

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

[illegible]

Table of Contents

About this Guide.....	7
Deprecation of webMethods Developer.....	7
Document Titles.....	7
Document Conventions.....	11
Online Information.....	12
Concepts.....	13
What Is webMethods Module for HIPAA?.....	14
webMethods Module for HIPAA Features.....	14
HIPAA Transactions that Module for HIPAA Supports.....	15
HIPAA Validation Levels.....	18
webMethods Module for HIPAA Packages.....	19
IS Document Types.....	19
webMethods Module for HIPAA Architecture.....	20
Process Overview.....	23
Sender-Side Processing.....	23
Receiver-Side Processing.....	25
Installing webMethods Module for HIPAA.....	29
Overview.....	30
Requirements.....	30
The Integration Server Home Directory.....	30
Installing Module for HIPAA.....	30
Installing the Module for HIPAA Samples Package.....	31
Uninstalling Module for HIPAA.....	31
Configuring the webMethods Module for HIPAA.....	33
Overview.....	34
Step 1: Install TN Document Types for HIPAA Transactions.....	34
Step 2: Define Profiles for Trading Partners.....	35
External ID Types in Profiles and EDI ID Qualifiers.....	35
Step 3: Configure Large Document Handling.....	36
Configuring Module for EDI.....	36
Configuring Trading Networks.....	36
Step 4: Create EDI Trading Partner Agreements.....	37
Step 5: Create HIPAA Trading Partner Agreements.....	38
HIPAA Parameters.....	38
Creating Clients that Send HIPAA Messages.....	43
Overview.....	44
Content Type to Use.....	45
Service the Client Invokes.....	45

How pub.esd.hipaa.receive Handles TA1 Transactions.....	45
How pub.esd.hipaa.receive Handles Other HIPAA Transactions.....	47
Code Source Management.....	49
Overview.....	50
Creating Database Tables.....	50
Importing Code Sources.....	51
Method 1.....	51
Method 2.....	52
Importing Code Sources into Module for HIPAA.....	53
Viewing a Codelist.....	54
Searching within a Code Source.....	54
Adding codes.....	54
Deleting codes.....	55
Dropping Database Tables.....	55
List of Code Sources.....	56
HIPAA Acknowledgments.....	59
Overview.....	60
Technical Acknowledgment.....	60
TA1 Code.....	61
Functional Acknowledgment (997).....	62
Functional Acknowledgment Error Codes.....	64
Error codes for AK304.....	65
Error codes for AK403.....	65
Error codes for AK501.....	66
Error codes for AK502 through AK506.....	66
Error codes for AK901.....	67
Error codes for AK905 through AK909.....	67
Implementation Acknowledgment (999).....	68
HIPAA CORE Compliance.....	69
CORE Phase Connectivity.....	70
Support for CORE Phase I and II.....	70
Processing HIPAA Transactions.....	71
Processing HIPAA Messages Sent to Integration Server.....	73
Overview.....	74
Before You Can Process Inbound HIPAA Messages.....	74
Using Processing Rules to Process Inbound HIPAA Messages.....	74
Defining a Processing Rule for an Envelope Document.....	75
Using Services to Process Inbound HIPAA Messages.....	76
Processing HIPAA Acknowledgments.....	79
Overview.....	80
Before You Can Process Inbound HIPAA Acknowledgments.....	80
Defining Processing Rules for Inbound HIPAA Acknowledgments.....	80

Defining a Processing Rule for a TA1 Technical Acknowledgment.....	80
Defining a Processing Rule for a 997 Functional Acknowledgment.....	81
WmHIPAA Services.....	83
pub.estd.hipaa:validate.....	84
pub.estd.hipaa:convertHipaaToIData.....	86
pub.estd.hipaa:convertIDataToHipaa.....	86
pub.estd.hipaa:normalizeName.....	87
pub.estd.hipaa.core:send.....	88
pub.estd.hipaa:receive.....	90
pub.estd.hipaa:recognizeAcknowledgements.....	91
pub.estd.hipaa:recognizeTA1.....	91
wm.estd.hipaa.core:receive.....	92
wm.estd.hipaa.core:receive_security_auth.....	92
wm.estd.hipaa.core.services:realtimeHttpReceive.....	93
pub.estd.hipaa:attachResponsePayload.....	93



About this Guide

This guide describes how to install, configure, and use webMethods Module for HIPAA to receive, parse, and validate all of the mandated HIPAA transactions and to respond with the appropriate acknowledgments.

To use this guide effectively, you should be familiar with:

- webMethods Integration Server and Integration Server Administrator, and understand the concepts and procedures described in the Integration Server administration guide.
- webMethods Trading Networks and webMethods Module for EDI, and understand the concepts and procedures described in the various Trading Networks and Module for EDI guides.
- Software AG Designer, and understand the concepts and procedures described in the Designer online help.
- My webMethods Server and its interface, My webMethods, and understand the concepts and procedures described in the My webMethods administration guide and *Working with My webMethods*.
- Have a basic knowledge of HIPAA standards and transactions as well as HIPAA terminology.

Deprecation of webMethods Developer

webMethods Developer is deprecated and does not support all the features of webMethods Integration Server 8.2. SoftwareAG recommends the use of SoftwareAG Designer for service development.

Document Titles

Some webMethods document titles have changed during product releases. The following table will help you locate the correct document for a release on the Software AG Documentation Web site or the Empower Product Support Web site.

Documentation	7.x	8.0	8.0 SP1	8.2	9.0 and later
---------------	-----	-----	------------	-----	---------------------

Designer Process Development online help

Documentation	7.x	8.0	8.0 SP1	8.2	9.0 and later
<i>webMethods BPM Process Development Help</i>				x	x
<i>webMethods Designer BPM Process Development Help</i>		x	x		
<i>webMethods Designer Process Development Help</i>	x				
Designer Service Development online help					
<i>webMethods Service Development Help</i>				x	x
<i>webMethods Designer Service Development Help</i>	x	x	x		
Developer user's guide					
<i>Developing Integration Solutions: webMethods Developer User's Guide</i>			x	x	
<i>webMethods Developer User's Guide</i>	x	x			
Integration Server administration guide					
<i>webMethods Integration Server Administrator's Guide</i>	x	x			x
<i>Administering webMethods Integration Server</i>			x	x	
Integration Server built-in services reference guide					
<i>webMethods Integration Server Built-In Services Reference</i>	x	x	x	x	x
Integration Server clustering guide					
<i>webMethods Integration Server Clustering Guide</i>	x	x	x	x	x

Documentation	7.x	8.0	8.0 SP1	8.2	9.0 and later
Integration Server publish-subscribe developer's guide					
<i>Publish-Subscribe Developer's Guide</i>	x	x	x	x	x
My webMethods administration guide					
<i>Administering My webMethods Server</i>			x	x	x
<i>My webMethods Server Administrator's Guide</i>	x	x			
Optimize administration guide					
<i>Administering webMethods Optimize</i>			x	x	x
<i>webMethods Optimize Administrator's Guide</i>	x	x			
Optimize user's guide					
<i>webMethods Optimize User's Guide</i>	x	x			x
<i>Optimizing BPM and System Resources with BAM: webMethods Optimize User's Guide</i>			x	x	
Process Engine administration guide					
<i>Administering webMethods Process Engine</i>		x	x	x	x
<i>webMethods Process Engine User's Guide</i>	x				
Trading Networks administration guide					
<i>webMethods Trading Networks Administrator's Guide</i>	x	x			x
<i>Building B2B Integrations: webMethods Trading Networks Administrator's Guide</i>			x	x	
Trading Networks built-in services reference guide					

Documentation	7.x	8.0	8.0 SP1	8.2	9.0 and later
<i>webMethods Trading Networks Built-In Services Reference</i>	x	x	x	x	x
Trading Networks concepts guide					
<i>webMethods Trading Networks Administrator's Guide and webMethods Trading Networks User's Guide</i>					x
<i>Understanding webMethods B2B: webMethods Trading Networks Concepts Guide</i>			x	x	
<i>webMethods Trading Networks Concepts Guide</i>	x	x			
Trading Networks user's guide					
<i>webMethods Trading Networks User's Guide</i>	x	x			x
<i>Managing B2B Integrations: webMethods Trading Networks User's Guide</i>			x	x	
webMethods installation guide					
<i>Installing webMethods Products and Using the Software AG Installer</i>				x	x
<i>Software AG Installation Guide</i>			x		
<i>webMethods Installation Guide</i>	x	x			
webMethods logging guide					
<i>webMethods Audit Logging Guide</i>			x	x	x
<i>webMethods Audit Guide</i>	x	x			
webMethods upgrade guide					

Documentation	7.x	8.0	8.0 SP1	8.2	9.0 and later
<i>Upgrading webMethods Products</i>				x	x
<i>webMethods Upgrade Guide</i>	x	x	x		

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention <i>folder.subfolder:service</i> .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
<i>Italic</i>	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

1 Concepts

■ What Is webMethods Module for HIPAA?	14
■ webMethods Module for HIPAA Features	14
■ HIPAA Transactions that Module for HIPAA Supports	15
■ HIPAA Validation Levels	18
■ webMethods Module for HIPAA Packages	19
■ IS Document Types	19
■ webMethods Module for HIPAA Architecture	20
■ Process Overview	23

What Is webMethods Module for HIPAA?

webMethods Module for HIPAA is a comprehensive and highly scalable solution that allows your organization to implement the HIPAA 5010 and 5010A standards. This webMethods component provides out-of-the-box functionality to validate all of the mandated HIPAA transactions as well as generate the appropriate acknowledgments. Module for HIPAA streamlines health care industry transactions by providing a solution for rapid and seamless integration of providers, payers, routers, and sponsors.

Note: Module for HIPAA runs on top of webMethods Module for EDI. When you install Module for HIPAA, it changes the behavior of some of the functions of webMethods Module for EDI to meet HIPAA standards. As a result, you must override EDI ID qualifiers to use those IDs in Module for HIPAA (for example, in EDI the ID qualifier code 30 is "ISO 6523" but in HIPAA it is "Federal Tax ID"). For more information about adding and overriding EDI ID qualifiers, see *webMethods Module for EDI Installation and User's Guide*. If you plan to process both HIPAA-related and non-HIPAA related EDI documents, Software AG recommends that you set up the processing on different machines. If you prefer to use the same machine, be careful when setting up and testing to ensure that the processing for HIPAA and non-HIPAA related documents functions as anticipated.

webMethods Module for HIPAA Features

Module for HIPAA runs on webMethods Integration Server, webMethods Trading Networks, and webMethods Module for EDI. Module for HIPAA provides an easy, secure, and reliable solution for seamless integration with external systems and trading partners.

Module for HIPAA provides support for the following.

- **HIPAA transactions.** This enables you to quickly implement production solutions for automating the many interactions between you and your trading partners. For a list of transactions that Module for HIPAA supports, see ["HIPAA Transactions that Module for HIPAA Supports" on page 15](#).
- **Validation.** Module for HIPAA provides out-of-the-box validation through Levels 1, 2, 3, and 5 as defined by WEDI-SNIP certification guidelines. In addition, you can split HIPAA data into separate files after validation so that the valid data can be reused. You can also customize messages to send to partners based on validation results. For a description of validation levels, see ["HIPAA Validation Levels" on page 18](#).
- **HIPAA 5010 and 5010A standards.** Module for HIPAA supports HIPAA 5010 and 5010A standards out-of-the-box. With pre-packaged Integration Server and Trading Networks documents, you can easily transform an X12 document into an equivalent IS canonical document and vice versa.

- **Functional and technical acknowledgments.** Module for HIPAA fully supports HIPAA-defined success/failure notification of envelope errors using TA1 technical acknowledgments. Module for HIPAA also supports 997 functional acknowledgments to communicate the validation results. Module for HIPAA lets you create acknowledgments automatically with a validation service.
- **Code sets.** HIPAA transactions are validated for code sets.
- **Error reports.** Module for HIPAA generates detailed error reports in HTML and XML formats. You can send these reports in an email message to someone in your enterprise or to a trading partner.
- **Leveraging existing enterprise solutions.** Module for HIPAA allows you to accept information from other EDI-based systems to populate documents in HIPAA format.
- **Transaction logging and audit trails.** Module for HIPAA ensures the integrity of all trading partner transactions with automatic archival of transaction messages.
- **CORE phase I and II support.** Module for HIPAA fully supports CORE phase I and II rules and their implementations that conform with real time transactions. The module currently does not support processing of batch transactions.

HIPAA Transactions that Module for HIPAA Supports

Module for HIPAA supports the following HIPAA transactions and addenda:

Action	Enables users to...	Module for HIPAA supports versions...
270 Health Care Eligibility Benefit Inquiry	Send a Health Care Eligibility Inquiry, also known as 270 to the trading partner (generally an insurance company) to determine whether a patient is eligible for certain claim benefits.	<ul style="list-style-type: none"> ■ 005010X279 (standard) ■ 005010X279A1 (addendum)
271 Health Care Eligibility Benefit Response	Send a Health Care Eligibility Response, also known as 271, response to the trading partner stating the patient's eligibility. The response document contains details such as eligibility status,	<ul style="list-style-type: none"> ■ 005010X279 (standard) ■ 005010X279A1 (addendum)

Action	Enables users to...	Module for HIPAA supports versions...
	maximum benefits, in-plan/out of plan benefits, and co-payments.	
276 Health Care Claim Status Request	Send a Health Care Claim Status Request, also known as 276, to request the current status of a specified claim.	■ 005010X212 (standard)
277 Health Care Claim Status Response	Send a Health Care Claim Status Response, also known as 277, to the requestor with the current status of the adjudication process. If the request matches more than one claim in the payer's system, the response may include multiple claims.	■ 005010X212 (standard)
278 Health Care Services Request for Review and Response	Send a Health Care Services Request for Review and Response, also known as 278, to health care provider, payers, delegated UMO entities, and other providers.	■ 005010X217 (standard)
820 Payroll Deducted and Other Group Premium Payment for Insurance Products	Send a Payroll Deducted and Other Group Premium Payment for Insurance Products, also known as 820, to initiate a payment with the remittance detail needed by the premium receiver, or without the remittance	■ 005010X218 (standard)

Action	Enables users to...	Module for HIPAA supports versions...
	detail and send the remittance detail separately to the premium receiver.	
834 Benefit Enrollment and Maintenance	Send a Benefit Enrollment and Maintenance Request, also known as 834, to transfer enrollment information from the sponsor to a payer.	■ 005010X220 (standard)
835 Health Care Claim Payment/Advice	Send a Health Care Claim Payment/Advice, also known as 835, to make a payment, send an Explanation of Benefits (EOB) remittance advice, or make a payment and send an EOB remittance advice from a health care payer to a health care provider, either directly or through a Depository Financial Institution (DFI).	■ 005010X221 (standard) ■ 005010X221A1 (addendum)
837 Health Care Claims fall into three categories: Professional, Institutional, and Dental. Module for HIPAA supports all three categories		
837 Health Care Claim Professional	Send a Health Care Claim Professional, also known as 837, to the trading partner (generally an insurance company) to submit health care claim billing information, encounter information, or both, from providers of health care services.	■ 005010X222 (standard) ■ 005010X222A1 (addendum)

Action	Enables users to...	Module for HIPAA supports versions...
837 Health Care Claim Institutional	Send a Health Care Claim Institutional, also known as 837, to the trading partner (generally an insurance company) to submit health care claim billing information, encounter information, or both, from providers of health care services.	<ul style="list-style-type: none"> ■ 005010X223 (standard) ■ 005010X223A1 (addendum) ■ 005010X223A2 (addendum)
837 Health Care Claim Dental	Send a Health Care Claim Dental, also known as 837, to the trading partner (generally an insurance company) to submit health care claim billing information, encounter information, or both, from providers of health care services.	<ul style="list-style-type: none"> ■ 005010X224 (standard) ■ 005010X224A1 (addendum) ■ 005010X224A1 (addendum)

HIPAA Validation Levels

The HIPAA Implementation Guidelines specify six levels of message validation. Module for HIPAA supports Levels 1, 2, 3, and 5.

Level	Description
1	Integrity. Validates the syntactical integrity of the X12 EDI document. Validation at this level includes testing for valid segments, segment order, and element attributes.
2	Requirement. Validates the syntax rules meet the HIPAA Implementation Guidelines. Validation at this level includes testing for required repeat counts, used and unused codes, and required or inter-segmental data elements.

Level	Description
3	Balancing. Validates the transaction for balanced field totals, record or segment counts, financial balancing of claims or remittance advice, and balancing of summary fields.
5	Code Set. Validates that the transaction uses valid code set values as described in the HIPAA Implementation Guidelines (for example, CPT, NDC).

webMethods Module for HIPAA Packages

Module for HIPAA contains the following packages (set of services and related files) that you install on Integration Server.

Package	Description
WmHIPAA	Contains general functionality and serves as the main holding area for application user interfaces. This package also contains shared components including implementations of common utilities, common validation services, and transport services. For detailed information about the Module for HIPAA services, see "WmHIPAA Services" on page 83 .
WmHipaaCodeSource	Contains the Washington Publishing Company (WPC) standard Codelist which the module requires to validate various fields in a HIPAA message. For detailed information about code sources, see "Code Source Management" on page 49 .

IS Document Types

An IS document type contains a set of fields that define the structure and type of data in a document (IData object). Use IS document types to specify input or output parameters for a service or specification, build a document or document list field, and use as the blueprint for pipeline validation and document (IData object) validation.

Module for HIPAA provides IS document types (in the WmHIPAA package under the HIPAASchema folder) for 5010 HIPAA message types. Use these documents to map the incoming HIPAA messages to IS documents for external systems and vice versa.

The WmHIPAASample package contains samples to simulate these HIPAA message conversions. The following IS document types are provided in the WmHIPAA package for the transactions:

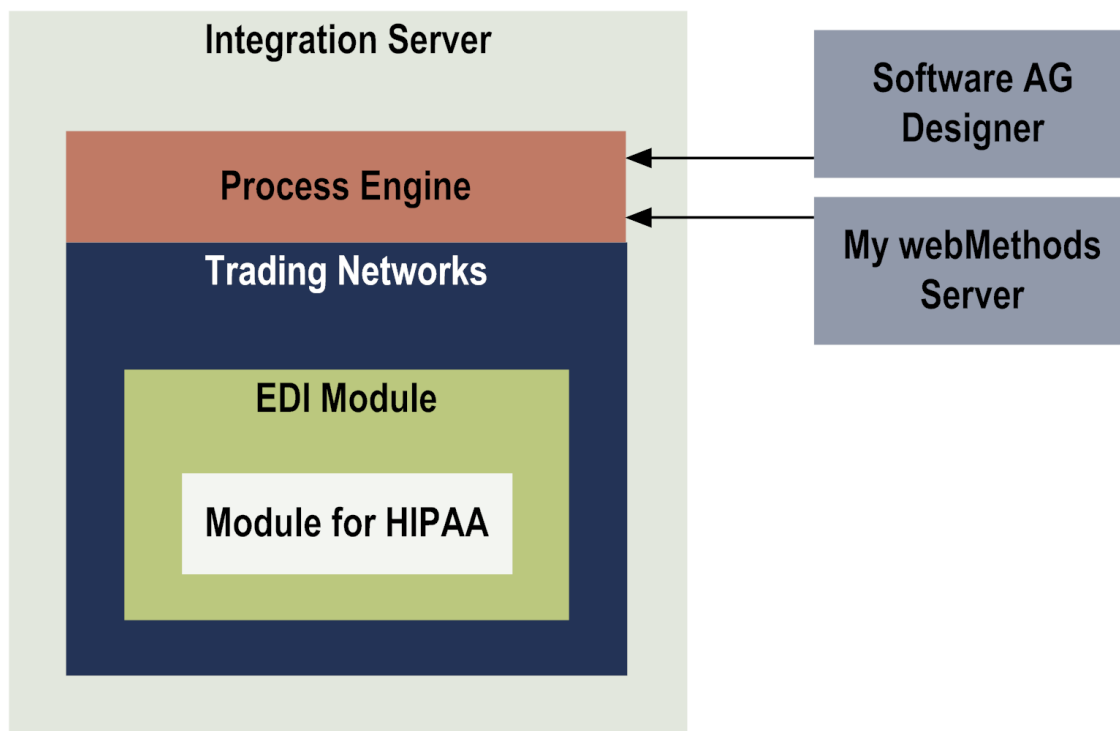
Transaction Type	IS Document Type
270-B1A1	HIPAA\Schema.X12.V5010.HS.X279:T270DT
271-A1	HIPAA\Schema.X12.V5010.HB.X279:T271DT
276-A1	HIPAA\Schema.X12.V5010.HR.X212:T276DT
277-A1	HIPAA\Schema.X12.V5010.HN.X212:T277DT
277-B3	HIPAA\Schema.X12.V5010.CA.X214:T277B3DT
278-A1	HIPAA\Schema.X12.V5010.HI.X217.Req:T278ReqDT
278-A3	HIPAA\Schema.X12.V5010.HI.X217.Res:T278ResDT
820-A1	HIPAA\Schema.X12.V5010.RA.X218:T820DT
834-A1A1	HIPAA\Schema.X12.V5010.BE.X220:T834DT
835-W1A1	HIPAA\Schema.X12.V5010.HP.X221:T835DT
837-Q1A1	HIPAA\Schema.X12.V5010.HC.X222:T837DT
837-Q3A2	HIPAA\Schema.X12.V5010.HC.X223:T837DT
837-Q2A2	HIPAA\Schema.X12.V5010.HC.X224:T837DT
999-A1	HIPAA\Schema.X12.V5010.IA.X231:T999DT

For more information about IS document types, see the Designer Service Development online help for your release. See “About this Guide” for specific document titles.

webMethods Module for HIPAA Architecture

The following diagram illustrates how Module for HIPAA fits into webMethods architecture.

Architecture and webMethods Components



Component	Description
Integration Server	Integration Server is the underlying foundation of webMethods architecture. It processes requests from and relays responses to an external system.
Trading Networks	Trading Networks enables your enterprise to link with other companies and exchanges to form a business-to-business trading network. For more information, see the Trading Networks administration guide for your release. See "About this Guide" for specific document titles.
Module for EDI	<p>Module for EDI enables your enterprise to receive and process EDI documents. To use Module for HIPAA, you use two of the Module for EDI packages:</p> <ul style="list-style-type: none"> ■ WmEDI package—contains the basic functionality that provides support for the EDI standard to webMethods architecture.

Component	Description				
	<ul style="list-style-type: none"> ■ WmEDIforTN package-allows the interaction between the WmEDI package and Trading Networks, which serves as a gateway for EDI document exchange. 				
Software AG Designer	<p>At design time, you can use the Service Development perspective of Software AG Designer to create, view, modify, and delete services and IS document types. Also use Designer to run services.</p> <p>You also use Designer to create process models that define the business processes (also known as a conversation) for your HIPAA implementation. When you generate the new process models, Designer creates the run-time elements (flow services and triggers) in Integration Server. Integration Server's process engine executes the business processes (conversations) at run-time.</p> <p>Also use process models to process HIPAA documents. For more information, see <i>webMethods Module for EDI Installation and User's Guide</i>, the Designer Service Development online help, and the Designer Process Development online help for your release. See "About this Guide" for specific document titles.</p>				
My webMethods Server	My webMethods Server is a run-time container for functions made available by webMethods applications, such as Integration Server, Trading Networks, and Module for HIPAA. For more information, see <i>Working with My webMethods</i> .				
Module for HIPAA	Module for HIPAA is a webMethods component that adds support for the HIPAA 5010 standard. The module contains two packages.				
	<table> <tr> <td>Module for HIPAA</td><td>Module for HIPAA can receive, parse, and validate a HIPAA message.</td></tr> <tr> <td>Module for HIPAA Code Source</td><td>Module for HIPAA Code Source contains the WPC standard Codelist to validate the various fields in a HIPAA message.</td></tr> </table>	Module for HIPAA	Module for HIPAA can receive, parse, and validate a HIPAA message.	Module for HIPAA Code Source	Module for HIPAA Code Source contains the WPC standard Codelist to validate the various fields in a HIPAA message.
Module for HIPAA	Module for HIPAA can receive, parse, and validate a HIPAA message.				
Module for HIPAA Code Source	Module for HIPAA Code Source contains the WPC standard Codelist to validate the various fields in a HIPAA message.				

Process Overview

To process HIPAA messages, use the facilities provided by:

- Module for HIPAA which provides the HIPAA-related services such as validate for all the mandated HIPAA transactions and responds with the appropriate acknowledgments.
- Trading Networks which handles the routing of messages to trading partners.

Additionally, you must add your own processing to do the following:

- **Send HIPAA messages to your trading partners.** If you are acting as a sender, you can use a Trading Networks delivery service to send a valid standard HIPAA message to your trading partner (acting as the receiver).
- **Send the appropriate acknowledgments to the HIPAA messages.** Module for HIPAA can be configured to send acknowledgments using TPA parameters. To send acknowledgments, use Trading Networks delivery features. For more information, see the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.
- **Process the HIPAA transactions to meet your specific needs.** For example, you might want to map the data from a 835 Health Care Claim Payment/Advice transaction to an external system document and send that document to the downstream system.

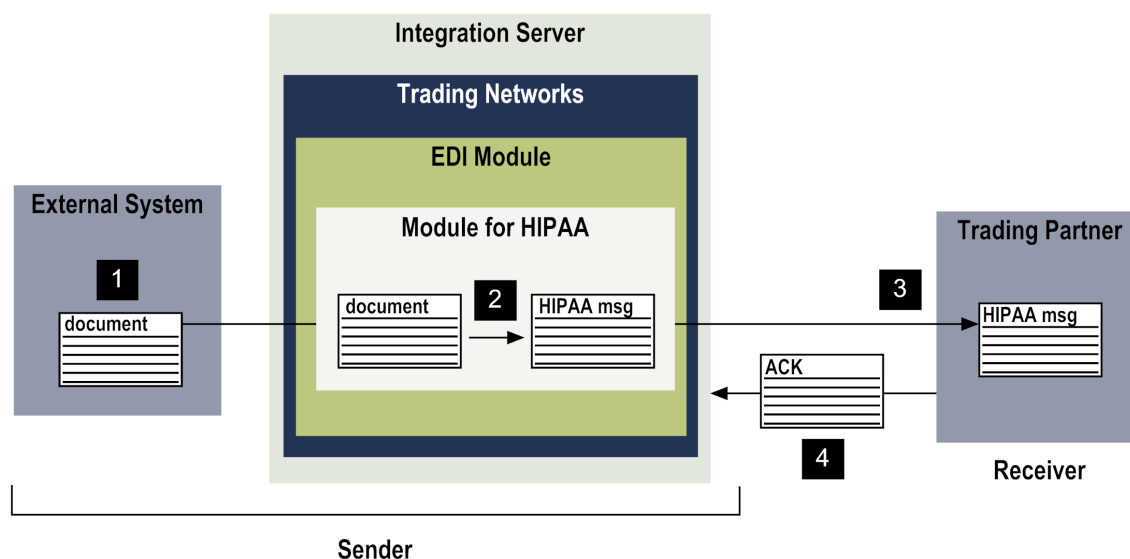
The Module for HIPAA sample illustrates how to map data to IS document types for processing by external systems.

To add your own processing, use either Trading Networks processing rules or Designer to define custom process models and business processes.

Sender-Side Processing

The sender forms a HIPAA message and sends it to a trading partner (that is, the receiver of the HIPAA message). The following diagram illustrates sender-side processing. For more information, see the table after the diagram.

Sender-Side Processing of HIPAA Messages



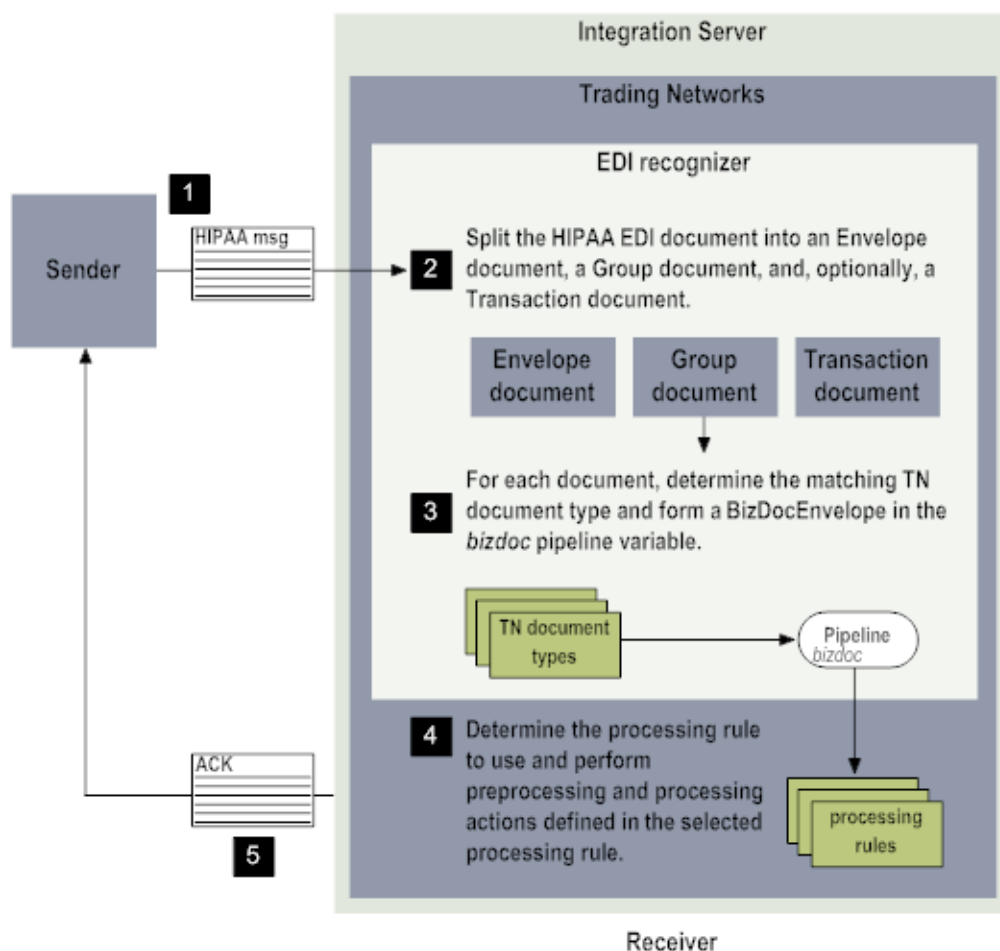
Step	Description
1	<p>The external system sends a document to Integration Server, which can be:</p> <ul style="list-style-type: none"> ■ In an internal format used by the external system ■ A valid HIPAA message
2	<p>The actions performed on Integration Server depend on the type of document sent. If the external system sends:</p> <ul style="list-style-type: none"> ■ Documents in an internal format, define logic on Integration Server to map the data from the internal format to a standard HIPAA message. ■ Valid HIPAA messages, Integration Server needs only to send the HIPAA message to the receiver. Use Trading Networks delivery features to send the HIPAA message. For more information, see the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.
3	<p>After a valid standard HIPAA message is available, use a Trading Networks delivery service to send the document to your trading partner (acting as the receiver). For more information, see the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.</p>

Step	Description
4	The trading partner (acting as the receiver) sends an acknowledgment to the sender, who processes it accordingly. For information about how to process acknowledgments, see "Processing HIPAA Acknowledgments" on page 79 .

Receiver-Side Processing

The following diagram illustrates receiver-side processing when using Trading Networks processing rules.

Receiver-Side Processing



Step	Description
1	Your trading partner creates a client that constructs a HIPAA standard message and sends the HIPAA message to your Integration Server.
2	<p>Integration Server receives the HIPAA message and sends it to Trading Networks for processing. Because the HIPAA message is an EDI document, Trading Networks passes the document to Module for EDI.</p> <p>The Module for EDI splits the HIPAA message into separate documents. Set a variable in an EDI trading partner agreement (EDITPA) to define the level of documents (Envelope, Group, or Transaction) that you want the HIPAA message split into. For HIPAA processing, you must split at least at the group level to form both Envelope and Group documents. For more information about EDITPAs and the EDITPA splitOption variable, see "Configuring the webMethods Module for HIPAA" on page 33 in this guide and also see <i>webMethods Module for EDI Installation and User's Guide</i>.</p>
3	<p>Module for HIPAA creates the following TN document types:</p> <ul style="list-style-type: none"> ■ 5010 270 ■ 5010 271 ■ 5010 276 ■ 5010 277 ■ 5010 278 ■ 5010 820 ■ 5010 834 ■ 5010 835 ■ 5010 837 ■ 5010 999

For each document (Envelope, Group, or Transaction) split from the original HIPAA message, the Module for EDI uses the TN document types to determine the type of document (for example, X12 Envelope, X12 Group, or X12 5010 835).

After recognizing the type of document using TN document types, the EDI recognizer forms a BizDocEnvelope for the EDI document. The BizDocEnvelope is in the bizdoc pipeline variable. A BizDocEnvelope contains the original document (Envelope, Group, or Transaction) and includes additional information that Trading Networks requires

Step	Description
	for routing and processing the document. In other words, the BizDocEnvelope represents a routable Trading Networks transaction.
4	After the BizDocEnvelope is formed, the document undergoes regular Trading Networks processing. Trading Networks determines which processing rule to execute on the document. For example, set up processing for the Envelope document to validate the envelope and generate TA1 technical acknowledgments and 997 functional acknowledgments, if appropriate. For more information about defining processing rules, see "Before You Can Process Inbound HIPAA Messages" on page 74 .
5	Acknowledgments (for example, 997 and TA1) can be returned to the sender using Trading Networks delivery features. For more information, see the Trading Networks administration guide for your release. See "About this Guide" for specific document titles.
Note:	When a received HIPAA transaction is split, the generated (Envelope, Group, and Transaction) document types are related and displayed in the My webMethods: Monitoring > Integration > B2B > Transactions > Related Documents column but the generated functional (997), implementation (999), and technical (TA1) acknowledgements are not related, by default.



2 Installing webMethods Module for HIPAA

■ Overview	30
■ Requirements	30
■ The Integration Server Home Directory	30
■ Installing Module for HIPAA	30
■ Installing the Module for HIPAA Samples Package	31
■ Uninstalling Module for HIPAA	31

Overview

This chapter explains how to install and uninstall webMethods Module for HIPAA 9.6. The instructions use the Software AG Installer and the Software AG Uninstaller wizards. For complete information about the wizards or other installation methods, or to install other webMethods products, see the webMethods installation guide for your release. See “About this Guide” for specific document titles.

Requirements

For a list of the operating systems and webMethods products required for the installation and operation of Module for HIPAA, see *webMethods eStandards Modules System Requirements*, available in the webMethods area of the Software AG Documentation website.

The Integration Server Home Directory

Beginning with Integration Server 9.6, you can create and run multiple Integration Server instances under a single installation directory. Each Integration Server instance has a home directory under *Integration Server_directory\instances\instance_name* that contains the packages, configuration files, log files, and updates for the instance.

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

This guide uses the *packages_directory* as the home directory in Integration Server classpaths. For Integration Server 9.6 and above, the *packages_directory* is *Integration Server_directory\instances\instance_name\packages_directory*. For Integration Server 9.5 and lower, the *packages_directory* is *Integration Server_directory\packages_directory*.

Installing Module for HIPAA

1. Download Software AG Installer from the [Empower Product Support Web site](#).
2. If you are installing the module on an existing Integration Server, shut down the Integration Server.
3. Start the Software AG Installer wizard.
4. Choose the webMethods release that includes the Integration Server on which you want to install the module.
5. Specify the installation directory to use (the default is Software AG).

- If you are installing on an existing Integration Server, specify the Software AG installation directory that contains the host Integration Server.
 - If you are installing both the host Integration Server and the module, specify the installation directory to use.
6. In the product selection list, select **eStandards > webMethods Module 9.6 for HIPAA**. The Installer automatically selects **Program Files** and **Code Source**. **Program Files** installs the WmHIPAA package and **Code Source** installs the WmHipaaCodeSource package which contains the default code sources bundled with the module.

Note: If you choose to not install code sources, clear the **Code Source** checkbox. You can install **Code Source** only when **Program Files** is selected.

7. Select any required products indicated in *webMethods eStandards Modules System Requirements*.

Installer installs the following components:

- webMethods Integration Server
 - webMethods Trading Networks
 - webMethods Module for HIPAA installed as WmHIPAA and WmHipaaCodeSource packages in the *Software AG_directory\Integration Server_directory\packages* directory.
- If Integration Server, Trading Networks, and Module for EDI are already installed from a previous installation, installer does not reinstall these products.

8. After installation completes, close Installer.
9. Start the Integration Server on which you installed webMethods Module 9.6 for HIPAA.

Installing the Module for HIPAA Samples Package

The Module for HIPAA samples package contains the sample services. The samples package is not installed with webMethods Module 9.6 for HIPAA. To download the WmHIPAASample package, go to the Code Sample section of the Software AG Developer Community for webMethods at <http://communities.softwareag.com/>.

Uninstalling Module for HIPAA

1. Shut down the Integration Server that hosts Module for HIPAA.
2. Start Software AG Uninstaller, as follows:

System	Instructions
Windows	In the Add or Remove Programs window, select the installation directory of the Integration Server on which Module for HIPAA is installed.

3. In the product selection list, select **eStandards > webMethods Module 9.6 for HIPAA**.
4. Restart the host Integration Server.
5. Uninstaller moves all webMethods Module 9.6 for HIPAA -related files that were installed into the *Integration Server_directory* \packages, or *Integration Server_directory* \instances\instance_name \packages directory. However, Uninstaller does not delete files that you created after you installed the module (for example, user-created or configuration files), nor does it delete the module directory structure. You can select the *Integration Server_directory* \packages, or *Integration Server_directory* \instances \instance_name \packages directory and delete the Module for HIPAA-related directory.

3

Configuring the webMethods Module for HIPAA

■ Overview	34
■ Step 1: Install TN Document Types for HIPAA Transactions	34
■ Step 2: Define Profiles for Trading Partners	35
■ Step 3: Configure Large Document Handling	36
■ Step 4: Create EDI Trading Partner Agreements	37
■ Step 5: Create HIPAA Trading Partner Agreements	38



Overview

This chapter describes how to set up the webMethods product suite so that you can send and receive HIPAA messages using the services in Module for HIPAA.

Important: The following procedure assumes that you have already installed Integration Server, Trading Networks, Module for EDI, and Module for HIPAA.

Step 1: Install TN Document Types for HIPAA Transactions

When Trading Networks receives a document, it uses the TN document types to determine the file type. This is referred to as *document recognition*. Trading Networks also uses the TN document type to determine which attributes to extract from the document. For more information about TN document types, see the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.

Note: When you install a TN document type, Module for EDI also installs the corresponding flat file schema for the EDI transaction set. For more information about flat file schemas used for EDI documents, see the Module for EDI documentation. For more information about flat file schemas in general, see *Flat File Schema Developer's Guide*.

Use Module for EDI to install TN document types for EDI documents into Trading Networks. You need to install the TN document types for:

- The types of EDI transactions that you plan to use.
- HIPAA acknowledgments 997, 999, and 277A.

To install TN EDI document types for HIPAA transactions

1. See *webMethods Module for EDI Installation and User's Guide* for complete information and the installation procedure on TN EDI document types and flat file schemas.
2. During the TN EDI document type installation, specify the following values:

For this field...	Specify...
Standard	X12
Version	5010
Transaction Set	Type of EDI document that corresponds to the TN document type that you want to install, for example, 835. Module for EDI automatically installs the TN document

For this field...	Specify...
	types for both the Envelope and Group (if not already installed). For example, the installation of a TN document type for an X12 transaction includes the X12 Envelope and X12 Group TN document types.

- Repeat steps 1 and 2 for each type of EDI document that you want to install.

Note: You do not need to install a TN document type for the TA1 technical acknowledgment document. The TA1 technical acknowledgment TN document type, X12 TA1 ACK, is automatically installed in Trading Networks when you install Module for EDI.

Step 2: Define Profiles for Trading Partners

You must define profiles for each trading partner with whom you will exchange HIPAA messages. Define profiles for the trading partners that will be identified as senders and receivers on the ISA (envelope) headers.

Use My webMethods to create profiles. For information about creating profiles, see the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.

External ID Types in Profiles and EDI ID Qualifiers

In a Trading Networks profile, specify external IDs to indicate how trading partners are identified within the HIPAA messages that they send. For example, if a trading partner uses a D-U-N-S number in a document, define a **DUNS** external ID type in the Trading Networks profile and specify the trading partner's D-U-N-S number as the corresponding external ID.

For EDI documents, the external IDs correspond to the sender IDs and receiver IDs in the ISA headers of the EDI documents and the external ID types correspond to the EDI ID qualifiers (for example, 01 for a D-U-N-S number).

Module for EDI installs external ID types into Trading Networks the first time you start Integration Server Administrator after installing and enabling Module for EDI. The following table lists these external ID types:

Trading Networks External ID Type	Corresponds to this EDI ID Qualifier
Carrier ID	27
DUNS	01

Trading Networks External ID Type	Corresponds to this EDI ID Qualifier
DUNS+4	14
Federal Tax ID	30
Fiscal Intermediary ID	28
Health Industry Number	20
Medicare ID	29
Mutually Defined	ZZ
NAIC Company Code	33
Note: You must override EDI ID qualifiers to use those IDs in Module for HIPAA (for example, in EDI the ID qualifier code 30 is "ISO 6523" but in HIPAA it is "Federal Tax ID"). For more information about adding and overriding EDI ID qualifiers, see <i>webMethods Module for EDI Installation and User's Guide</i> .	

Step 3: Configure Large Document Handling

To process large HIPAA documents using Module for HIPAA, configure Module for EDI and Trading Networks to handle large documents. This feature temporarily persists documents to local disk for memory and performance optimization.

Configuring Module for EDI

To configure Module for EDI to use large document handling, you must update specific Module for EDI properties. For more information about EDI large document handling and the properties to configure, see *webMethods Module for EDI Installation and User's Guide*.

Configuring Trading Networks

To configure Trading Networks to use large document handling, you must update specific Trading Networks and Integration Server properties.

For more information about Trading Networks large document handling and the properties to configure, see the Trading Networks administration guide for your release. See "About this Guide" for specific document titles.

Step 4: Create EDI Trading Partner Agreements

A trading partner agreement (TPA) is a Trading Networks object that defines how messages are exchanged between two trading partners. An EDI trading partner agreement (EDITPA) is a trading partner agreement that contains Module for EDI-specific settings. The EDITPA contains a set of variables that you provide to tailor how Module for EDI splits HIPAA messages.

Module for EDI supports both partner-specific and default EDITPAs. A partner-specific EDITPA contains variables specific to a pair of trading partners, where one is defined as the sender, and the other the receiver. If a partner-specific EDITPA is not defined, or if a value in a partner-specific EDITPA is not set, Module for EDI uses its default EDITPA. For more information about EDITPAs, see documentation for Module for EDI.

You must define EDITPAs for envelope sender/receiver pairs identified in the ISA headers of the HIPAA messages that you exchange, either by:

- Using the default EDITPA for all envelope-level sender/receiver pairs.
- Creating partner-specific EDITPAs for the envelope-level sender/receiver pairs.

For complete steps to create EDITPAs, see *webMethods Module for EDI Installation and User's Guide*. The following table shows the EDITPA settings for Module for EDI.

Note: Module for EDI only uses the EDITPA variables listed in this table. You can set the values of the other EDITPA variables (not listed below) to any value you choose.

EDITPA variable...	Value to use for the Module for EDI
<i>splitOption</i>	Interchange, Group, or Transaction
<i>GSRouting/routingMode</i>	<p>OFF</p> <p>This variable indicates the value for sender and receiver that Module for EDI should add to the Envelope, Group, and Transaction documents split from the HIPAA message.</p> <p>OFF indicates that Module for EDI uses the sender and receiver from the ISA header for all types of documents.</p>

Step 5: Create HIPAA Trading Partner Agreements

The HIPAA trading partner agreement (HIPAA TPA) defines how messages are validated and the acknowledgments are generated.

A HIPAA TPA contains settings specific to Module for HIPAA, based on the `wm.esd.hipaa.rec:HipaaParameters` IS document types. Modify these variables to tailor how messages are validated between two trading partners and how Module for HIPAA generates acknowledgments.

Module for HIPAA provides a default TPA for an Unknown sender and receiver, which contains the default settings for all senders and receivers. You can edit this TPA or create a trading partner-specific TPA.

To create a partner-specific TPA, in the Agreement Details screen in My webMethods define the following values:

- Specify HIPAA as the value for Agreement ID.
- Identify the sender and receiver.
- For IS document type, specify the value `wm.esd.hipaa.rec:HipaaParameters`.

For more information about creating and editing TPAs, see the chapter on defining and managing TPAs in the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.

HIPAA Parameters

The following sections describe the parameters in the Interchange, Group, and Transaction sections of the TPA. The values for these parameters are specific to ANSI X12 standard document types.

Interchange Parameters

This section describes the parameters in the Interchange section of the TPA.

Parameter	Description
<i>ControlVersion</i>	Specifies the value of ISA segment, element 12 (for example, 00501 for version 5010).
<i>SenderID</i>	Specifies the value of ISA segment, element 06. For an Unknown sender, leave this field blank to use the default HIPAA TPA.

Parameter	Description
<i>SenderQualifier</i>	Specifies the value of ISA segment, element 05. For an Unknown sender, leave this field blank to use the default HIPAA TPA.
<i>ReceiverID</i>	Specifies the value of ISA segment, element 08. For an Unknown receiver, leave this field blank to use the default HIPAA TPA.
<i>ReceiverQualifier</i>	Specifies the value of ISA segment, element 07. For an Unknown receiver, leave this field blank to use the default HIPAA TPA.
<i>TA1</i>	<p>Specifies when to generate a technical acknowledgment:</p> <ul style="list-style-type: none"> ■ always-Default. Always generate the acknowledgment. ■ never-Never generate the acknowledgment. ■ error-Generate the acknowledgment only if there are errors during validation of input data. ■ data-Generate the acknowledgment based on the data item. If there is no such item, this option is the same as always.

CORE Parameters

This section describes the parameters in the CORE section of the TPA.

Parameter	Description
<i>ReceiverUrl</i>	Specifies the URL to which the HIPAA CORE messages are sent.
<i>RealtimeResponseTimeout</i>	<p>Specifies the response timeout for any real time request received by the trading partner. The default value is 20 milliseconds.</p> <p>Note: If this parameter is not set in HIPAA TPA, the default value will be used.</p>

Group Parameters

This section describes the parameters in the Group section of the TPA.

Parameter	Description
997	<p>Specifies when to generate a functional acknowledgment:</p> <ul style="list-style-type: none"> ■ always-Default. Always generate the acknowledgment. ■ never-Never generate the acknowledgment. ■ error-Generate the acknowledgment only if there are errors during validation of input data.
999	<p>Specifies when to generate an implementation acknowledgment:</p> <ul style="list-style-type: none"> ■ always-Default. Always generate the acknowledgment. ■ never-Never generate the acknowledgment. ■ error-Generate the acknowledgment only if there are errors during validation of input data.
<i>VersionNumber</i>	<p>Specifies the value of GS segment, element 08.</p> <p>Note: Type only those version numbers that are related to the control version defined in the Interchange parameter. For example, if the Interchange parameter <i>ControlVersion</i> is set to 00501, select version numbers that start with "00501". Be careful not to have multiple configurations for the same version or validation will fail.</p>

Transaction Parameters

This section describes the parameters in the Transaction section of the TPA. You must create an entry for each type of transaction.

Note: Add only those transactions that are related to the version number specified in the Group parameter *VersionNumber*. For example, if *VersionNumber* is set to 005010X279, you can add transactions 270 and 271.

Parameter	Description
<i>TransactionID</i>	Specifies the transaction number. Valid values are 270, 271, 276, 277, 278-Req, 278-Res, 820, 834, 835, 837, and 999.
<i>AcknowledgmentOption</i>	Specifies the type of acknowledgment to generate and when to generate it:

Parameter	Description
	<ul style="list-style-type: none"> ■ always-Default. Always generate the acknowledgment. ■ never-Never generate the acknowledgment. ■ error-Generate the acknowledgment only if there are errors during validation of input data. <p>Acknowledgment document types are:</p> <ul style="list-style-type: none"> ■ 997 ■ 999 ■ 277A (generated for 837 transactions) <div> <p>Note:The 277A acknowledgement is set to never- by default. Change the value to always- to generate acknowledgements.</p> <p>webMethods Module for HIPAA does not generate 277A acknowledgments if there are errors during validation of input data.</p> </div>
<i>schemaPath</i>	Specifies the directory path to the schema. If this parameter is not specified, the module uses the schema specified in the <i>Integration Server_directory\instances\instance_name\packages\WmHIPAA\resources\schema</i> directory.
<i>ISDocPath</i>	Specifies the fully qualified name of the IS document created from the schema specified in <i>schemaPath</i> . If this parameter is not specified, the module uses the IS documents in the HIPAAschema directory of the module.
<i>SeverityDefinition</i>	Specifies the severity settings to use for validation. For details, see " Severity Definition Parameters " on page 41 , below.

Severity Definition Parameters

The Severity Definition section of the TPA defines the error severity categories and their names and values. The following table describes the parameters for configuring the error severity of WEDI-SNIP certification types. Using My webMethods, you can add a new row for each of the seven HIPAA validation levels, to customize error severity definitions for each validation level.

Parameter	Description
<i>ValidationLevel</i>	Specifies the HIPAA validation level for which to define error severity settings. Possible values are 1 through 7. For detailed explanations of each validation level, see "HIPAA Validation Levels" on page 18 .
<i>Severity</i>	<p>Specifies the severity of the error:</p> <ul style="list-style-type: none">■ Ignore-Accept input data, generate an acknowledgment. Do not log an error in the reports.■ Information-Accept input data and generate an acknowledgment. Log error in reports as severity, "Information."■ Warning-Accept input data with error(s) and generate an acknowledgment. Log error in reports as severity, "Warning."■ Error-Reject input data, generate an acknowledgment. Log the error in the reports as severity, "Normal."
<i>CustomErrorMessage</i>	Specifies custom error message text to display along with the default message in the report.
<i>ErrorIDs</i>	Specifies the error IDs associated with the validation level, severity, and custom error message in the report.

4

Creating Clients that Send HIPAA Messages

■ Overview	44
■ Content Type to Use	45
■ Service the Client Invokes	45
■ How pub.esd.hipaa.receive Handles TA1 Transactions	45
■ How pub.esd.hipaa.receive Handles Other HIPAA Transactions	47

Overview

You create an Integration Server client to send HIPAA messages to Integration Server. Examples of applications that might use clients to send HIPAA messages are:

- An external system that sends a HIPAA message (for example, SAP or Oracle).
- Integration Server sends a HIPAA message to another server.
- A trading partner that is not using webMethods software that sends a HIPAA message.

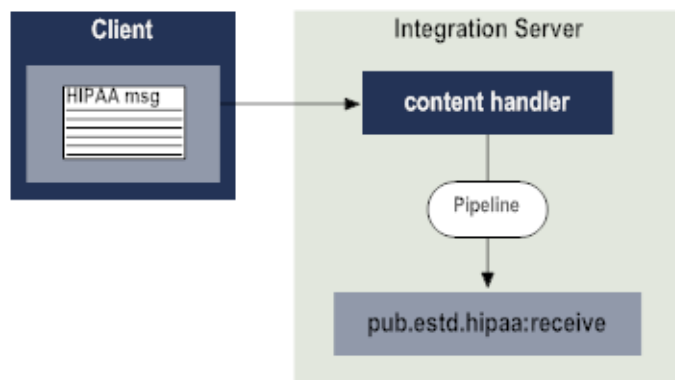
The client can use one of the following transport types to send the HIPAA messages:

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

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

When a client sends a message to Integration Server, the client must specify the content type of the data and identify which service to invoke to start processing the message. When Integration Server receives the message, it passes the message to the appropriate content handler based on the specified content type. The content handler, then, begins processing, which includes creating the pipeline. For more information about creating clients, see the Designer Service Development online help for your release. See “About this Guide” for specific document titles.

Client sends HIPAA messages to Integration Server



Content Type to Use

The content type your client should use to send the HIPAA messages to Integration Server depends on the type of HIPAA message that you send.

When your client sends....	It should use this content type
TA1 Technical Acknowledgment	application/x-wmflatfile
All other types of HIPAA messages	application/EDISStream
Note: For backward compatibility, Module for EDI also has content handlers to accept documents with the content types <code>application/EDI</code> and <code>application/X12</code> . With these content types, Module for EDI content handler must convert the document to a string and place it in the pipeline. This can potentially consume a lot of pipeline space and use a significant amount of memory. As a result, it is recommended that you use the content type <code>application/EDISStream</code> because it conserves system memory.	

Service the Client Invokes

After the content type handler forms the pipeline, it invokes the service that the client specifies. Your client should invoke the [pub.estd.hipaa:receive](#) service. The behavior of this service depends on whether you are sending a TA1 technical acknowledgment transaction or another type of HIPAA transaction.

How pub.estd.hipaa:receive Handles TA1 Transactions

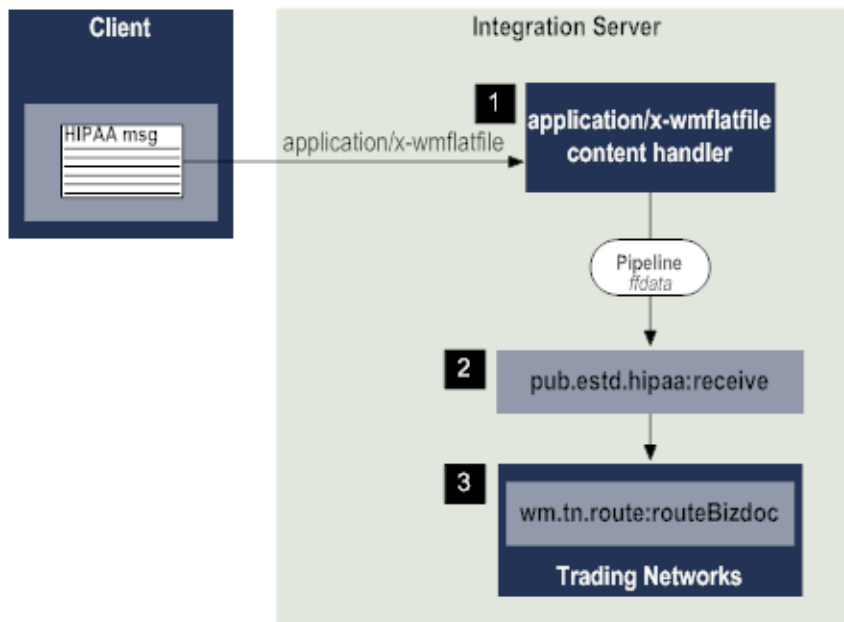
Because Module for EDI does not support TA1 technical acknowledgment transactions, Module for HIPAA adds this support by treating the TA1 technical acknowledgment as a flat file document in Trading Networks rather than as an EDI document.

When the client sends a TA1 technical acknowledgment message to Integration Server, the [pub.estd.hipaa:receive](#) service acts as a Trading Networks document gateway service. The gateway service places additional information about the TA1 technical acknowledgment in the pipeline that Trading Networks uses during its recognition processing. Trading Networks then matches the document to the X12 TA1 ACK TN document type and proceeds with its normal processing. For more information about

Trading Networks flat file processing, see the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.

The following diagram illustrates the processing that occurs when a client sends a TA1 technical acknowledgment to Integration Server.

Client sends a TA1 technical acknowledgment to Integration Server



Step	Description
1	The client sends the HIPAA message with the content type <code>application/ x-wmflatfile</code> to Integration Server, which in turn passes the HIPAA message to the <code>application/x-wmflatfile content handler</code> . The content handler performs initial processing, including forming the pipeline and placing the <code>ffdata</code> variable in the pipeline. The <code>ffdata</code> variable contains the HIPAA message data.
2	The content handler invokes the service specified by the client. For a HIPAA message, the client should specify the <code>pub.estd.hipaa:receive</code> service. This service determines that the HIPAA message is a TA1 technical acknowledgment, and therefore, acts as a gateway service for Trading Networks flat file processing. The <code>pub.estd.hipaa:receive</code> service adds information to the pipeline. It also creates the BizDocEnvelope and sets the TN document type for the HIPAA message to X12 TA1 ACK. The service then invokes the <code>wm.tn.route:routeBizdoc</code> service.

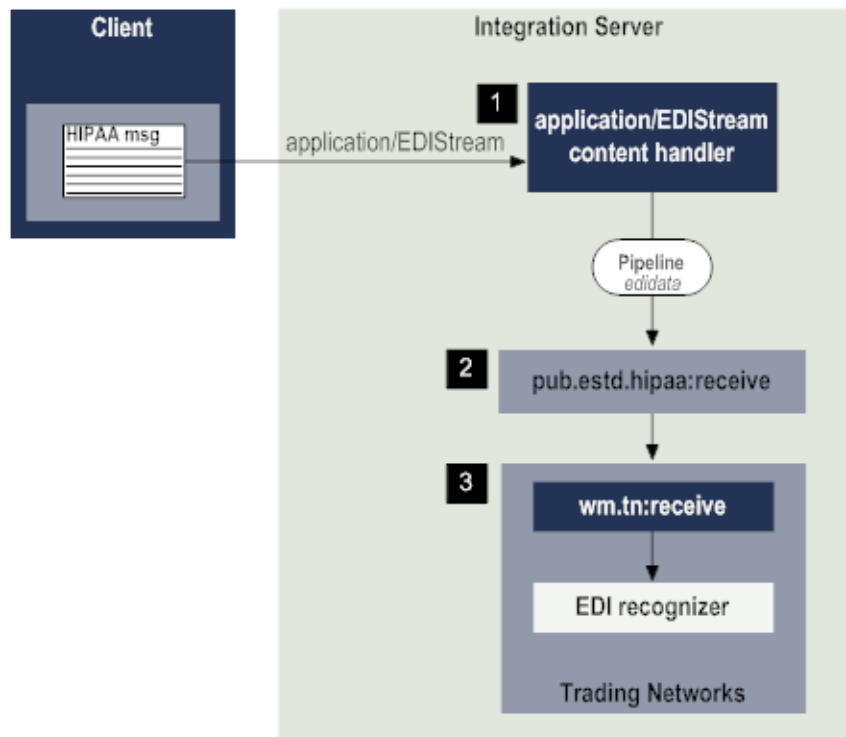
Step	Description
3	The <code>wm.tn.route:routeBizdoc</code> service sends the HIPAA message directly to Trading Networks processing rules, bypassing document recognition. Trading Networks document recognition is bypassed because the <code>pub.estd.hipaa:receive</code> service already performed this function by creating the BizDocEnvelope and determining the TN document type to use for the HIPAA message.

How `pub.estd.hipaa.receive` Handles Other HIPAA Transactions

All HIPAA transactions other than a TA1 technical acknowledgment are treated as regular EDI documents. When the client sends any HIPAA message other than a TA1 technical acknowledgment to Integration Server, the `pub.estd.hipaa:receive` service invokes the `wm.tn:receive` service, starting normal Trading Networks processing. Because the HIPAA message is an EDI document, Trading Networks passes the document to the EDI recognizer for processing. For more information about how EDI documents are processed in Trading Networks, see the documentation for Module for EDI.

The following diagram illustrates the processing when a client sends a HIPAA message that is not a TA1 technical acknowledgment to Integration Server. For more information, see the table below the diagram.

Client sends a HIPAA message other than a TA1 technical acknowledgment to Integration Server



Step	Description
1	The client sends the HIPAA message with the content type <code>application/EDISStream</code> to Integration Server, which in turn passes the HIPAA message to the <code>application/EDISStream content handler</code> . The content handler performs initial processing, including forming the pipeline and placing the <code>edidata</code> variable in the pipeline. The <code>edidata</code> variable contains the HIPAA message data.
2	The content handler invokes the service specified by the client. For a HIPAA message, this should be the <code>pub.estd.hipaa:receive</code> service. The <code>pub.estd.hipaa:receive</code> service determines that the HIPAA message is not a TA1 technical acknowledgment, and therefore, invokes only the <code>wm.tn:receive</code> service.
3	The <code>wm.tn:receive</code> service is the start of normal Trading Networks processing. Because the variable, <code>edidata</code> , is in the pipeline, Trading Networks passes the document to the <code>EDI recognizer</code> for EDI specific handling.

5 Code Source Management

- Overview 50
- Creating Database Tables 50
- Importing Code Sources 51
- Viewing a Codelist 54
- Adding codes 54
- Deleting codes 55
- Dropping Database Tables 55
- List of Code Sources 56

Overview

A *Code Source* contains a Codelist which is a predefined list of codes. Specific fields in a HIPAA message are represented using a code from this list. For example, a field representing the ZIP code in a HIPAA message should have code values from the code source, 51. Module for HIPAA requires code sources to validate the various fields in a HIPAA message.

The different industry standard code sources are bundled in the WmHipaaCodeSource package installed with webMethods Module 9.6 for HIPAA. For the complete list of code sources bundled with the module, see ["List of Code Sources" on page 56](#).

Create database tables that can hold the list of codes from each code source. Then import the code sources from the package into Module for HIPAA so that the module can identify and validate the codes in a HIPAA message.

Creating Database Tables

Create a database table for each code source you want to import into Module for HIPAA in Integration Server Administrator. webMethods provides database scripts to create database tables for storing the imported data.

To create a database table

1. Verify that the database instance to which Trading Networks is configured to connect, is running.
2. Navigate to the directory location, *Integration Server_directory* \instances\default \packages\WmHipaa\config\dbscripts and locate CodeSourceDBCreateTable.sql script file.
3. Edit the CodeSourceDBCreateTable.sql script file to create tables using SQL queries. Create a table for each code source you want to import.

The table should contain the following columns.

- a. "CODE" VARCHAR(*column-length*) NOT NULL PRIMARY KEY
- b. "SHORT_DESCRIPTION" VARCHAR(*column-length*)
- c. "LONG_DESCRIPTION" VARCHAR(*column-length*)
- d. "CUSTOM" CHAR
- e. "DELETED" CHAR

For example, to create a table that holds the list of codes from a code source 51, type the following SQL query.

```
CREATE TABLE "WMHIPAA_CODE_SET_51" ( "CODE" VARCHAR(255) NOT NULL PRIMARY KEY,
"SHORT_DESCRIPTION" VARCHAR(255),
"LONG_DESCRIPTION" VARCHAR(255),
```

```
"CUSTOM" CHAR, "DELETED" CHAR )
```

4. Run the SQL script file using the appropriate database client.
5. When prompted, connect to the same database instance that you configured when installing Trading Networks.

Importing Code Sources

Before you import code sources, you need to add the code sources to the WmHipaaCodeSource package. Use either of the following two methods to make the required code sources available to Module for HIPAA.

Method 1

To make the code sources available to the module using Method 1

1. Identify the code source IDs of the code sources to be imported. For example, the code source ID of a code source that contains a list of Zip codes is 51.
2. Create a directory, `codesource_codesourceid` in the *Integration Server_directory* \instances\instance_name\packages\WmHipaaCodeSource\resources\codesources\versionx directory. For example, `codesource_51`.

The directory `versionx`, represents the version of the imported code source where, *x* is the version number, for example, `version1`.

You must create directories for every imported code source and code source version, if they do not exist.

3. Create a text file, `metadata.txt` in the *Integration Server_directory* \instances\instance_name\packages\WmHipaaCodeSource\resources\codesources\versionx\codesource_codesourceid directory.
4. Define the following properties in the `metadata.txt` file.
 - a. `Description = Insert a description`
The description of the code source you are importing.
 - b. `TableName = Insert the name of the database table`
The name of the database table created during the installation of the module. For more information, see .
 - c. `CodeSetFile = Insert the directory path of the code source`
The directory path of the code source file containing the list of codes.
 - d. `Delimiter = Insert a delimiter`
Define a single character that separates columns in a code source. A delimiter can be alphabets, numerics, or special characters.

This property is mandatory.

Note: If the delimiter contains more than one character, the code source import will fail.

Use “\\” to define a special character as a delimiter. For example, \\\$ for \$, \\: for :, \\ \\u0009 for tab, \\ \\u0020 for space.

- e. `codeColumnNumber` = *Insert the column number that contains the code*

The column number in the code source file that contains the code.

This property is mandatory.

- f. `ShortDescriptionColumnNumber` = *Insert the column number that contains the short description*

The column number in the code source file that contains the short description of the code.

This property is optional. If not specified, the short description is considered as null.

- g. `LongDescriptionColumnNumber` = *Insert the column number that contains the long description*

The column number in the code source file that contains the long description of the code.

This property is optional. If not specified, the long description is considered as null.

- h. `ignoreFirstRow` = *Insert “true” or “false”*

Set this property to *true* if the first row of the code source files should be ignored. The default is *false*.

This property is optional. If not specified, the first row will *not* be ignored.

Method 2

To make the code sources available to the module using Method 2

1. Identify the code source IDs of the code sources to be imported. For example, the code source ID of a code source that contains a list of Zip codes is 51.
2. Create a directory, `codesource_codesourceid` in the *Integration Server_directory* \instances\instance_name\packages\WmHippaCodeSource\resources\codesources\versionx directory. For example, `codesource_51`.

The directory `versionx`, represents the version of the imported code source where, *x* is the version number, for example, `version1`.

You must create directories for every imported code source and code source version, if they do not exist.

3. Use a Java-Editor and write a Java program to import the code sources.

Your Java class must implement the following abstract class.

```
com.softwareag.estd.hipaa.codesource.extractors.ICodeSetHandler
```

4. Implement the following method in your Java program.

```
public abstract ArrayList<CodeSetData> extractCodes(String version)
throws HipaaException;
```

The Array list of CodeSetData objects contains the fields, code, shortDescription, and longDescription. The field, code is mandatory. The shortDescription and longDescription fields are optional.

Note: Ensure that your Java program can return the code values to the code field.

5. Compile your program and copy the generated Java class file in the *Integration Server_directory \instances\instance_name \packages \WmHipaaCodeSource\code\class* directory.
6. Create a text file, metadata.txt and define the following properties in the specified order.
 - a. CodeSetHandler = *Insert the directory path of the ICodeSetHandler abstract class*
The directory path of the implementation of ICodeSetHandler class.
 - b. Description = *Insert a description*
The description of the code source you are importing.
 - c. TableName = *Insert the name of the database table*
The name of the database table created during the installation of the module. For more information, see .
7. Copy the metadata.txt file to *Integration Server_directory \instances \instance_name \packages \WmHipaaCodeSource \resources \codesources \versionx \codesource_codesourceid* directory.

Importing Code Sources into Module for HIPAA

1. In Integration Server Administrator select **Solutions>HIPAA**.
The **Code Sources** page is displayed in a new browser window.
2. On the **Code Sources** home page, click **Import** on the left pane to view the list of code sources that you can import.
3. Select the code sources you want to import, for example **537** and click **Import**.

The status of the imported code sources is displayed on the page. A detailed error message is displayed on the page in case of a failure.

4. Click **View** on the left pane to view the list of code sources imported into the database table.

Viewing a Codelist

1. Click **View** on the left pane of the **Code Sources** page.

The **View** page is displayed in the browser window with the list of imported code sources.

2. Click the icon  under the **Action** column of a code source.

The **Codelist** page is displayed with the list of codes available in the code source and their description.

You can change the number of codes displayed on a page by typing a value in the **Number of entries to display** field.

Searching within a Code Source

Use the **Search** feature on the **Codelist** page to find information about any code that is available in a code source. Type a complete or partial search string in the text box and click **Search**.

To search using a partial search string, use the format `%search string%`.

Adding codes

You can add new codes to an imported code source by creating a text file containing the list of custom codes. Each code must be defined in a new line and use a field separator, ^ in the code definition. The following format must be used to define the code:

```
code1^shortdescription1^longdescription1
```


```
code2^shortdescription2^longdescription2
```

where, *shortdescription* and *longdescription* are optional parameters.

To add custom codes to an imported code source

1. Click **View** on the left pane of the **Code Sources** page.

The **View** page with a list of imported code sources is displayed on the right pane.

2. On the **View** page, click the icon  under the **Action** column of a code source.

The **Custom** page is displayed in the browser window which contains a **Browse** and an **Add** button.

3. Click **Browse** to navigate to the text file that contains the custom codes to be added.
4. Click **Add**.

The new codes are added to the code source and can be viewed on the **Codelist** page.

Note: If the text file contains a code that already exists in the imported code source, a detailed error message is displayed on the **Custom** page and the new codes in the file will not be added to the code source.


Deleting codes

You can delete codes from an imported code source by creating a text file containing the list of codes to be deleted. Each code must be defined in a new line and use the following format:

```
code1
```

```
code2
```

To delete codes from an imported code source

1. Click **View** on the left pane of the **Code Sources** page.
The **View** page with a list of imported code sources is displayed on the right pane.
2. On the **View** page, click the icon  under the **Action** column of a code source.
The **Custom** page is displayed in the browser window which contains a **Browse** and a **Delete** button.
3. Click **Browse** to navigate to the file that contains the codes to be deleted.
4. Click **Delete**.

Note: The module ignores codes that are present in the text file but not in the imported code source.

Dropping Database Tables

You can drop database tables that are created to store imported code sources. webMethods provides database scripts to drop database tables storing the imported data.

To drop a database table

1. Verify that the database instance to which Trading Networks is configured to connect, is running.
2. Navigate to the directory location, *Integration Server_directory* \instances\default \packages\WmHipaa\config\dbscripts and locate CodeSourceDBDropTable.sql script file.
3. Edit the CodeSourceDBDropTable.sql script file to drop tables using SQL queries.
4. Run the SQL script file using the appropriate database client.
5. When prompted, connect to the same database instance that you configured when installing Trading Networks.

List of Code Sources

The WmHipaaCodeSource package installed with webMethods Module 9.6 for HIPAA is bundled with the following code sources.

Code Source	Description
1	Military Rank and Health Care Service Region
102	Languages ISO 639 Language Code
130	Healthcare Common Procedure Coding System
131	International Classification of Diseases 9th Revision Clinical Modification (ICD-9-CM)
139	Claim Adjustment Reason Code
206	Government Bill of Lading Office Code
22	Canadian Provinces and Territories/Mexican States
229	Diagnosis Related Group Number (DRG)
240	National Drug Code by Format
245	National Association of Insurance Commissioners Code (NAIC)
284	Nature of Injury Code

Code Source	Description
359	Treatment Codes
407	Occupational Injury and Illness Classification Manual
411	Remittance advice remark codes
468	Ambulatory Payment Classification (APC)
507	Health care claim status category code
508	Health care claim status code
51	U.S. ZIP Codes
513	Home Infusion EDI Coalition (HIEC) Product/Service Code List
530	National Council for Prescription Drug Programs Reject/Payment codes
537	Centers for Medicare & Medicaid Services National Provider Identifier
576	Workers Compensation Specific Procedure and Supply Codes
663	Logical Observation Identifier Names and Codes (LOINC)
682	Health Care Provider Taxonomy
716	Health Insurance Prospective Payment System(HIPPS) Rate code for skilled Nursing Facilities
844	Eligibility Category
859	Classification of Race or Ethnicity
860	Race or Ethnicity Collection Code
896	International Classification of Diseases 10th Revision Procedure Coding System (ICD-10-PCS)

Code Source	Description
897	International Classification of Diseases 10th Revision Clinical Modification (ICD-10-CM)
94	International Organization for Standardization (Date and Time)
5.1 and 5.2	Names of Countries, Currencies and Funds
237	Place of Service Codes for Professional Claims

6

HIPAA Acknowledgments

■ Overview	60
■ Technical Acknowledgment	60
■ TA1 Code	61
■ Functional Acknowledgment (997)	62
■ Functional Acknowledgment Error Codes	64
■ Implementation Acknowledgment (999)	68

Overview

Module for HIPAA supports technical acknowledgments (TA1s), functional acknowledgments (FAs), and implementation acknowledgments (999).

Technical Acknowledgment

A Technical Acknowledgment (TA1) notifies the sender whether the X12 interchange is successfully received by the receiver. The acknowledgment reports any syntactic errors in the Interchange Control Header and Interchange Control Trailer.

The TA1 segment in the acknowledgment contains the same *interchange control number* as specified in the control header of the X12 interchange for which the acknowledgment is prepared.

The TA1 does not report the status of the functional groups and transaction sets within the interchange envelope.

The following table describes the fields of a TA1 segment.

Fields	Description
<i>Interchange control number</i>	The control number present in the control header envelope that uniquely identifies the X12 interchange.
<i>Interchange date</i>	Indicates the date, in YYMMDD format, when the X12 interchange was prepared. For example, if the interchange was prepared on January 2, 2014, the interchange date would be 140102.
<i>Interchange time</i>	Indicates the time, in 24-hour format (HHMM), when the X12 interchange was prepared. For example, if the interchange was prepared at 5 hours and 15 minutes, the interchange time would be 0515.
<i>Interchange acknowledgement code</i>	Indicates the Interchange Acknowledgment Code. Valid values are: <ul style="list-style-type: none">■ A — Accepted■ E — Errors. The file contains errors.
<i>Interchange error code</i>	A three-digit number that corresponds to a TA1 code. For more information about TA1 codes, see TA1 Code .

TA1 Code

The following table lists the TA1 codes and their description:

TA1 Code	Description
000	The TA1 segment is accepted.
001	The interchange control number in the header and trailer do not match. The acknowledgment uses the value in the header.
005	The interchange ID qualifier for sender is not valid.
006	The interchange ID for sender is not valid.
007	The interchange ID qualifier for recipient is not valid.
008	The interchange ID for recipient is not valid.
009	The interchange receiver ID is unknown.
010	The Authorization Information Qualifier value is not valid.
011	The Authorization Information value is not valid.
012	The Security Information Qualifier value is not valid.
013	The Security Information value is not valid.
014	The Interchange Date value is not valid.
015	The Interchange Time value is not valid.
016	The Interchange Standards Identifier value is not valid.
017	The Interchange Version Identifier value is not valid.
018	The Interchange Control Number value is not valid.
019	The Acknowledgment Requested value is not valid

TA1 Code	Description
020	The Test Indicator value is not valid.
021	The Number of Included Groups, or the Number of Included Errors value is not valid.
024	The Interchange content is not valid.

Functional Acknowledgment (997)

A Functional Acknowledgment (FA) is a transaction set sent by the receiver of a HIPAA transmission to the sender, acknowledging that the message has been received and its syntax is acceptable. Functional acknowledgments do not indicate that the document has been processed by the receiver.

The FA reports the status of the functional groups, transaction sets, and segments within the X12 interchange envelope.

The following table describes the structure of the segments and fields in a functional acknowledgment:

Segment	Field	Description
AK1		Contains information about the group. For example, AK1*HS*123456*005010X279A1~
	AK101	The name of the group in the X12 interchange.
	AK102	The control number of the group.
	AK103	The code indicating the version, release, and industry identifier in the GS segment of the group.
AK2		Contains information about the transaction within the group. For example, AK2*270*1234*005010X279~
	AK201	The transaction type in the X12 interchange.
	AK202	The control number of the transaction.
	AK203	The code indicating the implementation convention reference in the ST segment of the transaction.

Segment	Field	Description
AK3		Contains the error and error details in a data segment. For example, AK3*NM1*4*2100A*8~
	AK301	The name of the data segment in the X12 interchange that contains an error.
	AK302	The position of the data segment from the start of the transaction set.
	AK303	The ID of a bounded loop.
	AK304	A number that represents an error code. For more information about AK304 error codes, see Error codes for AK304 .
AK4		Contains the error and error details of the data element in a data segment. For example, AK4*4*373*8*asdfghjk~
	AK401-1	The position of the data element in the data segment that contains the error.
	AK401-2	The position of the composite data element in the data segment that contains the error.
	AK401-3	The position of the repeating data element.
	AK402	The data element reference number.
	AK403	A number that represents an error code. For more information about AK403 error codes, see Error codes for AK403 .
	AK404	A copy of the data element that contains an error.
AK5		Contains the status information of the transaction set. For example, AK5*R*5~
	AK501	The status of the transaction set. Valid values are: <ul style="list-style-type: none"> ■ A — Accepted ■ R—Rejected. The file contains errors.

Segment	Field	Description
	AK502 through AK506	A number that represents an error code. For more information about error codes for AK502 through AK506, see Error codes for AK502 through AK506 .
AK9		Contains the status information of the group. For example, AK9*R*1*1*0~
	AK901	The status of the group. Valid values are: <ul style="list-style-type: none"> ■ A — Accepted ■ R—Rejected. The file contains errors.
	AK902	The number of transaction sets included in the functional group trailer.
	AK903	The number of transaction sets present within the group.
	AK904	The number of accepted transaction sets within the group.
	AK905 through AK909	A number that represents an error code. For more information about error codes for AK905 through AK909, see Error codes for AK905 through AK909 .

Functional Acknowledgment Error Codes

The following data elements in a functional acknowledgment contain error codes for different error scenarios:

- AK304
- AK403
- AK501
- AK502 through AK506
- AK901
- AK905 through AK909

Error codes for AK304

The following table lists the error codes for AK304 field and the error description:

AK304 Code	Description
1	The Segment ID is not recognized.
2	An unexpected segment is present in the transaction set.
3	A mandatory segment is missing in the transaction set.
4	The number of loops in the transaction set exceeds the maximum number of permitted loops.
5	The number of segments in the transaction set exceeds the maximum number of permitted segments.
6	The segment is not in the defined transaction set.
7	The segment is not in the proper sequence.
8	The segment contains one or more errors in a data element.

Error codes for AK403

The following table lists the error codes for AK403 field and the error description:

AK403 Code	Description
1	A mandatory data element is missing in the segment.
2	The data element required for the conditional validation is missing.
3	The number of data elements exceed the number of permitted data elements.
4	The data element is too short.
5	The data element is too long.

AK403 Code	Description
6	The character in the data element is not valid.
7	The code value is not valid.
8	The date is not valid.
9	The time is not valid.
10	The segment contains data elements that are excluded.

Error codes for AK501

The following table lists the error codes for AK501 field and the error description:

AK501 Code	Description
A	The transaction is accepted.
R	The transaction is rejected because it contains errors.

Error codes for AK502 through AK506

The following table lists the error codes for fields AK502 through AK506 and the error description:

Codes for AK502 through AK506	Description
1	The transaction set is not supported.
2	The trailer is missing in the transaction set.
3	The control number in the header and trailer of the transaction set do not match.
4	The number of segments specified in the trailer of the transaction set does not match the actual count.
5	One or more segments in the transaction set contains an error.

Codes for AK502 through AK506	Description
6	The transaction set identifier is either missing or not valid.
7	The transaction set control number is either missing or not valid.

Error codes for AK901

The following table lists the error codes for AK901 field and the error description:

AK901 Code	Description
A	The transaction is accepted.
R	The transaction is rejected because it contains errors.

Error codes for AK905 through AK909

The following table lists the error codes for fields AK905 through AK909 and the error description:

Codes for AK905 through AK909	Description
1	The functional group is not supported.
2	The version of the functional group is not supported.
3	The trailer is missing in the functional group.
4	The control number in the header and trailer of the functional group do not match.
5	The number of transaction set specified in the header of the functional group does not match the actual count.
6	The syntax of the control number for the functional group is incorrect.

Implementation Acknowledgment (999)

An Implementation Acknowledgment (999) notifies the sender whether the X12 interchange is successfully received by the receiver. The acknowledgment reports any syntactic errors in the Functional Group Response Header and Functional Group Response Trailer.

The 999 segment in the acknowledgment contains the same *functional group response number* as specified in the control header of the X12 interchange for which the acknowledgment is prepared.

The 999 does not report the status of the functional groups and transaction sets within the interchange envelope.

7

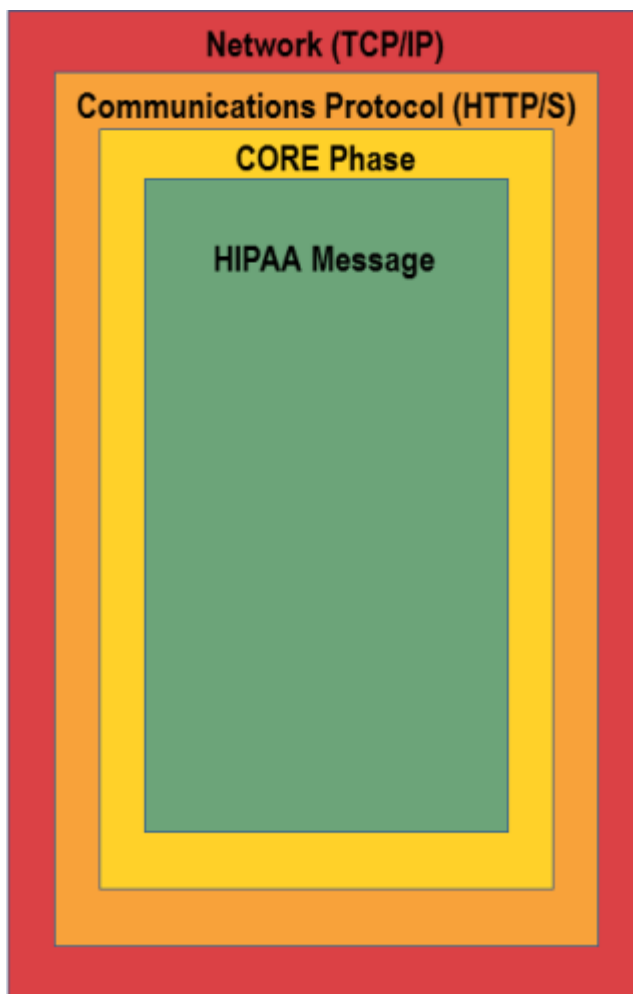
HIPAA CORE Compliance

■ CORE Phase Connectivity	70
■ Support for CORE Phase I and II	70
■ Processing HIPAA Transactions	71

CORE Phase Connectivity

The CORE Phase Connectivity rules define a process which supports two message envelope standards, HTTP MIME Multipart and SOAP for interchange of HIPAA messages between trading partners. CORE Phase reduces the number of envelope methods and facilitates interoperability between Module for HIPAA and trading partners.

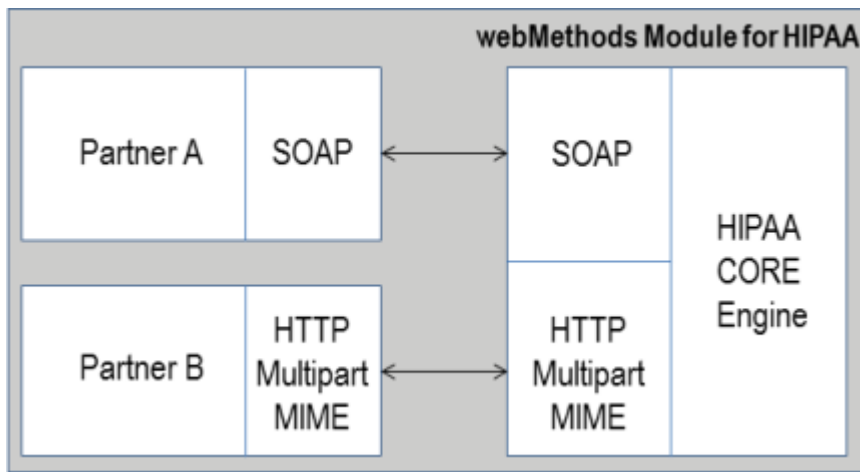
The following diagram illustrates the communication layers involved in sending and receiving a HIPAA CORE message.



Support for CORE Phase I and II

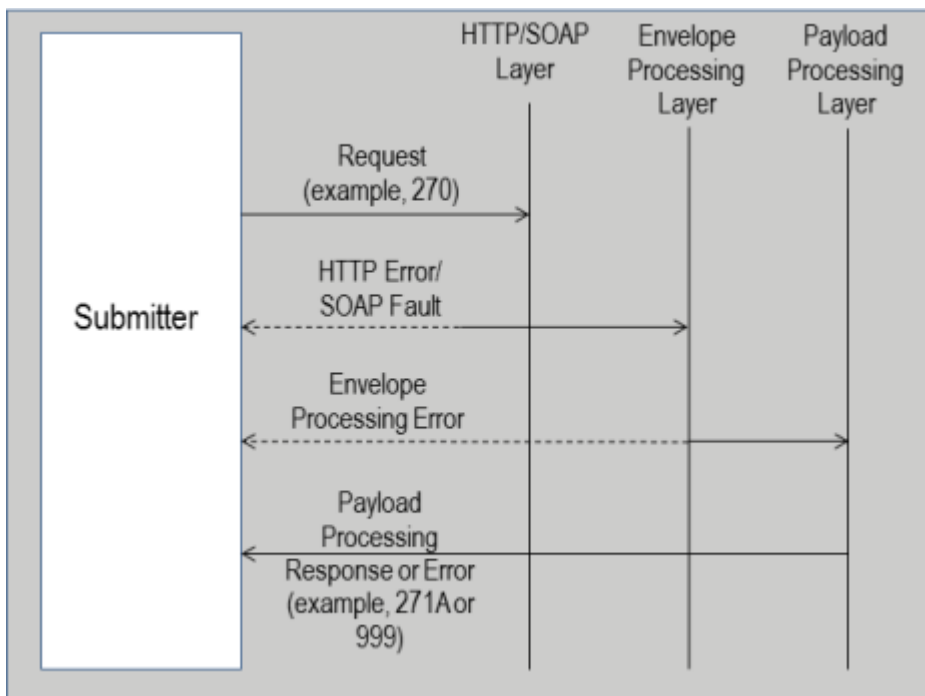
webMethods Module for HIPAA supports CORE Phase I and Phase II Connectivity with Eligibility Benefit Inquiry and Response (270/271) messages in Real time mode.

The following diagram illustrates how Module for HIPAA implements CORE Phase to process HTTP MIME Multipart and SOAP transactions real time.



Processing HIPAA Transactions

webMethods Module for HIPAA processes real time request messages that use HTTP or SOAP envelopes and sends appropriate response messages to the submitter of the request. The module enables secure exchange of messages between trading partners by providing submitter authentication using username and password. Authentication can be enabled using the "[pub.estd.hipaa.core:send](#)" on [page 88](#) service. The following diagram illustrates the processing of a HIPAA transaction.





8 Processing HIPAA Messages Sent to Integration Server

- Overview 74
- Before You Can Process Inbound HIPAA Messages 74
- Using Services to Process Inbound HIPAA Messages 76

Overview

This chapter describes how to configure Integration Server to process inbound HIPAA messages according to the HIPAA standard, including:

- Setting up Integration Server so that your services receive an inbound HIPAA message.
- Validating the HIPAA message.
- Responding with the appropriate acknowledgments (for example, TA1, 997, 999, and 277A).

Important: This chapter describes processing to comply with HIPAA standards for validating and sending appropriate acknowledgments. It does not describe how to process the actual transactions. Transaction processing is the same as for any other EDI document. For information about processing inbound EDI documents, including how to map data from an EDI document to an external system document, see *webMethods Module for EDI Installation and User's Guide*.

Before You Can Process Inbound HIPAA Messages

Before setting up processing for inbound HIPAA message, do the following:

- Install the TN document types and flat file schemas for the HIPAA transactions that you want to process. For instructions, see ["Step 1: Install TN Document Types for HIPAA Transactions" on page 34](#).
- Define profiles for the senders and receivers identified in the ISA headers of the HIPAA messages. For instructions, see ["Step 2: Define Profiles for Trading Partners" on page 35](#).
- Define EDITPA settings for the sender/receiver pairs identified in the ISA headers of the HIPAA messages. For instructions, see ["Step 4: Create EDI Trading Partner Agreements" on page 37](#).
- Define a HIPAA-specific TPA if required for the sender/receiver pairs identified in the ISA headers of the HIPAA message. For instructions, see ["Step 5: Create HIPAA Trading Partner Agreements" on page 38](#).

Using Processing Rules to Process Inbound HIPAA Messages

As described in ["Process Overview" on page 23](#), when Integration Server receives a HIPAA message, it passes the HIPAA message to Trading Networks. Because the HIPAA message is an EDI document, Trading Networks passes the document to the EDI recognizer for Module for EDI-specific recognition processing.

The EDI recognizer splits the document based on the EDITPA *splitOption* variable. To comply with HIPAA standards, you must set the *splitOption* variable to Group or Transaction, so that the EDI recognizer forms at least the envelope and group documents from the HIPAA message.

This section describes how to configure processing rules for the envelope and group documents. You should set up one processing rule for an envelope document and another for a group document.

Note: If you set the *splitOption* variable to *transaction*, the EDI recognizer also creates transaction documents that each contain a single transaction set from the HIPAA message. Define how to process the transaction (for example, map the data to another document to send to your external system). This chapter does not describe how to do this processing.

Defining a Processing Rule for an Envelope Document

To specify how to process the envelope document, you must define a processing rule validates the ISA, GS, and ST segments, in order to comply with the HIPAA standard.

To create a processing rule for the envelope

Create a processing rule in My webMethods. For information about how to create a processing rule, see the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.

1. Set processing rule criteria. Set the following criteria on the **Criteria** tab of the processing rule:

Criteria tab field	Set to...
Sender	Any Senders . You can also select Selected Senders , if desired.
Receiver	Enterprise . Important: You must select Enterprise for this parameter so that the processing rule is only invoked when you are receiving an envelope document (and not when sending one).
Document Type	Selected Document Types. Select document type X12 Envelope Specify X12 Envelope as the TN document type for the envelope document to ensure the processing rule is only invoked when processing an envelope document.

Important: If the envelope validation fails, the EDI recognizer does not split the group and transaction documents from the HIPAA message. As a result, group and transaction documents are not processed. Only the envelope document is passed to Trading Networks for processing, so your logic can handle the error and send a TA1 technical acknowledgment, if appropriate. For more information about how the HIPAA message is split into envelope, group, and/or transaction documents, see ["Process Overview" on page 23](#).

2. Set processing actions. On the **Action** tab of the processing rule do the following:
 - a. Select **Perform the following actions**.
 - b. Select **Execute a service** and specify a service that you created to process the envelope document.

Use the Module for HIPAA `awm.estd.hipaa.sample:processHipaaMessage` sample service as a guideline for creating your own service.

3. For more information about the logic your service must use to meet HIPAA standards and the built-in services provided by Module for HIPAA that you can use as guidelines, see ["Using Services to Process Inbound HIPAA Messages" on page 76](#).

Using Services to Process Inbound HIPAA Messages

This section describes the logic and processing actions that are required for services that process envelope documents, in order to comply with the HIPAA standard.

1. Validate the envelope and perform HIPAA validation levels 1-7 according to the configuration options you selected in ["Step 5: Create HIPAA Trading Partner Agreements" on page 38](#).
2. Process the message based on whether envelope validation errors occur:
 - If envelope validation errors occur, generate a negative TA1 technical acknowledgment and save it to the Trading Networks database. To return the acknowledgment to the trading partner, use Trading Networks delivery features. For more information, see the Trading Networks administration guide for your release. See "About this Guide" for specific document titles..

Important: If the validate pre-processing action determines that the envelope is not valid, the EDI recognizer does not split group and transaction documents from the HIPAA message. Only the envelope document is passed to Trading Networks for further processing.

- If envelope validation errors do not occur, determine whether the sender requested a TA1 technical acknowledgment. If so, generate a TA1 technical acknowledgment and save it to the Trading Networks database. To return the acknowledgment to the trading partner, use Trading Networks delivery features.

For more information, see the Trading Networks administration guide for your release. See “About this Guide” for specific document titles..

3. Generate the configured acknowledgments, such as a 997 functional acknowledgment, to report the validation result and save it to Trading Networks.
4. Using Trading Networks, return an acknowledgment to the trading partner who sent the HIPAA message.
5. Optionally, you may want to update your external system based on information in the HIPAA message. To do so, map data from the HIPAA message to the data format required by your external system and then send the document to that system. For more information about mapping data from HIPAA messages (EDI documents) to another format, see *webMethods Module for EDI Installation and User's Guide*.

The following table lists the built-in services that Module for HIPAA includes to help you perform the above actions. For more information about these services, see ["WmHIPAA Services"](#) on page 83.

Action	Built-in Service to Use
Validate the HIPAA message.	pub.estd.hipaa:validate



9 Processing HIPAA Acknowledgments

- Overview 80
- Before You Can Process Inbound HIPAA Acknowledgments 80
- Defining Processing Rules for Inbound HIPAA Acknowledgments 80



Overview

This chapter describes how to configure Integration Server to process inbound HIPAA acknowledgments such as TA1 technical acknowledgments, 997 functional acknowledgments, 999 implementation acknowledgments, and 277A acknowledgments.

The HIPAA standard does not mandate how you process acknowledgments. Typically, you would map data from the acknowledgment to another document format, which you could then return to an external system.

Before You Can Process Inbound HIPAA Acknowledgments

Before you configure processing for inbound HIPAA acknowledgments, do the following:

- Install the TN document types and flat file schemas for the X12 997 HIPAA transaction. For instructions, see ["Step 1: Install TN Document Types for HIPAA Transactions" on page 34](#). The X12 TA1 ACK TN document type is automatically installed with Module for HIPAA.
- Define profiles for the senders and receivers identified in the ISA headers of the HIPAA acknowledgments. For instructions, see ["Step 2: Define Profiles for Trading Partners" on page 35](#).
- Define EDITPA settings for the sender/receiver pairs identified in the ISA headers of the HIPAA acknowledgments. For instructions, see ["Step 4: Create EDI Trading Partner Agreements" on page 37](#).
- Define a HIPAA-specific TPA if required for the sender/receiver pairs identified in the ISA headers of the HIPAA message. For instructions, see ["Step 5: Create HIPAA Trading Partner Agreements" on page 38](#).

Defining Processing Rules for Inbound HIPAA Acknowledgments

This section describes how to configure processing rules for TA1 and 997 HIPAA acknowledgments. You should define one processing rule for a TA1 technical acknowledgment and another one for a 997 functional acknowledgment.

Defining a Processing Rule for a TA1 Technical Acknowledgment

To specify how to process a TA1 technical acknowledgment, you must define a processing rule and specify this rule in the **Execute a service** processing action for the TA1 technical acknowledgment.

To create a processing rule for a TA1 technical acknowledgment

Create a processing rule in My webMethods. For information about how to create a processing rule, see the chapter about defining processing rules in the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.

Note: No specific settings are required on the **Pre-Processing** tab of the processing rule for a Group document.

1. Set processing rule criteria. Set the following criteria on the **Criteria** tab of the processing rule:

Criteria tab field	Set to...
Sender	Any Senders You can change this to selected senders, if desired.
Receiver	Enterprise It is important to select Enterprise so that the processing rule is only invoked when you are receiving a TA1 technical acknowledgment (and not when sending one).
Document Type	Selected Document Type. Select the document type X12 TA1 ACK Specify X12 TA1 ACK as the TN document type for the TA1 technical acknowledgment to ensure the processing rule is only invoked for the TA1 technical acknowledgment.

2. Set processing actions. On the **Action** tab of the processing rule, do the following:
 - a. Select **Perform the following actions**.
 - b. Select **Execute a service** and specify a service that you created to process the TA1 technical acknowledgment.

Defining a Processing Rule for a 997 Functional Acknowledgment

To define how to process a 997 functional acknowledgment, you must define a processing rule and specify this rule in the **Execute a service** processing action for the 997 functional acknowledgment.

To create a processing rule for a 997 functional acknowledgment

You create processing rules using My webMethods. For information about how to create processing rules, see the chapter about defining processing rules in the Trading

Networks administration guide for your release. See “About this Guide” for specific document titles..

Note: No specific settings are required on the **Pre-Processing** tab of the processing rule for a Group document.

1. Set processing rule criteria. Set the following criteria on the **Criteria** tab of the processing rule.

Criteria tab field	Set to...
Sender	Any Senders You can change this to selected senders, if desired.
Receiver	Enterprise It is important to select Enterprise so that the processing rule is invoked only when you are receiving a 997 functional acknowledgment (and not when sending one).
Document Type	Selected Document Types. Select the document type X12 5010 997 Specify X12 5010 997 as the TN document type for the 997 functional acknowledgment to ensure that the processing rule is only invoked for this acknowledgment type.

2. Set processing actions. On the **Action** tab of the processing rule, do the following:
 - a. Select **Perform the following actions**.
 - b. Select **Execute a service** and specify a service that you created to process the X12 5010 997 document.

10

WmHIPAA Services

■ pub.estd.hipaa:validate	84
■ pub.estd.hipaa:convertHipaaToIdData	86
■ pub.estd.hipaa:convertIdDataToHipaa	86
■ pub.estd.hipaa:normalizeName	87
■ pub.estd.hipaa.core:send	88
■ pub.estd.hipaa:receive	90
■ pub.estd.hipaa:recognizeAcknowledgements	91
■ pub.estd.hipaa:recognizeTA1	91
■ wm.estd.hipaa.core:receive	92
■ wm.estd.hipaa.core:receive_security_auth	92
■ wm.estd.hipaa.core.services:realtimeHttpReceive	93
■ pub.estd.hipaa:attachResponsePayload	93

This section describes the services available in the `pub.estd.hipaa` folder.

pub.estd.hipaa:validate

Validates HIPAA 5010 messages, returns the required acknowledgments, and formats acknowledgments and validation results in XML- and HTML-formatted reports.

Input Variables

<i>hipaaDataFile</i>	String Optional. The file absolute path of the HIPAA message to be validated.
<i>ediDataContentPart</i>	Document Optional. The BizDocContentPart document to be validated. For more information about the structure of this document, see the description for <code>wm.tn.rec:BizDocContentPart</code> in the Trading Networks administration guide for your release. See “About this Guide” for specific document titles.
<i>split</i>	String Optional. Whether to split valid HIPAA data and invalid HIPAA data into separate files after validation so that the valid data can be reused. Specify one of the following: <ul style="list-style-type: none">■ <code>true</code> - Split valid and invalid HIPAA data into separate files after validation.■ <code>false</code> - Default. Do not split the data into separate files.
<i>hipaaMessage</i>	String Optional. The HIPAA message to be validated.

Output Variables

<i>XML_Report</i>	String Conditional. Detailed report of the validation results in XML format.
<i>HTML_Report</i>	String Conditional. Detailed report of the validation results in HTML format.
<i>TA1</i>	String List Conditional. List of the technical acknowledgments corresponding to each interchange validated.
<i>997</i>	String List Conditional. List of the functional acknowledgments corresponding to each functional group validated.

999	String List Conditional. List of the implementation acknowledgments corresponding to each functional group validated.
277A	String List Conditional. List of the acknowledgments generated for 837 transactions validated.
<i>validData</i>	String Conditional. Generated when <i>split</i> is set to <code>true</code> . Contains the data that the validation process determines as valid.
<i>invalidData</i>	String Conditional. Generated when <i>split</i> is set to <code>true</code> . Contains the data that the validation process determines as invalid.
<i>hipaaIData</i>	Document Conditional. The IS document representation of the HIPAA message.
<i>errorMessage</i>	String Conditional. The error message corresponding to the error code generated while validating the message.

Usage Notes

Either *hipaaDataFile* or *ediDataContentPart* must be specified as input. This service formats acknowledgments and validation results in HTML- and XML-formatted reports that contain the following information:

- Indication of whether the data file passed or failed validation.
- Identifying information about the interchange control number and version, the error type and number of errors, and the type of transaction the interchange contained.
- If errors occurred, details about the rejected transaction, including:
 - Error ID
 - Detailed description of the error
 - Error data - the part of the data that caused the error during validation
 - WEDI-SNIP certification type: the seven levels of validation as described in ["Process Overview" on page 23](#)
 - Error severity

The service puts these reports in the pipeline. You can send these reports in an email message to the appropriate person (for example, to correct errors). You can also customize these reports to ignore certain validation level messages, display a custom user message, and prevent the generation of certain acknowledgments, by configuring

HIPAA TPA parameters. For details, see ["Step 5: Create HIPAA Trading Partner Agreements" on page 38.](#)

pub.estd.hipaa:convertHipaaToIData

Converts a HIPAA message into an IS document (IData object) that contains the internal document format.

Input Variables

<i>hipaaMessage</i>	Object Mandatory. Contains the HIPAA message to be converted into an IS document.
<i>documentType</i>	String Optional. Contains the namespace to the IS document type of the HIPAA message. The service displays the document type in the format <i>namespace:IS document type</i> (for example, HIPAASchema.X12.V5010.HR.X212:T276DT).
<i>charsetEncoding</i>	String Optional. Used to determine the encoding when reading the HIPAA message. By default the characters are encoded using UTF-8.

Output Variables

<i>hipaaIData</i>	Document Mandatory. The IS document representation of the HIPAA message.
<i>errorMessage</i>	String Conditional. The error message corresponding to the error code generated while validating the message.

pub.estd.hipaa:convertIDataToHipaa

Converts an IS document (IData object) that contains the internal document format into a HIPAA message.

Input Variables

<i>hipaaIData</i>	Document Mandatory. The IS document representation of the HIPAA message.
<i>delimiters</i>	Document Optional. Delimiters used to create the output HIPAA message. If no delimiters are specified, the convertIDataToHipaa

service uses the default delimiters. Delimiters must always be a single character.

Value	Description
<i>segment</i>	String Optional. The segment terminator character that you want the service to append to the end of each record in the output String.
<i>dataElement</i>	String Optional. The field separator that you want the service to insert between each field for each segment in the output String.
<i>componentElement</i>	String Optional. The subfield separator that you want the service to use for composite elements.
<i>repetition</i>	String Optional. The field separator that you want the service to insert between repeating fields of an IS document.

Output Variables

<i>hipaaMessage</i>	String Mandatory. Contains the HIPAA message converted from the IS document.
<i>errorMessage</i>	String Conditional. The error message corresponding to the error code generated while validating the message.

pub.estd.hipaa:normalizeName

Creates a normalized name of an individual based on the standards specified in the Phase II CORE 258 document. This service ensures that the module maintains a unique identity of the individual.

Input Variables

<i>name</i>	String Mandatory. Name of the individual to be normalized.
-------------	---

Output Variables

<i>normalizedName</i>	String Mandatory. The normalized name created by the service.
<i>errorCode</i>	String Conditional. The service throws an error code 73 if the input variable is empty or the name is invalid.
<i>errorMessage</i>	String Conditional. The error message corresponding to the error code.

pub.estd.hipaa.core:send

Sends HIPAA CORE messages to the trading partner.

Input Variables

<i>EnvelopeStandard</i>	String List Mandatory. The envelope standard of the CORE message. The available options are: <ul style="list-style-type: none">■ SOAP■ HTTP
<i>InputMessageType</i>	String List Mandatory. The CORE message type to be sent to the trading partner. The available options are: <ul style="list-style-type: none">■ X12_270_Request_005010X279A1
<i>InputMessage</i>	String List Mandatory. The CORE message to be sent to the trading partner.
<i>SenderID</i>	String Mandatory. The unique ID of the trading partner sending the CORE message.
<i>ReceiverID</i>	String Mandatory. The unique ID of the trading partner receiving the CORE message.
<i>ProcessingMode</i>	String List Mandatory. The method used to process requests between the sender and receiver. Module for HIPAA uses Real time method to process requests.
<i>URL</i>	String List Optional. The URL of the trading partner to which you want to send the message. If a URL is not specified, the service uses the URL provided in the ReceiverUrl field of the TPA.

Note: The receiver URL must be provided in the following formats:

- `http://<hostname:port>/ws/wm.estd.hipaa.core:receive`
- `http://<hostname:port>/ws/wm.estd.hipaa.core:receive_security_auth`
- `http://<hostname:port>/invoke/wm.estd.hipaa.core.services/realtimeHttpReceive`

For more information, see

["wm.estd.hipaa.core:receive" on page 92](#),
["wm.estd.hipaa.core:receive_security_auth" on page 92](#),
 and ["wm.estd.hipaa.core.services:realtimeHttpReceive" on page 93](#).

Authentication

String Optional. Authentication required to access the trading partner's SOAP or HTTP web service. Configure the following parameters.

Message Authentication required to process the CORE compliant message. For a SOAP message, the service adds WS-Security Username Token to the SOAP message body. For an HTTP message, the service adds Username and Password to the MIME body.

UserName **String** Mandatory. The authentication user name to process the CORE compliant message.

Password **String** Mandatory. The authentication password to process the CORE compliant message.

Transport Authentication required to access the trading partner's web services.

Note: Integration Server provides transport level security through Secure Sockets Layer (SSL) for the module's message transactions. You must configure SSL on both, enterprise and trading partner's Integration Server. For more information, see *webMethods Integration Server Administrator's Guide*.

UserName **String** Mandatory. The authentication user name to access the SOAP or HTTP services.

*Password***String** Mandatory. The authentication password to access the SOAP or HTTP services.

Output Variables

<i>responseMessageType</i>	String Mandatory. The response message type corresponding to the CORE message.
<i>responseMessage</i>	String Mandatory. The response message to confirm that the CORE message was received.
<i>errorCode</i>	String Conditional. The error code indicating whether or not the exchange of message between the sender and receiver was a success.
<i>errorMessage</i>	String Conditional. The error message corresponding to the error code.
<i>fault</i>	Document Conditional. Contains information about the faults that occurred while processing the request.
<i>isEnvelopeError</i>	String Conditional. Indicates whether or not there is an error in the message envelope. The value is <code>true</code> if there is an error. Else, the value is <code>false</code> .
<i>isFault</i>	String Conditional. Indicates whether or not there is a fault in the CORE message. The value is <code>true</code> if there is a fault. Else, the value is <code>false</code> .
<i>statusCode</i>	String Conditional. The status code indicating the status of an HTTP response.
<i>statusMessage</i>	String Conditional. The status message corresponding to the status code.

pub.estd.hipaa:receive

This service receives, recognizes, and saves a HIPAA transaction or acknowledgment to the Trading Networks database.

Input Variables

ffdata **Object** (optional) The HIPAA transaction, 997 functional acknowledgment, or TA1 technical acknowledgment.

Output Variables

None.

Usage Notes

Use this service to receive a HIPAA Real time transaction, a TA1 technical acknowledgment from a trading partner, or acknowledgments such as 997 and 999. When sending the HIPAA message, the trading partner must set the content-type for the post to:

- application/x-wmflatfile when sending a TA1 technical acknowledgment.
- application/EDISTream, application/EDI, or application/X12 when sending a HIPAA transaction, 997 functional acknowledgment, or 999 implementation acknowledgment.

pub.estd.hipaa:recognizeAcknowledgements

This service recognizes and persists an acknowledgment in the Trading Networks database.

Input Variables

edidata **Object** Acknowledgment data.

Output Variables

None.

pub.estd.hipaa:recognizeTA1

This service recognizes and persists a technical acknowledgment (TA1) to the Trading Networks database.

Input Variables

ffdata **String** Technical acknowledgment data.

- prtIgnoreDocument* **String** Indicates whether to create a new process model instance for the TA1 document with the generated conversation ID. Specify one of the following:
- `true`-Do not create a process model instance.
 - `false`-Create a new process model instance to listen for the TA1 document using the generated conversation ID. Use this setting only if a process model exists to listen for the TA1 document.

Output Variables

None.

wm.estd.hipaa.core:receive

This web service receives, recognizes, validates, and saves a CORE SOAP HIPAA transaction to the Trading Networks database, and sends an acknowledgment such as TA1, 999, 997, or CORE envelope error.

Input Variables

None.

Output Variables

None.

Usage Notes

Use the following URL format in the URL input of the send service ["pub.estd.hipaa.core:send" on page 88](#) when sending messages to this receive service.

`http://<hostname:port>/ws/wm.estd.hipaa.core:receive`

wm.estd.hipaa.core:receive_security_auth

This web service receives, recognizes, validates, and saves a CORE SOAP HIPAA transaction that contains WS-Security Username and Password Token added in the message header to the Trading Networks database, and sends a response message or acknowledgment such as TA1, 999, 997, or CORE envelope error.

Input Variables

None.

Output Variables

None.

Usage Notes

Use the following URL format in the URL input of the send service
pub.estd.hipaa.transport:send when sending messages to this receive service.

http://<hostname:port>/ws/wm.estd.hipaa.core:receive_security_auth

wm.estd.hipaa.core.services:realtimeHttpReceive

This service receives, recognizes, validates, and saves a CORE HTTP HIPAA transaction to the Trading Networks database, and sends a response message or acknowledgment such as TA1, 999, 997, or CORE envelope error.

Input Variables

None.

Output Variables

None.

Usage Notes

Use the following URL format in the URL input of the send service
pub.estd.hipaa.transport:send when sending messages to this receive service.

http://<hostname:port>/invoke/wm.estd.hipaa.core.services/realtimeHttpReceive

pub.estd.hipaa:attachResponsePayload

This is a custom service which runs a response service to generate a real time response. The module invokes this service internally when a real time request is received by the trading partner.

You must create your own response service to generate a real time response. This response service must be placed directly below **wm.estd.hipaa.util:extractContentFromBizdoc** within the service. This service takes BizdocEnvelope as input from the pipeline.

A sample service **wm.estd.hipaa.sample.util:getDefaultPayload** is provided in the Module for HIPAA Sample Package. For information about the default payload service, see *webMethods Module for HIPAA Sample Package User's Guide Version 9.6*.

Input Variables

None.

Output Variables

ResponsePayload **String** Conditional. The real time response generated by the custom response service.