Specify a delivery method that is defined in the receiver's Trading Networks profile.

- For EDIINT AS1, specify one of the following:
 - `PrimarySMTP` (corresponds to the Trading Networks Primary E-mail delivery method)
 - `SecondarySMTP` (corresponds to the Trading Networks Secondary E-mail delivery method)
- For EDIINT AS2, specify one of the following:
 - `PrimaryHTTP`
 - `SecondaryHTTP`
 - `PrimaryHTTPS`
 - `SecondaryHTTPS`
- For EDIINT AS3, specify the following:
 - `AS3`

data

Document The payload that you want to send.

Variable in <i>data</i>	Description
<i>contentType</i>	The content type to assign to the outbound message.
<i>stream</i>	The <code>java.io.InputStream</code> that you map from the EDI or XML data.

requestMDN

String Whether you want the receiver to return an MDN. Specify one of the following:

Value of <i>requestMDN</i>	Meaning
<code>none</code>	Do <i>not</i> request a return MDN.
<code>synchronousMDN</code>	Request a return synchronous MDN.
<code>asynchronousMDN</code>	Request a return asynchronous MDN.

Note: If you specify `PrimarySMTP`, `SecondarySMTP`, `PrimaryFTPS`, or `SecondaryFTPS` for *deliveryMethod*, you can only receive an asynchronous MDN.

requestSignedReceipt String Whether you want the MDN to be signed.

Note: *requestSignedReceipt* is ignored when *requestMDN* is `false`.

Value of

requestSignedReceipt

Meaning

true

Request a signed MDN.

false

Request a plain (not signed) MDN.

senderID

Document Identification of the sender of the EDIINT message.

Variable in *senderID*

Description

id

The sender's external ID. That is, the identification that you want for the sender in the message.

idTypeDesc

Optional. The external ID type for the sender ID you specified in *id*. This is an external ID type as defined in Trading Networks.

Note: By default, the service uses the appropriate external ID type based on the value you specify for the *deliveryMethod* parameter. For example, if you specify AS3 for the *deliveryMethod* parameter, the service uses the EDIINT AS3 external ID type. Specify a value for *idTypeDesc* only if you want to override this default.

receiverID

Document Identification of the receiver of the EDIINT message.

Variable in <i>senderID</i>	Description
<i>id</i>	The receiver’s external ID. That is, the identification that you want for the receiver in the message.
<i>idTypeDesc</i>	Optional. The external ID type for the receiver ID you specified in <i>id</i> . This is an external ID type as defined in Trading Networks.
<p>Note: By default, the service uses the appropriate external ID type based on the value you specify for the <i>deliveryMethod</i> parameter. For example, if you specify AS3 for the <i>deliveryMethod</i> parameter, the service uses the EDIINT AS3 external ID type. Specify a value for <i>idTypeDesc</i> only if you want to override this default.</p>	

ConversationID

String (optional) Conversation ID for the outbound EDIINT message.

The *conversationID* parameter is an identifier that links all documents that are part of the same business process (also called a conversation). That is, all documents in the same business process need to have the same *conversationID*. As described in the *webMethods Trading Networks User’s Guide*, Trading Networks can extract *conversationIDs* from EDI documents and use them to pass documents to the webMethods Process Run Time (PRT) after Trading Networks performs the actions identified by a processing rule.

In this field, you might want to specify the same conversation ID as that of the payload that you are sending. The EDIINT Module automatically assigns to a return MDN the same conversation ID that is assigned here.

Note: Leave this field blank unless you own a license for the webMethods Process Run Time (PRT).

Output Parameters

None

Usage Notes

For information about how to use this service, see [Chapter 3, “Creating a Client to Submit a Document Using EDIINT”](#) and [Chapter 5, “Using EDIINT to Deliver Outbound Documents”](#).

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.

AS1

See [EDIINT AS1](#).

AS2

See [EDIINT AS2](#).

AS3

See [EDIINT AS3](#).

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.

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.

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*).

document type

See *IS Document Type* or *TN document type*.

EDIINT

A standard defined by the Internet Engineering Task Force (IETF) that defines a protocol for using the Internet to securely exchange business documents (EDI, XML, or other). EDIINT stands for "Electronic Data Interchange-Internet Integration" or "EDI over the Internet." EDIINT has three version: *EDIINT AS1*, *EDIINT AS2*, and *EDIINT AS3*.

EDIINT AS1

EDIINT Applicability Statement 1, which is a version of the EDIINT standard that uses SMTP (e-mail) to transport documents.

EDIINT AS2

EDIINT Applicability Statement 2, which is a version of the EDIINT standard that uses HTTP (or HTTP/S) to transport documents.

EDIINT AS3

EDIINT Applicability Statement 3, which is a version of the EDIINT standard that uses FTP over SSL (also known as FTPS) to transport documents.

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. Additionally, the EDI Module adds extended fields to profiles for use when transporting documents using EDIINT.

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.

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.

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).

MDNs

EDIINT receipts for messages that are received. The receiver sends the receipt back to the sender. MDN stands for message disposition notifications.

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.

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.

Index

A

- activity log, defined 81
- Agreement ID, specifying for AS3 40
- algorithm, used in EDIINT encryption 11
- application/edi-consent payload processing 46
- application/EDIFACT payload processing 46
- application/edi-X12 payload processing 46
- application/XML payload processing 46
- architecture of webMethods EDIINT Module 12
- AS1, EDIINT version
 - defined 10
 - delivery methods for 31
 - enabling e-mail exchange 35
 - importance of checking mailbox size 35
 - profile field, for EDIINT 31
 - SMTP, configuring 35
 - specifying inbound e-mail address 32, 50
- AS1MDNURL extended profile field for inbound AS1 MDNs 32, 50
- AS2, EDIINT version
 - defined 10
 - delivery methods for 31
 - profile field, for EDIINT 31
 - SMTP, configuring 35
 - specifying inbound address 32, 50
- AS2MDNURL extended profile field for inbound AS2 MDNs 32, 50
- AS3, EDIINT version
 - defined 10
 - directories for downloading/uploading 37
 - overview of 23
 - Trading Partner Agreements (TPAs), creating 38
- asynchronous MDN 11, 16, 22, 58
 - wm.EDIINT.delivery:deliveryDocument service 22

B

- bizdoc variable
 - defined 81
 - EDIINT document 16, 20
 - MDN 18, 22

BizDocEnvelope

- defined 81
- for MDN 22
- formed by wm.EDIINT:receive service 16, 18
- formed by wm.EDIINT:send service 20
- inbound
 - document 20
 - EDIINT document 16
 - MDN 18, 22
- built-in services
 - EDIINT documents and MDNS 13
 - wm.EDIINT:receive service 14
- business process
 - defined 81
 - for inbound EDIINT documents 15
 - on EDIINT payload 17
 - sample process model to send outbound EDIINT documents 25
- business-level processing
 - EDIINT 50
 - logic on payloads of EDIINT documents 58
 - payload of EDIINT document 56

C

- clients for inbound processing using EDIINT exchange protocol 13
- compression, EDIINT messages 11
- configuration file, WmEDIINT properties.cnf 37, 46
- configuring
 - to receive inbound EDIINT documents via SMTP 35
 - to send outbound EDIINT documents via SMTP 36
- content handlers
 - EDIINT 14
 - EDIINT documents 13
 - MDNs 13
- content type, EDIINT 14
- contentType input variable 47
- conventions used in this document 7

conversation ID
 in sample process model to send outbound EDIINT documents 25
 input parameter of process model 25
 MDN 25
 outbound EDIINT document 25
conversation, defined 81
copying document attributes 39
Corporate tab fields, profile for EDIINT 29
creating outbound EDIINT message 77

D

Decrypt/Encrypt subtab, for EDIINT profile 34

defining TPA 38

definitions

 activity log 81

 bizdoc variable 81

 BizDocEnvelope 81

 business process 81

 conversation 81

 conversation ID 81

 EDIINT 10

 AS1 10

 AS2 10

 AS3 10

 extended (profile) fields 82

 external ID 82

 external ID type 82

 IData object 82

 Integration Server 12

 IS document type 82

 MDNs (message disposition notifications) 11

 message disposition notifications (MDNs) 11

 pipeline 83

 process model 83

 process run time 83

 processing rules 83

 profiles 83

 TN document types 83

 TPA 38

 Trading Partner Agreement (TPA) 38

 WmEDI package 12

 WmEDIforTN package 12

delivering

 EDIINT documents 64

 MDN asynchronously 11, 16

 MDN synchronously 11, 20

 wm.EDIINT.delivery:delivery Document service 20

Delivery Method tab of profile, for EDIINT 31

deliveryDocument service 52

DES encryption algorithm 32

diagram of

 EDIINT Module and Integration Server architecture 12

 EDIINT Module and the webMethods Architecture 12

 inbound EDIINT document processing 15

 outbound EDIINT processing 18

 processing inbound MDNs 17

 sample process model that waits sends an EDIINT document and receives the corresponding MDN 25

 sending outbound EDIINT documents 19

 sending outbound MDN 21

digital signature

 outbound EDIINT documents 34

 verifying for EDIINT document 34

disabling

 port when EDIINT packed is disabled 35

 Trading Networks payload processing 46

document attributes

 copying 39

 EDIINT MDN Disposition 50

 EDIINT MDN Original Message 50

 EDIINT MDN Received MIC 50

 EDIINT Message Digest 49

 EDIINT Message ID 49

 EDIINT Message Type 49

 EDIINT transport-level processing 49

 provided for EDIINT processing 49

document recognition, wm.EDIINT.rules:processMsg service 22

document types

See IS document types and TN document types

documentation

 additional 8

 conventions used 7

 feedback 7

documents

 EDIINT, *See* EDIINT documents

 processing rule for inbound 16, 22

Drummond Group 11

E

EDI documents

- EDIINT payload 17
- payload, business-level processing 58
- sending to EDIINT 18

EDI Module MDNs

- inbound processing 17
- outbound processing 20

EDI over the Internet, *See* EDIINT

EDIINT 48

- built-in services 13
- content types 54
- defined 10
- delivery method 31
- digitally sign outbound documents 34
- document attributes 49
- documents, *See* EDIINT documents 32
- EDIINT Process
 - MDN Message processing rule 18
 - Message processing rule 16, 22
- EDIINT Send
 - MDN Message processing rule 22
 - Message processing rule 20
- exchange protocol 13
- Execute a Service action 16
- extended profile fields 32, 50
- features supported by EDIINT Module 11
- field group 50
- FTPS, configuring 28
- HTTP, configuring 28
- HTTPS, configuring 28
- including information in profiles 28
- interoperability testing 11
- MDN TN document type 18, 20, 22
- MDNs, *See* MDNs
- messages, *See* EDIINT messages
- MIME messages 19
- outbound processing 18
- packaging a document for transport 18
- payload, processing of 14
- processing rules for transport-level processing 51
- processing rules, *See* processing rules
- profile fields 29
- receipts, *See* MDNs
- S/MIME messages 19
- security information 34

- sending EDI documents to 18
- sending non-EDI documents to 18
- services invoked by processing rules 52
- services to package and transport documents 18
- SMTP, configuring 35
- Submit payload to TN check box 46
- TN document types 13, 22, 49
- transport-level processing 50, 56, 64
- User Process Payload Service field 47
- wm.EDIINT:receive service 16, 22, 56

EDIINT documents

- acceptable versions (AS1, AS2, AS3) 10
 - attributes 49
 - content type 14
 - conversation ID 25
 - created by wm.EDIINT:send service 19
 - cryptographic permutations 11
 - defined 10
 - delivering outbound, prerequisites 64
 - EDIINT Send Message processing rule 20
 - encrypting outbound 34
 - Execute a Service action 20
 - inbound
 - EDIINT Process Message processing rule 16, 22
 - flow 16, 18, 22
 - processing 13
 - User Status 68
 - outbound 19, 25
 - encryption options 32, 50
 - processing 18
 - User Status 69
 - packaging for transport 18
 - payload 15
 - posted by HTTP, example of 57
 - processing inbound documents, prerequisites 56
 - processing rule for inbound 16
 - processing rule for outbound 20
 - receiving encrypted 34
 - S/MIME type 59
 - sample process model of outbound 25
 - setting S/MIME type for outbound document 64
 - setting up to process payloads 58
 - verifying digital signature 34
 - viewing on Transaction Analysis screen 68
- EDIINT ID Match option 31

EDIINT MDN

- See also* MDNs
 - Disposition document attributes 50
 - entry point 60
 - inbound User Status 69
 - Original Message ID document attributes 50
 - outbound User Status 69
 - posted by HTTP, example of 61
 - processing inbound MDN, prerequisites 60
 - Received MIC document attributes 50
 - TN document type 49
 - viewing on Transaction Analysis screen 68
- EDIINT Message Digest document attribute 49
- EDIINT Message ID document attribute 49
- EDIINT Message Type document attribute 49
- EDIINT messages
- constructing outbound 77
 - receiving 74, 76
 - retrieving AS3 messages 76
- EDIINT Module
- architecture of 12
 - conditions to ignore payload settings 47
 - content handler installed with 14
 - EDIINT TN document types 16
 - overview 10
 - Trading Networks objects 48
 - wM.EDIINT:send service 64
- EDIINT Process MDN Message processing rule 51
- EDIINT Process Message processing rule 51, 57
- sender requesting MDN 16
- EDIINT recognizer, setting user status 16, 18, 20
- EDIINT Send MDN Message processing rule 51, 58
- EDIINT Send Message processing rule 51
- EDIINT TN document type 16
- EDIINT transport, sample process model 25
- EDIINTAS3 38
- Agreement ID 40
 - IS document type 40
 - input parameters 41
 - output parameters 44
- e-mail, EDIINT exchange 35
- enabling Trading Networks payload processing 46

encrypted

- EDIINT documents 34
 - S/MIME type 33, 59
- encrypting outbound EDIINT documents 34
- encryption
- EDIINT documents
 - outbound choices 32
 - specifying outbound 32
 - EDIINT messages 11
 - types for trading partners 11
 - wM.EDIINT:send service 19
- Encryption Algorithm extended profile field 32, 50
- entry point for inbound EDIINT MDNs 60
- errors
- and not invoking wM.EDIINT:send service directly from
 - Execute a Service processing action 65
 - payload 59
 - ProcessMDNMsg ERROR 69
 - ProcessMsg ERROR 68
 - SendMDNMsg ERROR 69
 - SendMsg ERROR 69
- example of
- EDIINT document posted by HTTP 57
 - EDIINT MDN posted by HTTP 61
- Execute a Service processing action 16, 18, 20, 22
- not invoking wM.EDIINT:send service directly from 65
- extended profile fields
- and EDIINT 13
 - defined 82
 - EDIINT field group 50
 - EDIINT, AS1MDNURL 50
 - EDIINT, AS2MDNURL 50
 - EDIINT, Encryption Algorithm 50
 - EDIINT, S/MIME Type 50
 - for EDIINT 32
- external ID 82
- external ID types
- defined 82
 - EDIINT ID Match option 31
 - EDIINT, specifying how to match 29
 - specifying 29

F

- false setting, `wm.EDIINT.submit Payload` configuration property 46
- FTP user root directory 37
- FTPS port, adding for EDIINT 28
- FTPUserName extended profile field for outbound AS3 messages and MDNs 32

H**HTTP**

- example of EDIINT document 57
- example of EDIINT MDN 61
- sending MDN 20
- HTTP port, adding for EDIINT 28
- HTTPS port, adding for EDIINT 28

I

- IData object defined 82
- IETF, and EDIINT 10
- IMAP protocol 35
- inbound EDI documents requesting MDN 20
- inbound EDIINT documents 14
- inbound EDIINT message, configuring SMTP settings 35
- inbound flow, EDIINT messages 16, 18, 22
- inbound processing
 - clients to send EDI documents using EDIINT exchange protocol 13
 - EDIINT documents
 - MDNs 17
- input variables
 - `contentType` 47
 - stream 47
 - `wm.EDIINT.TPA EDIINTAS3` document type 41
 - `wm.EDIINT:receive` service 74
 - `wm.EDIINT:retrieveAS3Message` service 76
 - `wm.EDIINT:send` service 19, 77
- Integration Server
 - and EDIINT Module architecture 12
 - and EDIINT Module, diagram 12
 - defined 12
- Internet Engineering Task Force (IETF) 10
- interoperability testing 11

invoking

- `wm.EDIINT.delivery:deliveryDocument` service 22
- `wm.EDIINT.rules`
 - `deliveryDocument` service 20
 - `processMDN` service 18
 - `processMsg` service 16, 22
 - `processPayload` service 17
 - `sendMsg` service 20
- `wm.EDIINT.rules:sendMDN` service 22
- `wm.EDIINT:send` service 19, 25, 64

IS document type

- defined 82
- specifying for input parametersAS3 41
- specifying output parameters for AS3 44

K

- key length 11

L**logic**

- business-level on EDIINT document 17
- to perform transport-level processing for inbound EDIINT documents 14, 19

M

- MD5 algorithm usage 11

MDNs

- and `wm.EDIINT.rules:processMsg` service 57
- asynchronous 11, 16, 22
- automatically sending outbound 20
- conversation ID assigned 25
- creating with `wm.EDIINT.rules:processMsg` service 22
- defined 11
- delivering synchronously or asynchronously 58
- EDIINT MDN TN document type 18, 20
- EDIINT, *See also* EDIINT MDN
- Execute a Service action 18, 22
- HTTP connection of inbound EDIINT document 20
- inbound
 - AS1 address 32, 50
 - AS2 address 32, 50
 - document 20
 - EDIINT Process MDN Message processing rule 18, 22
 - processing 17
 - `wm.EDIINT:receive` service 18
- outbound 20

- processing rule for inbound 18, 22
- receipt exchange 10
- receipt of EDIINT document 57
- receiving inbound via SMTP 35
- sample process model for outbound EDIINT documents 25
- sender
 - not requesting 16
 - requesting 16, 20
- sending outbound via SMTP 36
- separate transaction 20
- signed 16
- SMTP, configuring 35
- synchronous 11, 20, 22
- synchronous and asynchronous 32
- unsigned 16
- wm.EDIINT.delivery:deliveryDocument service 22
- wm.EDIINT.rules
 - processMDN service 18
 - processMsg service 16, 22
 - sendMDN service 22
- message disposition notifications 11
- messages
 - EDIINT MIME 19
 - EDIINT S/MIME 19
- MIME
 - messages 19
 - package 16
- modifying profile to include EDIINT information 28

N

- new document attribute 39
- non-EDI documents
 - EDIINT payload 17
 - payload, business-level processing 58
 - sending to EDIINT 18

O

- outbound choices for EDIINT message encryption 11
- outbound EDIINT message, configuring SMTP settings 36
- outbound processing
 - EDIINT 18
 - EDIINT documents 18
 - MDNs, sending 20
 - wm.EDIINT:send service 19

P

- package a document for transport using EDIINT 18
- packages
 - EDI Module 12
 - WmEDI 12
 - WmEDIforTN 12
- payload
 - business-level logic 58
 - business-level process for EDIINT document 50
 - business-level processing 56
 - conditions EDIINT Module ignores settings 47
 - EDI document 58
 - non-EDI document 58
 - processing by Trading Networks 46
 - S/MIME type affecting processing 59
 - submitting 46
 - Trading Networks to process 46
- payload of EDIINT document
 - business-level logic 17
 - EDI document 17
 - in sample process model 25
 - non-EDI document 17
 - sending to Trading Networks 15, 17
- performing business-level processing on payloads of EDIINT documents 58
- pipeline, defined 83
- plain S/MIME type 33, 59
- POP3 protocol 35
- port
 - adding for FTPS 28
 - adding for HTTP 28
 - adding for HTTPS 28
 - adding for SMTP 35
 - for MDNs via HTTP 32
 - webMethods/Email 35
- prerequisites for
 - delivering outbound EDIINT documents 64
 - processing inbound EDIINT documents 56
 - processing inbound EDIINT MDNs 60
- Primary E-mail delivery method for EDIINT AS1 31
- Primary HTTP delivery method for EDIINT AS2 31
- Primary HTTPS delivery method for EDIINT AS2 31

- process model
 - defined 83
 - sample outbound EDIINT documents and corresponding MDN 25
 - process run time, defined 83
 - processing
 - business level on payload of EDIINT document 50
 - business-level for inbound EDIINT documents 15
 - EDIINT transport-level 64
 - inbound
 - EDIINT documents 15, 56
 - prerequisites 56
 - EDIINT MDNs, prerequisites 60
 - outbound EDIINT documents, prerequisites 64
 - payloads 46
 - rules, *See* processing rules
 - transport-level for
 - inbound EDIINT documents 14
 - outbound EDIINT documents 19
 - processing actions, Execute a Service 18
 - processing rules
 - defined 83
 - EDIINT
 - and user status 16, 20
 - Process MDN Message 18
 - Process Message 16, 22, 51, 57
 - Send MDN Message 22, 51
 - Send MDN Message service 58
 - Send Message 20, 51
 - services 52
 - transport-level processing 51
 - Execute a Service action 16, 18, 20, 22
 - inbound
 - EDIINT documents 16, 22
 - MDNs 18, 22
 - MDNs 16
 - and user status 18
 - outbound EDIINT documents 20
 - payload of non-EDI document 58
 - sender requesting MDNs 16
 - user status
 - EDIINT 16
 - EDIINT documents 20
 - MDN 18
 - wm.EDIINT.rules:processMsg service 51
 - processMDN service 52
 - ProcessMDNMsg DONE user status 69
 - ProcessMDNMsg ERROR user status 69
 - ProcessMDNMsg IGNORED user status 69
 - ProcessMDNMsg user status 18, 69
 - ProcessMsg ERROR user status 68
 - ProcessMsg PAYLOAD user status 68
 - processMsg service 52
 - ProcessMsg user status 16, 68
 - processPayload service 52
 - profiles
 - defined 83
 - EDIINT information 28
 - program code conventions in this document 7
 - protocol, EDIINT 10
- ## R
- RC2 128 (128 bits) encryption algorithm 32
 - RC2 40 (40 bits) encryption algorithm 32
 - RC2 64 (64 bits) encryption algorithm 32
 - receive service 74
 - receiving
 - EDIINT messages 74, 76
 - encrypted EDIINT documents 34
 - inbound EDIINT message via SMTP 35
 - inbound MDN via SMTP 35
 - retrieveAS3Message service 76
- ## S
- S/MIME
 - EDIINT message 19
 - package 16
 - setting for outbound EDIINT document 64
 - version 2 10, 11
 - S/MIME Type 59
 - affecting payload processing 59
 - encrypted 59
 - extended profile field 33, 50, 58
 - plain 59
 - type partners use for payloads sent and received 50
 - samples
 - process model of outbound EDIINT documents and corresponding MDN 25
 - Secondary E-mail delivery method for EDIINT AS1 31
 - Secondary HTTP delivery method for EDIINT AS2 31
 - Secondary HTTPS delivery method for EDIINT AS2 31

- Secure/Multipurpose Internet Mail Extensions 11
- Security tab profile, for EDIINT 34
- send service, usage 74, 76, 77
- sender
 - not requesting MDN 16
 - requesting MDN 16, 20
 - asynchronous 22
 - synchronous 22
- sending
 - documents to EDIINT 18
 - EDIINT documents 64
 - EDIINT MDNs outbound 20
 - MDN 20
 - outbound
 - EDIINT documents 19
 - EDIINT message via SMTP 36
 - MDN via SMTP 36
 - wm.EDIINT:send service 19
- sendMDN service 52
- SendMDNMsg ERROR user status 69
- SendMDNMsg user status 69
- SendMsg DONE user status 69
- SendMsg ERROR user status 69
- sendMsg service 52
- SendMsg user status 20, 22, 69
- SendMsg WAITMDN user status 69
- services
 - packaging and transporting a document, using EDIINT 18
 - processing rules for EDIINT documents 52
 - sendMDN service 52
 - sendMsg service 52
 - wm.EDIINT
 - receive service 14, 16, 18, 22
 - send service 19
 - wm.EDIINT.delivery:deliveryDocument service 20, 22
 - wm.EDIINT.rules
 - processMDN service 18
 - processMsg service 16, 17, 22, 51, 57
 - processPayload service 17
 - sendMDN service 22, 58
 - sendMsg service 20
 - wm.EDIINT:receive service 60
 - wm.EDIINT:send service 64
 - WmEDIsample package, location of 7
- setting up processing payloads from EDIINT documents 58
- settings, S/MIME type for outbound EDIINT document 64

- SHA-1 hash algorithm 11
- Sign sub-tab, for EDIINT profile 34
- signed MDN 16
- signed S/MIME type 33, 59
- signedAndEncrypted S/MIME type 33, 59
- SMTP method, configuring settings to enable EDIINT message exchange 35
- SMTP port, adding for EDIINT 35
- SMTP Server field 36
- stream input variable 47
- Submit payload to TN check box 46
- synchronous MDN 11, 20, 22, 58

T

- TN document types
 - defined 83
 - EDIINT 49
 - EDIINT MDN 49
 - EDIINT MDN TN 22
 - EDIINT TN 16, 18, 20
 - for EDIINT documents 13
 - MDNs 13
 - payload of non-EDI document 58
 - provided for EDIINT processing 49
 - wm.EDIINT:receive service 16, 18
 - wm.EDIINT:send service 20
- TPA 38
 - defining 38
 - EDIINTAS3 38
 - list of fields 39
 - new with empty fields 39
- Trading Networks
 - adding business-level processing for EDIINT document payloads 14
 - and EDIINT Module architecture 12
 - EDIINT
 - Process MDN Message processing rule 18
 - Process Message processing rule 16, 22
 - Send MDN Message processing rule 22
 - Send Message processing rule 20
 - objects provided for EDIINT 48
 - processing payloads 46
 - TPAs 38
 - wm.EDIINT.rules:processMsg service 22
 - wm.EDIINT:receive service 16, 18, 22
 - wm.EDIINT:send service 20

- TPA
See Trading Partner Agreements
- Trading Partner Agreement (TPA), defined 38
- transport-level processing
 logic for inbound EDIINT documents 14
 logic for outbound EDIINT documents 19
 processing rules 51
- TripleDES encryption algorithm 32
- troubleshooting information 7
- true setting, `wm.EDIINT.submit Payload` property 46
- typographical conventions in this document 7
- U**
- Uniform Code Council 11
- unsigned MDN 16
- User Process Payload Service field 47
- user status
 EDIINT 16
 EDIINT document 20
 inbound EDIINT documents 68
 inbound EDIINT MDNs 69
 MDN 18, 22
 outbound EDIINT documents 69
 outbound EDIINT MDNs 69
 ProcessMDNMessage 18
 ProcessMessage 16
 SendMsg 20, 22
- User Status system attribute, viewing EDIINT documents and MDNs 68
- `userFtpRoot` directory 37
- V**
- Value (external ID) profile field for EDIINT 29
- viewing EDIINT documents and MDNs 68
- W**
- `webMethods EDIINT Module`, *See* EDIINT Module
- `webMethods/Email port EDIINT SMTP` settings 35
- `wm.EDIINT.delivery:deliveryDocument` service 22
 delivering EDIINT document to trading partner 20
- `wm.EDIINT.rules`
 folder 52
 processMsg service 51
 outbound MDNs for EDIINT document 57
 sendMDN service 58
- `wm.EDIINT.rules:deliveryDocument` service 20
- `wm.EDIINT.rules:processMDN` service 18
- `wm.EDIINT.rules:processMsg` service 16
 invoking `wm.EDIINT.rules:processPayload` service 17
- MDNs
 creating 22
 sender requesting 16
 opening MIME or S/MIME package 16
 Trading Networks 22
- `wm.EDIINT.rules:processPayload` service 17
- `wm.EDIINT.rules:sendMDN` service 22
- `wm.EDIINT.rules:sendMsg` service 20
- `wm.EDIINT:receive` service 14, 16, 18, 22, 74
 accepting inbound MDN 18
 and TN document types 18
 input variables 74
 passing EDIINT document to Trading Networks 22
 processing inbound EDIINT documents 56
 processing inbound EDIINT MDNs 60
 Trading Networks 18
- `wm.EDIINT:retrieveAS3Message` service 76
 input variables 76
- `wm.EDIINT:send` service 20, 74, 76, 77
 and TN document types 20
 creates EDIINT document 19
 delivering outbound documents 64
 in sample process model 25
 input variables 19, 77
 packaging document and sending it using EDIINT 54
 sending document to EDI module 19
 Trading Networks 20
- `WmEDI` package, defined 12
- `WmEDIforTN` package, defined 12
- `WmEDIINT` port for SMTP 35
- `WmRoot` port for SMTP 35

