

iWay

iWay Integration Solution for HIPAA User's Guide Version 7.0.x and Higher

Active Technologies, EDA, EDA/SQL, FIDEL, FOCUS, Information Builders, the Information Builders logo, iWay, iWay Software, Parlay, PC/FOCUS, RStat, Table Talk, Web390, WebFOCUS, WebFOCUS Active Technologies, and WebFOCUS Magnify are registered trademarks, and DataMigrator and Hyperstage are trademarks of Information Builders, Inc.

Adobe, the Adobe logo, Acrobat, Adobe Reader, Flash, Adobe Flash Builder, Flex, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2018, by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Contents

Preface	7
Documentation Conventions	
Related Publications	9
Customer Support	9
Help Us to Serve You Better	
User Feedback	
Information Builders Consulting and Training	
1. Introducing the iWay Integration Solution for HIPAA	
The HIPAA Mandate	
Achieving Administrative Simplification	
Promoting HIPAA Compliance and Integration	
Legacy Integration	
Features of the iWay Integration Solution for HIPAA	
HIPAA Information Roadmap	
2. Deployment Information for Your iWay Integration Solution	21
iWay Products	
iWay Service Manager.	
iWay Transformer	
Understanding Pipelined Transformations.	
Output Node Name	
iIT Designer	
Using a Channel to Construct a HIPAA Message Flow	23
Components of a Channel	
Components of the iWay Integration Solution for HIPAA	
Ebix	27
Preparsers	
HipaaSplitterPreParser	
HIPAABatchSplitter	
Acknowledgement Service	
Deidentification Service	
Preemitter	

\sim	nto	nto
1.0	rne	ents
~~		

	Data Segments and Data Elements	29
3. 0	Configuring the EDI Activity Driver	31
	HIPAA EDI Activity Driver Overview	
	Configuring the EDI Data Provider Using iWay Service Manager	
	Configuring the EDI Activity Driver Using iWay Service Manager	
4. V	Norking With HIPAA Inbound and Outbound Applications Using iWay	
I	ntegration Tools (iIT)	
	HIPAA Inbound and Outbound Applications Overview	41
	HIPAA Inbound and Outbound Applications Prerequisites	
	Extracting HIPAA User Samples	42
	Importing HIPAA User Samples to iWay Integration Tools as a Workspace	45
	Publishing iWay Integration Applications to the iWay Service Manager Registry	
	Deploying iWay Integration Applications to iWay Service Manager	
	Setting Registers in the iWay Service Manager Administration Console	
	Stopping Inbound (HIPAA to XML) and Outbound (XML to HIPAA) Processing	60
	Testing the Sample HIPAA Applications	62
5. I	nbound Processing: HIPAA to XML	67
5. I	nbound Processing: HIPAA to XML	67
5. I	nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML	67 67
5. I	nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console.	67 67 69 69
5. I	nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry.	67 67 69 69 70
5. I	nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets.	67 67 69 70 73
5. I	nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets. Defining an Inlet.	67 67 69 70 73 74
5. I	nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets. Defining an Inlet. Defining a Route.	67 67 69 70 73 74 74
5. I	nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets. Defining an Inlet. Defining the Outlets.	67 67 69 70 73 74 74 83 102
5. I	nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets. Defining an Inlet. Defining a Route. Defining the Outlets. Defining a Channel.	67 67 69 70 70 73 74 83 102 110
5. I	nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets. Defining an Inlet. Defining a Route. Defining the Outlets. Defining a Channel. Reusing Your Channel Configuration.	67 67 69 70 73 74 74 83 102 110 114
5. I 6. (nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets. Defining an Inlet. Defining a Route. Defining the Outlets. Defining a Channel. Reusing Your Channel Configuration.	
5. I 6. (nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets. Defining an Inlet. Defining a Route. Defining the Outlets. Defining a Channel. Reusing Your Channel Configuration.	
5. I 6. (nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets. Defining an Inlet. Defining a Route. Defining the Outlets. Defining a Channel. Reusing Your Channel Configuration. Dutbound Processing: XML to HIPAA HIPAA Outbound Processing Overview Sample Configuration for Outbound Processing: XML to HIPAA	
5. I 6. (nbound Processing: HIPAA to XML HIPAA Inbound Processing Overview Sample Configuration for Inbound Processing: HIPAA to XML Accessing the iWay Service Manager Administration Console. Adding an Ebix to the Registry. Adding Special Register Sets. Defining an Inlet. Defining a Route. Defining the Outlets. Defining a Channel. Reusing Your Channel Configuration. Dutbound Processing: XML to HIPAA HIPAA Outbound Processing Overview Sample Configuration for Outbound Processing: XML to HIPAA Accessing the iWay Service Manager Administration Console.	

	Adding Special Register Sets	
	Defining an Inlet.	
	Defining a Route	124
	Defining an Outlet	139
	Defining a Channel	142
	Reusing Your Channel Configuration.	146
A.	Ebix-Supported Transaction Sets	147
	Supported HIPAA Transactions	
В.	Using iWay Integration Tools to Configure an Ebix for HIPAA	149
	Using iIT to Configure an Ebix File for HIPAA Overview	149
	Using iIT to Configure an Ebix File for HIPAA Prerequisites	149
	Downloading and Extracting an Ebix	149
	Working With iWay Integration Tools (iIT)	150
C.	Using HIPAA Special Register (SREG) Types	171
	HIPAA Special Register (SREG) Types	171
D.	Using HIPAA Separators and Terminators	
	HIPAA Separators and Terminators	175
Е.	Sample HIPAA Files	177
	- Sample HIPAA Message	
	Sample 997 Functional Acknowledgment	
	Sample Output XML Document	
	Sample Output Validation Report	
F.	Tutorial: Adding a Detail Line Counter to a Purchase Order Transform	183
	Configuring the Required Variables	
	Using the Graphical Mapping Builder	

Contents

This documentation describes how to configure and use the iWay Integration Solution for HIPAA. It is intended for developers to enable them to parse, transform, validate, store, and integrate health care information into the existing enterprise and pass information electronically, to partners, in HIPAA-mandated format.

Note: This Release 7.0.x content is currently being updated to support iWay Release 8.0.x software. In the meantime, it can serve as a reference for your use of iWay Release 8. If you have any questions, please contact *Customer_Success@ibi.com*.

How This Manual Is Organized

This manual includes the following chapters:

	Chapter/Appendix	Contents
1	Introducing the iWay Integration Solution for HIPAA	Describes the mandate of the Health Insurance Portability and Accountability Act (HIPAA) and how the components of the iWay Integration Solution for HIPAA streamline the flow of information between health care partners. Provides a roadmap to information on other products used with the iWay Integration Solution for HIPAA.
2	Deployment Information for Your iWay Integration Solution	Describes the iWay products used with your iWay Integration Solution for HIPAA and provides a roadmap to full information on those products. Introduces the concept of a channel for the construction of a message flow in iWay Service Manager.
3	Configuring the EDI Activity Driver	Describes how to configure the EDI Activity Driver using iWay Service Manager.
4	Working With HIPAA Inbound and Outbound Applications Using iWay Integration Tools (iIT)	Describes how to work with HIPAA inbound and outbound applications using iWay Integration Tools (iIT).

	Chapter/Appendix	Contents
5	Inbound Processing: HIPAA to XML	Includes an overview of the iWay business components and processing steps in a basic inbound message flow. The message flow converts a document from HIPAA format to XML format. Also includes instructions for configuring a basic inbound message flow.
6	Outbound Processing: XML to HIPAA	Includes an overview of the iWay business components and processing steps in a basic outbound message flow. The message flow converts a document from XML format to HIPAA format. Also includes instructions for configuring a basic outbound message flow.
A	Ebix-Supported Transaction Sets	Describes the HIPAA transaction sets supported by the iWay Integration Solution for HIPAA in the Ebix files supplied with the product.
В	Using iWay Integration Tools to Configure an Ebix for HIPAA	Describes how to use iWay Integration Tools (iIT) to configure an e-Business Information Exchange (Ebix) file for HIPAA.
С	Using HIPAA Special Register (SREG) Types	Describes the Special Register (SREG) types that are created during HIPAA to XML transactions and 997/999 creation.
D	Using HIPAA Separators and Terminators	Includes a list of separators and terminators that are allowed.
E	Sample HIPAA Files	Includes a sample HIPAA input document in Electronic Data Interchange (EDI) format, output XML document, and a 997 Functional Acknowledgment that results from inbound processing.
F	Tutorial: Adding a Detail Line Counter to a Purchase Order Transform	Provides a tutorial that describes how to add a detail line counter, such as a variable, to a purchase order transform.

Documentation Conventions

The following table describes the documentation conventions that are used in this manual.

Convention	Description
THIS TYPEFACE or this typeface	Denotes syntax that you must enter exactly as shown.
this typeface	Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.
underscore	Indicates a default setting.
Key + Key	Indicates keys that you must press simultaneously.
{}	Indicates two or three choices. Type one of them, not the braces.
	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis ().
	Indicates that there are (or could be) intervening or additional commands.

Related Publications

Visit our Technical Documentation Library at *http://documentation.informationbuilders.com*. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have any questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing tips and techniques. Access Focal Point at http://forums.informationbuilders.com/eve/forums.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, *http:// www.informationbuilders.com*. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of *http://www.informationbuilders.com* also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (xxxx.xx) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following tables list the environment information our consultants require.

Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following table lists the deployment information our consultants require.

Adapter Deployment	For example, JCA, Business Services Provider, iWay Service Manager
Container	For example, WebSphere

Version	
Enterprise Information System (EIS) - if any	
EIS Release Level	
EIS Service Pack	
EIS Platform	

The following table lists iWay-related information needed by our consultants.

iWay Adapter	
iWay Release Level	
iWay Patch	

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Did the problem arise through a service or event?	
Provide usage scenarios or summarize the application that produces the problem.	
When did the problem start?	
Can you reproduce this problem consistently?	
Describe the problem.	
Describe the steps to reproduce the problem.	
Specify the error message(s).	

Request/Question	Error/Problem Details or Information
Any change in the application environment: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem <i>not</i> occur?	

The following is a list of error/problem files that might be applicable.

- Input documents (XML instance, XML schema, non-XML documents)
- □ Transformation files
- Error screen shots
- Error output files
- Trace files
- Service Manager package to reproduce problem
- Custom functions and agents in use
- Diagnostic Zip
- Transaction log

For information on tracing, see the *iWay* Service Manager User's Guide.

User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. Please use the Reader Comments form at the end of this document to communicate your feedback to us or to suggest changes that will support improvements to our documentation. You can also contact us through our website, *http://documentation.informationbuilders.com/connections.asp.*

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website (*http://education.informationbuilders.com*) or call (800) 969-INFO to speak to an Education Representative.



Introducing the iWay Integration Solution for HIPAA

The iWay Integration Solution for HIPAA transforms HIPAA documents into standard XML format, or transforms XML representations into HIPAA format.

This section provides an overview of HIPAA and describes the features that are provided by the iWay Integration Solution for HIPAA.

In this chapter:

- The HIPAA Mandate
- Promoting HIPAA Compliance and Integration
- Features of the iWay Integration Solution for HIPAA
- HIPAA Information Roadmap

The HIPAA Mandate

The U.S. Congress enacted the Health Insurance Portability and Accountability Act (HIPAA) to reform the health insurance market. HIPAA requires national, uniform standards for the electronic transmission of health care information to simplify the health care administration and financial processes.

The iWay Integration Solution for HIPAA, based on these standards, promotes the integration of over 200 enterprise data and application systems.

The Health Insurance Portability and Accountability Act of 1996 (Public Law 104-191, known as HIPAA) includes a provision for Administrative Simplification. This provision required the Secretary of the Department of Health and Human Services to adopt standards to support the electronic exchange of administrative and financial health care transactions, primarily between health care providers and health care plans.

HIPAA mandates the adoption of standards for such transactions and defines specifications for implementing each standard. The iWay Integration Solution for HIPAA is based on the addenda version of HIPAA 004010, which was released in October, 2002, and is referred to as 004010A1.

Achieving Administrative Simplification

Administrative Simplification means adopting uniform business practices (billing, computer systems, and communication) so that providers and payers can easily interact through one another's proprietary systems.

The Administrative Simplification provisions of HIPAA standardize forms and methods of completing claims, and other payment-related documents, and assign a universal identifier to health care providers. These provisions serve to increase computer use and efficiency when exchanging health care information.

HIPAA addresses the following areas of Administrative Simplification:

- Electronic Data Interchange (EDI) is the electronic transfer of information between trading partners in a standard format. It enables partners to exchange information and transact business quickly and efficiently. HIPAA includes standard electronic formats for transactions such as enrollment, eligibility, payment and remittance advice, claims, health plan premium payments, health claim status, and referral certification and authorization.
- **Code sets** include data elements that uniformly document why patients are seen and what treatment (procedures) they receive during health care encounters.
- □ Identifiers are numbers that identify health care providers, health plans, employers, and individuals (patients). These identifiers simplify administrative processes such as referrals and billing, improve accuracy of data, and reduce costs.
- □ Security refers to the mandatory standards developed and adopted for all health plans, clearing houses, and providers. Everyone involved in the health care industry is required to comply at all stages of transmission and storage of health care information. This compliance ensures the integrity and confidentiality of the records at all phases of the process.
- □ **Privacy** refers to standards that define appropriate and inappropriate disclosures of individually identifiable health information and protection of patient rights.

The benefits of Administrative Simplification include:

- Lowered administrative costs
- Enhanced accuracy of data and reports
- Increased customer satisfaction
- Reduced cycle time
- Improved cash management

Promoting HIPAA Compliance and Integration

The iWay Integration Solution for HIPAA enables health care providers to integrate internal patient care and financial systems with external trading partner systems, in compliance with the HIPAA mandate. It enables secure and auditable business-to-business processes and information exchange regardless of document format.

The iWay Integration Solution for HIPAA:

- Enables applications to receive and publish HIPAA transactions across TCP/IP, HTTP, and IIOP networks.
- ❑ Shares the iWay e-Business Information Exchange (Ebix) framework. An Ebix provides an archiving executable solution for e-business metadata components, which allows integration with iWay Service Manager. The iWay Integration Solution for HIPAA uses the metadata in the Ebix that is compliant with the HIPAA standard for all document processing.
- Establishes a common development environment inside multiple message brokers and application servers including IBM WebSphere Business Integration Message Broker, Microsoft Commerce Server, Microsoft BizTalk Server, and Oracle 9iAS.

Legacy Integration

The iWay Integration Solution for HIPAA supports over 200 enterprise data and application systems, simplifying and accelerating HIPAA compliance regardless of the diversity of the backend environment. It integrates legacy applications that use different platforms, operating systems, and databases, as well as software used by facilities such as reference labs and imaging centers. It allows users with different communication protocols, APIs, front-end environments, and security frameworks to communicate, without requiring custom coding.

Legacy applications typically include CICS, VSAM, or IMS. The move to distributed computing has resulted in disparate applications based on AS/400, HP3000, and UNIX, such as MUMPS, Ingres, and Informix, which are supported in the integration solution. The iWay Integration Solution for HIPAA also protects your investment in packaged Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), and Supply Chain Management (SCM) applications.

Features of the iWay Integration Solution for HIPAA

The standards-based iWay Integration Solution for HIPAA reduces the amount of effort it takes to integrate HIPAA documents with your internal enterprise applications and third-party trading partners. It includes conversion and validation of documents from HIPAA to XML format, making it easy to include HIPAA documents in your XML-based integration projects.

Features of the iWay Integration Solution for HIPAA include:

- Integration with iWay Service Manager to provide bi-directional (synchronous and asynchronous) conversion of HIPAA formats and XML. Interactions between application servers, integration brokers, third-party software packages, and messaging services are also supported.
- Integration with iWay Trading Partner Manager to provide routing, custom transformation by document, and other value-added features.
- Integration with more than 200 other information assets, including J2EE-based back-office systems; data structures such as DB2, IMS, VSAM, and ADABAS; and front-office systems based on Sybase.
- Integration with leading application servers, integration brokers, and development environments. Supported software platforms include BEA WebLogic, IBM WebSphere, Sun Java Enterprise System, and Oracle Application Server.
- □ Out-of-the-box support for HIPAA ANSI X12 4010A1 transaction sets. For details on the supported transaction sets, see *Ebix-Supported Transaction Sets* on page 147.
- Reusable framework for parsing, transforming, and validating HIPAA documents without the need to write custom code.
- Data dictionary approach that facilitates HIPAA-to-XML transformations. The iWay Integration Solution for HIPAA uses dictionaries to transform data from HIPAA format to any other format, or from any format to HIPAA format. It supports flat files, comma-delimited files, popular relational database formats, XML, and more.
- Pre-built data dictionaries, XML schemas, transformation templates, and rule files for automatic transformation and validation of input and output documents.

HIPAA Information Roadmap

The following table lists the location of deployment and user information for products used with the iWay Integration Solution for HIPAA.

Product	For more information, see
iWay Service Manager	Chapters 3, 4, 5, and 6 of this guide
	iWay Service Manager User's Guide

Product	For more information, see
iWay Transformer	iWay Transformer User's Guide
iWay Integration Tools (iIT) Designer	iWay Integration Tools Designer User's Guide

iWay Integration Tools (iIT) Designer (previously known as iWay Designer) is a GUI tool that is delivered as a plugin with iIT. For more information, see the *iWay Integration Tools Designer User's Guide*.



Deployment Information for Your iWay Integration Solution

This topic describes the iWay products used with your iWay Integration Solution for HIPAA and provides a roadmap to full information on those products.

It also introduces the concept of a channel for the construction of a message flow in iWay Service Manager.

In this chapter:

- iWay Products
- Using a Channel to Construct a HIPAA Message Flow
- Components of the iWay Integration Solution for HIPAA

iWay Products

Your iWay integration solution works in conjunction with one or more of the following products:

- □ iWay Service Manager
- iWay Transformer

iWay Service Manager

iWay Service Manager is an open transport service bus that provides graphical tools to create sophisticated integration services without writing custom integration code by:

- Using metadata from target applications
- Transforming and mapping interfaces
- Managing stateless processes

Its capability to manage complex integration interactions makes it ideally suited to be the foundation of a service-oriented architecture.

For more information, see the IWay Service Manager User's Guide.

iWay Transformer

iWay Transformer is a rule-based data transformation tool that converts an input document of one data format to an output document of another data format or structure. The easy-to-use graphical user interface and function tool set facilitate the design of transform projects that are specific to your requirements.

For more information, see the *iWay Transformer User's Guide*.

Understanding Pipelined Transformations

A standard transformation process builds an internal output tree in order to manipulate the tree structure for complex transformations. However, this is not always necessary if the transformation requires only one-to-one mapping.

The definition of one-to-one mapping is that every input will be transformed to the corresponding output document and no function or explicit configuration will be used for the data (or tree) manipulation. If these conditions are met, then very fast and memory effective transformation can be achieved.

A pipelined transformation process is an implementation of one-to-one mapping, which is currently available for HIPAA to XML transformations. The current pipelined process will be enhanced to provide support for both directions for the formats that are listed in the following table.

Format	Non-XML > XML Pipeline	XML > Non-XML Pipeline
HIPAA	Yes	Yes

Since pipelining will be available in both directions, it has been decided to add the pipelined transformation process as an option when running transformations for Ebix components. Ebix components will:

- □ Use one set of metadata for pipelined and standard transformations.
- Have a flag for the run time mode of the transformation process, either Standard (default) or Pipelined.

Output Node Name

Since the pipelined transformation process does not read mapping definitions, output node names are formatted from the data dictionary. The name pattern for the node is identical to the one used to build Ebix metadata, known as long_names, which also depends on the data format being used.

For more information on using pipelined transformations, see the *iWay Transformer User*'s *Guide*.

Error Handling

The error handling functionality is identical to the standard transformation process.

ilT Designer

The capability of graphically visualizing a business process is a powerful and necessary component of any e-Business offering. iIT Designer, a Windows-based design-time tool, provides a visual and user-friendly method of creating a business process, also called a process flow. Through a process flow, you control the sequence in which tasks are performed and the destination of the output from each task.

For more information, see the *iWay Integration Tools Designer User's Guide*.

Using a Channel to Construct a HIPAA Message Flow

The use of iWay Service Manager utilizes a channel, which is a container for all the iWay business components used in a HIPAA message flow.

At a high level, a channel accepts input data via an **inlet**, processes the data via a **route**, and outputs the resulting data via an **outlet**. Another component in the process is an **Ebix** (e-Business Information Exchange). An Ebix provides an archiving executable solution for e-business metadata components, which allows integration with iWay Service Manager for end-to-end e-business document processing.

The following diagram shows the channel components available in the construction of a message flow.

In the following diagram, the value **n** underneath a component name indicates how many instances of that component you can have in a channel configuration—zero, one, or more than one. For example, n = 1 for Inlet means that you can have only one inlet on the channel.



Required components are in boldface type.

Components of a Channel

A channel consists of:

- □ An inlet, which defines how a message enters a channel.
- □ A route, which defines the path a message takes through a channel.
- □ An outlet, which defines how transformed messages exit a channel.
- ❑ An e-Business Information Exchange (Ebix), which is a collection of metadata that defines the structure of data.

iWay Service Manager provides a design-time repository called the Registry, where you assemble and manage the components in a channel.

An inlet can contain:

- A listener (required), which is a protocol handler responsible for picking up an incoming message on a channel.
- A decryptor, which applies a decryption algorithm to an incoming message and verifies the security of the message.
- ❑ A preparser, which is a logical process that converts an incoming message into a processable document. The preparsed document then passes through the standard transformation services to reach the designated processing service.
- A route can contain:
- An in transformer, which is an exit sequence that applies to a message before processing occurs.
 - ❑ A reviewer, which is either the first exit to receive a document after parsing (inbound), or the last exit to receive a document prior to the actual emit operation (outbound). These exits are intended for envelope handling but can be used for any desired purpose.
 - ❑ Validation rules, which apply validation using the rules validation engine. Rules are provided when the iWay Integration Solution for HIPAA is installed.
 - ❑ A transform, which is a transformation definition file that contains sets of rules, interpreted and executed by a transformation engine. Transformation is the process by which data is transformed from one structure/format to another.
- ❑ A process, which is a stateless, lightweight, short-lived microflow that is executed by iWay Service Manager on a message as it passes through the system. Processes that are published using iIT Designer are available in the Registry and can be bound to channels as routes.
 - □ A process flow process.
 - An agent list.
 - A service, which is an executable Java procedure that handles the business logic of a message.

- ❑ An adapter, which refers to a target that represents a specific instance of a connection to a back-end system.
- An out transformer, which is an exit sequence that applies to a message after processing occurs.
 - ❑ A transform, which is a transformation definition file that contains sets of rules, interpreted and executed by a transformation engine. Transformation is the process by which data is transformed from one structure/format to another.
 - ❑ Validation rules, which apply validation using the rules validation engine. Rules are provided when the iWay Integration Solution for HIPAA is installed.
 - ❑ A reviewer, which is either the first exit to receive a document after parsing (inbound), or the last exit to receive a document prior to the actual emit operation (outbound). These exits are intended for envelope handling but can be used for any desired purpose.
- ❑ An outlet (optional), which is responsible for all aspects of preparing a document for emission and then emitting it.
 - A preemitter, which is a logical process that handles a document immediately before transmission. Normally it converts an XML document into non-XML format.
 - □ An encryptor, which can be called to encrypt an outgoing document.
 - □ An emitter, which is a transport protocol that sends a document to its recipient.
- An **outlet** can contain:
- A preemitter.
- An encryptor.
- Multiple emitters.

For details on the preceding components, see the *iWay* Service Manager User's Guide.

Components of the iWay Integration Solution for HIPAA

iWay business components used in the construction of a message flow for HIPAA transactions include:

- Ebix (e-Business Information Exchange)
- Preparser
- Acknowledgement service

- Deidentification service
- Preemitter

Ebix

iWay Software provides various e-Business Information Exchange (Ebix) files used in conjunction with the iWay integration solutions. In iWay Service Manager, the iWay Integration Solution for HIPAA contains an Ebix file for the supported HIPAA version.

An Ebix file for HIPAA is named HIPAA_Version.ebx, where Version is the HIPAA version number. For example, the Ebix file for HIPAA version 4010A1 is named HIPAA_4010A1.ebx.

For details on the supported HIPAA transaction sets, see *Ebix-Supported Transaction Sets* on page 147.

An Ebix is a collection of metadata that defines the structure of data. The Ebix supplied with the iWay Integration Solution for HIPAA defines the structure of supported HIPAA messages.

Each Ebix includes:

- Pre-built data dictionaries. The structure of each HIPAA document is described by two data dictionaries:
 - □ Header dictionary, which describes the enveloping structure of the document.
 - Document dictionary, which describes the segments and elements that compose each document.

The dictionaries from the Ebix are used to transform the structure of a document per HIPAA regulation.

- XML schemas that define the structure and content of the HIPAA messages in iWay XML format.
- □ HIPAA to XML transformation templates, and XML to HIPAA templates, for the supported HIPAA transaction sets.
- Rule files for each message. The iWay Integration Solution for HIPAA uses these rule files to validate inbound and outbound documents.

Preparsers

A preparser is an iWay business component that converts a non-XML document into XML format. The preparser for the iWay Integration Solution for HIPAA converts an incoming HIPAA formatted document to iWay XML format.

The HipaaSplitterPreParser is provided by iWay Software for the iWay Integration Solution for HIPAA.

HipaaSplitterPreParser

The HipaaSplitterPreParser (com.ibi.preparsers.HIPAASplitPP) parses a HIPAA input file that contains one or more interchanges (ISA) and multiple documents, and creates multiple XML output files. One XML output file is produced for each document.

For example, if the HIPAA input file contains three documents within one ISA, the HipaaSplitterPreParser creates three XML output files, one per document.

Use the HipaaSplitterPreParser for large files with multiple documents within one ISA; if there is a specific business requirement to create separate XML output files; or if you receive multiple documents within one ISA and want to separate each document for further business processing. You can also use the HipaaSplitterPreParser if there is only one document within the ISA.

HIPAABatchSplitter

The HIPAABatchSplitter (com.ibi.preparsers.XDHIPAABatchSplitter) parses a HIPAA input file that contains one or more interchanges (ISA) and multiple documents. You must use this preparser with the HIPAAPreParser (com.ibi.preparsers.XDHIPAApreParser). The HIPAABatchSplitter should not be used as a standalone preparser. To successfully transform an inbound HIPAA file using this preparser, you must also include the HIPAAPreParser in your channel inlet.

One XML output file is produced for each document that is processed through this inlet definition. For example, if your HIPAA input contains three documents within an ISA, the HIPAABatchSplitter / HIPAAPreparser will create three XML output files, one for each document.

Acknowledgement Service

An acknowledgement service is an iWay business component used in inbound processing to create a functional acknowledgement (997 or 999) for inbound messages.

An acknowledgement indicates that an inbound document was received and validated for structure against the appropriate standard. An acknowledgement does not indicate that a document was processed.

An acknowledgement is typically routed back to the originator of the inbound document or to the next step in the integration process. It is a best business practice to send an acknowledgement to the originator of the inbound document. The acknowledgement service for the iWay Integration Solution for HIPAA is called HipaaAckAgent (com.ibi.agents.XDHipaaAckAgent).

Deidentification Service

The Deidentification service (com.ibi.agents.XDDeidentifyAgent) provides algorithms that can be used to implement the deidentification of protected health information in accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. Multiple algorithms can be configured since a combination of algorithms will be needed to deidentify the data correctly.

The Deidentification service takes an XML document as input. The first configured algorithm takes this document as input and modifies it in place. The result is fed into the next configured algorithm and so on. The result of the last configured algorithm is the XML document returned by the service.

For more information on configuring and using the Deidentification service, see the *iWay* Service Manager Component Reference Guide.

Preemitter

A preemitter is a logical process that handles a document immediately before transmission.

Typically a preemitter is used to convert an XML document to non-XML format. The iWay Integration Solution for HIPAA uses a preemitter in outbound processing to convert the XML-formatted HIPAA document to a HIPAA formatted document.

The XML structure must be compliant with the schema supplied in the Ebix.

The preemitter for the iWay Integration Solution for HIPAA is called XDHIPAAPreEmitter (com.ibi.preemit.XDHIPAAPreEmitter).

Data Segments and Data Elements

The following example provides a sample 835 Health Care Claim/Payment Advice document. Each line is called a *Data Segment* and begins with the *Segment Name*. For example, 'N1' represents the payer name and identification while 'CLP' represents the claim payment information.

```
ISA*00*
                 *00*
                               *ZZ*SUBMITTER ID *ZZ*RECEIVER
ID *020423*1453*U*00401*000000905*1*T*:
GS*HP*SENDER CODE*RECEIVER CODE*20020423*1455*1*X*004010X091A1
ST*835*998877
BPR*C*750*C*ACH*CTX*01*998877661*DA*338899*1872994342*8856666666*01*11223
445*DA*143453454444*20020422
TRN*1*99887766554433*1872994342
REF*EV*R94395
DTM*405*20020423
N1*PR*BCBS MO
N3*123 MAIN STREET
N4*KANSAS CITY*MO*64108
N1*PE*GENERAL HOSPITAL*FI*871234599
N3*123 GENERAL ST
N4*DALLAS*TX*75043
LX*1
CLP*111222333*1*800*750*0*12*987654321
CAS*CO*A2*50
NM1*OC*1*DOE*JOHN*W***34*446928421
SVC*HC:99211*800*800
DTM*150*20020422
DTM*151*20020422
SE*19*998877
GE*1*1
IEA*1*00000905
```

Following the Segment Name are a number of *Data Elements*. In the N1 segment, the code 'PR' stands for payer name and address. Data elements are separated by a single character, usually an asterisk (*). A segment ends with a single character– in this example a tilde (~).

Other HIPAA documents such as an 820 Payment Order/Remittance Advice document will have their own sets of data segments and data elements. Segments such as the N1 overlap many transaction sets, but an 820 or 835 will have its own segments and elements that are unique to healthcare.

Any number of data segments come together to form a transaction set. In this example, there are 19, as shown in the control counter stored in the very last segment (SE).

Please note that the layout of an 835 Health Care Claim/Payment Advice document that is sent from insurance company #1 to a healthcare provider #2 will be different from the one that is sent by insurance company #3. As a result, healthcare providers must be able to process different 835 document layouts.



Configuring the EDI Activity Driver

This section describes how to configure the EDI Activity Driver using iWay Service Manager.

In this chapter:

- □ HIPAA EDI Activity Driver Overview
- Configuring the EDI Data Provider Using iWay Service Manager
- Configuring the EDI Activity Driver Using iWay Service Manager

HIPAA EDI Activity Driver Overview

The EDI Activity Driver is an extension of the Activity Facility in iWay Service Manager. It is used to log events as messages are processed. Logging can occur when:

- □ a message is acquired.
- □ a message is emitted.
- an error occurs.
- □ a component such as an agent or process flow is called.

For more information about the Activity Facility, see the iWay Service Manager User's Guide.

Using iWay Service Manager, you must first configure the EDI data provider and then the Activity Facility handler.

Configuring the EDI Data Provider Using iWay Service Manager

This section describes how to configure the EDI data provider.

Procedure: How to Configure the EDI Data Provider

To configure the EDI data provider:



1. In the left console pane of the Server menu, select *Data Provider*.

iWay Integration Solution for HIPAA User's Guide

The Data Provider pane opens.

Data Provider

Listed below are the data provider definitions that are available in the base configuration of this server.

- JDBC Connections - JDBC or Java Database Connectivity is a standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call-level API for SQL-based database access. The listings below define JDBC connections used within iWay Service Manager. iWay components that use JNDI can access a JDBC provider as a DataSource by setting the initial context factory to com.ibi.jndi.XDInitialContextFactory and using the name jdbc/provider name.			
Name		Driver	
No connections have been define	No connections have been defined		
New JLINK Servers - JLINK is a technology that can be used to access information hosted by iWay, WebFOCUS and EDA data servers. The servers listed below are defined for use with JLINK.			
Name	Name Description Type		Туре
No servers have been defined			
New			

The tables that are provided list the configured JDBC and JLINK data providers that are available. By default, no data providers are configured.

2. In the JDBC area, click New to configure a new JDBC data provider.

JDBC Connection Pool Pro	operties
Name *	Enter the name of the JDBC data provider to add.
	EDI_Activity_DB
Driver Class	The JDBC driver class is the name of the class that contains the code for this JDBC Driver.
	com.mysql.jdbc.Driver
	Select a predefined database or enter your own.
Connection URL	The JDBC connection URL to use when creating a connection to the target database. The URL generally includes the server name or IP address, the port or service, the data source name, and a driver specific prefix.
	jdbc:mysql://localhost:3306/IWay
	Select a predefined connection URL template or enter your own.
User	User name with respect to the JDBC URL and driver.
	iway
Password	Password with respect to the JDBC URL and driver.
	••••
Connection Pool Properti	es
Initial Pool Size *	Number of connections to place in the pool at startup.
	1
Maximum Number of Idle Connections *	Maximum number of idle connections to retain in the pool. O means no limit except what is enforced by the maximum number of connections in the pool.
	1
Maximum Number of	Maximum number of connections in the pool. O means no limit.
Connections "	1
Login Timeout	Time in seconds to wait for a pooled connection before throwing an exception. O means wait forever.

The configuration pane for the JDBC data provider opens.

- 3. In the Name field, enter a name for the new JDBC data provider, for example, EDI_Activity_DB.
- 4. From the Driver Class drop-down list, select an appropriate driver or enter the specific driver name (class) that you are using, for example:

```
com.mysql.jdbc.Driver
```

5. From the Connection URL drop-down list, select an appropriate connection URL or enter the specific driver connection URL that you are using, for example:

jdbc:mysql://localhost:3306/IWay

- 6. In the User field, enter a user name with respect to the JDBC URL and driver.
- 7. In the Password field, enter a password with respect to the JDBC URL and driver.
- 8. In the Initial Pool Size field, enter the number of connections to place in the connection pool during startup.

9. In the Maximum Number of Idle Connections field, enter the maximum number of idle connections to retain in the connection pool.

A value of zero means that there is no limit, except what is enforced by the maximum number of connections in the connection pool.

10. In the Maximum Number of Connections field, enter the maximum number of connections in the connection pool.

A value of zero means that there is no limit.

11. Click Add.

The JDBC data provider that you configured is added to the JDBC Connections list, as shown in the following image.

Data Provider

Listed below are the data provider definitions that are available in the base configuration of this server.

JDBC			
Connections - JDBC or Java Database Connectivity is a standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call-level API for SQL-based database access. The listings below define JDBC connections used within iWay Service Manager. iWay components that use JNDI can access a JDBC provider as a DataSource by setting the initial context factory to com.ibi.jndi.XDInitialContextFactory and using the name jdbc/provider name.			
Name		Driver	
EDI_Activity_DB		com.mysql.jdbc.Driver	
New Delete Rename Copy			
Servers - JLINK is a technology that can be used to access information hosted by iWay, WebFOCUS and EDA data servers. The servers listed below are defined for use with JLINK.			
Name	Description		Туре
No servers have been defined			
New			

Configuring the EDI Activity Driver Using iWay Service Manager

This section describes how to configure the EDI Activity Driver.

Procedure: How to Configure the EDI Activity Driver

To configure the EDI Activity Driver:



1. In the left console pane of the Server menu, select Activity Facility.

The Activity Facility pane opens.

Activity Facility Listed below are the activity (sometimes called audit) handlers that have been co server has to be stopped and started for any change to take effect. Configured Activity Handlers	onfigured. You can add to this li	st or delete from it. The
Name Type Active		
No activity handlers have been defined		
Add		

The table that is provided lists the configured Activity Facility handlers. Initially, no handlers are shown.

2. Click Add to configure a new Activity Facility handler.

Activity		
Туре	The type is the specific class of handler in use	
	EDI Activity Logs	
Name	The handler will be known by this name in the system. Names must be unique.	
	EDI Activity Logger	
Description	Describe the purpose of this handler	
Active	Active handlers perform work in the server. Inactive handlers remain defined but are not used during this server run. To change the active state, after updating you must cold restart the server. true Pick one V	

The configuration pane for the Activity Facility handler opens.

- 3. From the Type drop-down list, select EDI Activity Logs.
- 4. Enter a unique name for the EDI Activity Driver and a brief description.
- 5. From the Active drop-down list, select *true*.

6. Configure the JDBC driver for the database you are using.

Configuration Parameter	rs
JNDI Factory Name	JNDI initial context factory class used to access data source. Use com.ibi.jndi.XDInitialContextFactory for an iWay JDBC provider or leave blank for JVM default.
	com.ibi.jndi.XDInitialContextFactory
JNDI Name *	JNDI Name for the data source this driver will use. To use an iWay JDBC provider, enter the JNDI name as jdbc/provider name otherwise the defined provider's information will be used.
	jdbc/EDI_Activity_DB
Table *	Table name to which to write log.
	IAM_ACTIVITY
Compression	What form of compression, if any, should be used on the messages. Compression saves space at the expense of time.
	none
	Pick one

If the database tables do not exist, they will be automatically created when the iSM is restarted.

7. Provide values for the remaining parameters, as defined in the following table.

Parameter Name	Туре	Description
JNDI Factory Name	String	The JNDI initial context factory class that is used to access a data source. Use com.ibi.jndi.XDInitialContextFactor y for an iWay JDBC provider or leave this field blank for the JVM default.
JNDI Name	String	The JNDI name for the data source this driver will use. To use an iWay JDBC provider, enter the JNDI name as jdbc/< <i>data provider</i> <i>name</i> >, where <i>data provider name</i> is the name of the EDI Activity Driver that was specified in step 4. Otherwise the information for the defined provider will be used.
Table	String	Table name for the activity log. This must be a valid identifier in the database being used. If the table does not exist at startup, it will be created automatically.
Parameter Name	Туре	Description
--------------------------	---------------------------	--
Compression	Drop-down list	Specify whether the messages are to be compressed. Values include:
		none (default)
		□ smallest
		□ fastest
		standard
		Huffman
Start Events	Boolean Drop-down list	If set to <i>true</i> (default), the input messages will be recorded in the activity log. This value must be set to <i>true</i> for use of the audit reports in the console.
Internal Events	Boolean Drop-down list	If set to <i>true</i> , system events are included in the activity log. System events include activities such as parsing and transformations (optional). False is selected by default.
Security Events	Boolean Drop-down list	If set to <i>true</i> (default), security events are recorded. This includes digital signature, and so on. However, console activity is not recorded.
Business Error Events	Boolean Drop-down list	If set to <i>true</i> , business errors are recorded, such as rules system violations. False is selected by default.
Emit Events	Boolean Drop-down list	If set to <i>true</i> (default), output messages from emitter services will be recorded. This is required for use of the audit log reports in the console.

Parameter Name	Туре	Description
End Events	Boolean Drop-down list	If set to <i>true</i> (default), the end of message processing will be recorded in the activity log. This is required for use of the audit log reports in the console.
Notes Table	String	Table name for the notes table, which contains log annotations. If the table does not exist at startup, it will be created automatically.
MAC Algorithm	String Drop-down list	The Message Authentication Code (MAC) algorithm. None (default) indicates a MAC should not be computed.
MAC Provider	String Drop-down list	The Message Authentication Code (MAC) provider. Not Specified indicates the default provider should be used. The remaining available value is <i>SunJCE</i> .
MAC Secret Key	String	The Message Authentication Code (MAC) secret key to use.

8. Click Update.

If necessary, start the database services.

9. Restart iSM to start the EDI Activity Driver and begin logging.

The EDI Activity Driver inserts records into the configured activity database. The records are designed for fast writing rather than for ease of later analysis. A set of inquiry service agents suitable for use in a process flow is available to assist during the analysis of the log. Users are cautioned that iWay does not guarantee the layout of the record from release to release, and this should be checked against the actual schema.

Database Field	Description
recordkey	Unique record identifier.

Database Field	Description
recordtype	Type of this record - the event being recorded.
	101 - Message start.
	131 - Entry to event (see subtype codes below).
	132 - Normal exit from event.
	133 - Failed exit from event.
	151 - Ancillary message (usually rules violation).
	□ 181 - Emit.
	191 - Message end.
signature	Encoding of the listener name and protocol.
protocol	Name of the protocol.
address	Address to which an emit is to be issued. The format depends on the protocol.
tstamp	Timestamp of record.
correlid	ISA13
tid	Transaction ID assigned to this message.
msg	Message appropriate to this record type. For example, an input message contains the original message received, if possible. Streaming input does not contain a record.
context	Serialized special registers that were in the context at the time the record was written.
text	Message text for business errors (rules system violations).

Database Field	Description
status	Status code recorded.
	□ 0 - Success
	1 - Success, message end (191 record)
	10 - Rules error
subtype	Event code for event records.
	1 - Preparser
	2 - Parser
	□ 3 - In reviewer
	□ 5 - In validation
	□ 6 - In transform
	□ 7 - Agent or flow
	8 - Out transform
	9 - Out validation
	□ 11 - Preemitter
	1000 - input record written to table before transformation
partner_to	ISA06
partner_from	ISA08
encoding	Encoding of the listener that obtained the document.
mac	Not used in this version.
Driver version	1.0 in 8.0 SM



Working With HIPAA Inbound and Outbound Applications Using iWay Integration Tools (iIT)

This chapter describes how to work with HIPAA inbound and outbound applications using iWay Integration Tools (iIT).

In this chapter:

- □ HIPAA Inbound and Outbound Applications Overview
- HIPAA Inbound and Outbound Applications Prerequisites
- Extracting HIPAA User Samples
- Importing HIPAA User Samples to iWay Integration Tools as a Workspace
- Devision Publishing iWay Integration Applications to the iWay Service Manager Registry
- Deploying iWay Integration Applications to iWay Service Manager
- Setting Registers in the iWay Service Manager Administration Console
- Stopping Inbound (HIPAA to XML) and Outbound (XML to HIPAA) Processing
- Testing the Sample HIPAA Applications

HIPAA Inbound and Outbound Applications Overview

This chapter provides instructions to create, import, export, and work with HIPAA inbound and outbound applications using iWay Integration Tools (iIT). In addition, you will learn how to create an iWay Integration Application (iIA) for deployment based on the sample data.

What will the Application do?

The iIAs will be used to transform HIPAA to XML for inbound processing and XML to HIPAA for outbound processing.

The inbound application channel creates an XML representation of a HIPAA (ANSI X12N formatted) inbound message, a functional acknowledgement (997 or 999), and an XML-formatted validation report. The documents are routed to designated folders based on the success or failure results of the transformation and HIPAA validation.

The outbound application channel creates an ANSI X12N formatted HIPAA message from XML.

HIPAA Inbound and Outbound Applications Prerequisites

Before you continue, ensure that the following prerequisites are met:

- You have a working knowledge of iWay Service Manager (iSM) and iWay Integration Tools (iIT).
- □ iSM Version 8.0 is installed.
- □ iWay HIPAA Adapter is installed.
- □ iIT Version 8.0 is installed.
- System and channel Special Registers (SREGs) are updated to match your directory structure, as shown in *How to Extract User Samples for HIPAA* on page 42.

Extracting HIPAA User Samples

This section describes how to extract user samples for HIPAA.

Procedure: How to Extract User Samples for HIPAA

1. Download the Hipaa_usr_samples.zip file containing HIPAA user sample workspace from the following website:

http://techsupport.informationbuilders.com

The downloaded HIPAA_usr_samples.zip contains the following files:

- HIPAA_Accelerator.zip
- Hipaa_usr_samples_ilT_workspace.zip
- 2. Save the Hipaa_usr_samples_iIT_workspace.zip file to a folder on your local drive.

3. Save and extract the HIPAA_Accelerator.zip file to a location where you want to store your data, as shown in the following image.

C:\HIPAA_Accelerator			
	×	Name 🔺	Туре
🖃 🗁 HIPAA_Accelerator	~	HIPAA_in	File Folder
🖃 🚞 HIPAA_in		HIPAA_out	File Folder
포 🚞 IB_Archive			
Error			
🛅 IB_Output			
🛅 IB_Report			
IB_TransformGood			
CB_Archive	_		
CB_Error			
C OB_Output			
CB_Report			
CB_TransformGoo			
🖃 🧰 HIPAA_out			
IB_Archive			
🛅 IB_Error			
🛅 IB_Output			
IB_Report			
IB_TransformGood			
DB_Archive			
CB_Error			
CB_Output			
CB_Report			
OB_TransformGooi			

- 4. The HIPAA_Accelerator.zip file contains sample input and output data that you can use.
 - □ Inbound test data is located in the following folder:

\HIPAA_Accelerator\HIPAA_in\IB_Archive

There are two subfolders, 4010_hipaa and 5010_hipaa.

For example:

🔄 Organize 👻 📗 Vie	ws 👻	1.1		1.1	
Favorite Links	Name -	▼ Date modified	 Type 	▼ Size	
Dec mente	h2704010x092a1c	0 1/17/2014 5:15 AM	DATA File		1K
Documents	h2714010x092a1c	0 1/17/2014 5:15 AM	DATA File		1 K
Pictures	h8204010x061a1c	0 1/17/2014 5:24 AM	DATA File		21
Music	h8344010x095a1c	0 1/17/2014 5:18 AM	DATA File		21
Recently Changed	h8354010x091a1c	0 1/17/2014 5:17 AM	DATA File		31
Organize - III Vie	\A_Accelerator\HIPAA_in\IB_ :ws ▼	Archive\5010_hipaa			
Organize - 111 Vie	AA_Accelerator\HIPAA_in\IB_ ws ▼ Name ◆	Archive\5010_hipaa	- Type	↓ Size	
Organize - III Vie	A Accelerator\HIPAA_in\IB ws Name h837d5010x224a2	Archive\5010_hipaa	 ✓ Type 1 DATA File 	→ Size	21
Organize Vie avorite Links Documents	A Accelerator\HIPAA_in\IB ws Name h837d5010x224a2 h837dp5010x225a	Archive\5010_hipaa	 ✓ Type 1 DATA File 1 DATA File 	↓ Size	21
Organize	AA_Accelerator\HIPAA_in\IB_ ws Name h837d5010x224a2 h837dp5010x225a h837f5010x223a2c	Archive\5010_hipaa	Type DATA File DATA File DATA File DATA File	↓ Size	21
Organize	AA_Accelerator\HIPAA_in\IB_ ws Name Na	Archive\5010_hipaa	Type DATA File DATA File DATA File DATA File DATA File	↓ Size	21
Organize	A Accelerator \HIPAA_in \IB ws Name h837d5010x224a2 h837dp5010x225a h837dp5010x225a2c h2705010x279a1c h2715010x279a1c	Archive\5010_hipaa	 Type DATA File DATA File DATA File DATA File DATA File DATA File 	↓ Size	2 8 14 55
Organize Vie Organize Vie Organize Vie Organize Vie Organize Vie Organize Vie Organize Vie Organize Vie Organize Vie Organize Vie Organize Organize Vie Organize	A Accelerator \HIPAA_in \IB ws Name h837d50 10x224a2 h837d50 10x224a2 h837d50 10x225a h837i50 10x225a 10x225a h27150 10x279a 1ct h27150 10x279a 1ct h27150 10x279a 1ct h27550 10x210c0 1	Archive\5010_hipaa Date modified 1/17/2014 8:48 AM 1/17/2014 8:44 AM 0 1/17/2014 8:44 AM 0 1/17/2014 8:52 AM 1/17/2014 8:52 AM	 Type DATA File 	↓ Size	2 8 14 55 17 2
Organize Vie Vie Vie Organize Vie Organize Vie Vie Vie Vie Vie Vie Vie	A Accelerator \HIPAA_in \IB Name * h837d50 10x224a2 h837d50 10x225a2 h837d50 10x225a2c h837i50 10x229a 1ct h27050 10x279a 1ct h27050 10x279a 1ct h27550 10x210c01 h27550 10x210c01	Archive\5010_hipaa Date modified 1/17/2014 8:48 AM 1/17/2014 8:44 AM 0 1/17/2014 8:44 AM 0 1/17/2014 8:52 AM 0 1/17/2014 8:52 AM 1/17/2014 8:52 AM 1/17/2014 8:49 AM	Type DATA File DATA File DATA File DATA File DATA File DATA File DATA File	↓ Size	21 81 141 551 17 21
Organize Vie Or	A Accelerator \HIPAA in \IB Name N	Archive\5010_hipas	Type DATA File DATA File DATA File DATA File DATA File DATA File DATA File DATA File	↓ Size	21 81 141 551 171 21 11
Organize Vie avorite Links Documents Pictures Music Recently Changed Searches Public	A Accelerator \HIPAA in \IB Name N	Archive\5010_hipas	Type DATA File DATA File DATA File DATA File DATA File DATA File DATA File DATA File DATA File	↓ Size	21 81 141 551 171 11 11

Outbound test data is located in the following folder:

\HIPAA_Accelerator\HIPAA_out\OB_Archive

There are two subfolders, 4010_xml and 5010_xml.

For example:

🕘 Organize 🔻 📗 Vie	ws 🔻				
Favorite Links	Name A	-	Date modified	▼ Type	▼ Size
Doguments	h837d4010x0	197a1c	1/17/2014 5:15	AM XML File	81
	h837d4010x0	97a1c	1/17/2014 5:15	AM XML File	81
Pictures	h837p4010x0	98a1c	1/17/2014 5:15	AM XML File	22
Music	h837p4010x0	98a1c	1/17/2014 5:15	AM XML File	22
Recently Changed	h8204010x06	1a1c0	1/17/2014 5:15	AM XML File	12
P Searches	h8204010x06	1a1c0	1/17/2014 5:15	AM XML File	12
Public	h8354010x09	1a1c0	1/17/2014 5:15	AM XML File	26
- Conc	h8354010x09	1a1c0	1/17/2014 5:15	AM XML File	26
	hia 1834c001	4010_01	1/17/2014 5:15	AM XML File	12
	hia 1834c001	4010	1/17/2014 5:15	AM XML File	12
Corganize 👻 📋 Viet	uter V Local Disk (C:) V	HIPAA_A	ccelerator - HIP.	AA_out + OB	Archive + 5010_xml
-avorite Links	h277u5010x2	14c01	1/23/2014 5:35 4	M XML File	9
	h837d5010x2	24a2c	1/23/2014 5:35 A	M XML File	21
Documents					
Documents Pictures	h837i5010x22	3a2c0	1/23/2014 5:35 A	M XML File	164
Documents Pictures Music	h837i5010x22	3a2c0	1/23/2014 5:35 A 1/23/2014 5:35 A	M XML File	164 164
Documents Pictures Music	h837i5010x22 h837i5010x22 h837i5010x22 h837p5010x2	3a2c0 3a2c0 22a1c	1/23/2014 5:35 A 1/23/2014 5:35 A 1/23/2014 5:35 A	M XML File M XML File M XML File	164 164 17
Documents Pictures Music Recently Changed	h837i50 10x22 h837i50 10x22 h837p50 10x22 h837p50 10x2	3a2c0 3a2c0 22a1c 22a1c	1/23/2014 5:35 / 1/23/2014 5:35 / 1/23/2014 5:35 / 1/23/2014 5:35 /	M XML File M XML File M XML File M XML File	164 164 17 17

Importing HIPAA User Samples to iWay Integration Tools as a Workspace

This section describes how to import HIPAA user samples to iWay Integration Tools (iIT) as a workspace.

Procedure: How to Import HIPAA User Samples to iWay Integration Tools as a Workspace

1. Start iWay Integration Tools (iIT).

2. Right-click anywhere inside the Integration Explorer tab and select *Import...* from the context menu, as shown in the following image.



The Import dialog opens, as shown in the following image.

1 militare	
elect Create new projects from an archive file or directory.	Ľ
Select an import source:	
type filter text	
Archive File Existing Projects into Workspace File System File System File System CVS File System File System File System File System File	

3. Expand the General folder, select Existing Projects into Workspace, and then click Next.

The Import Projects pane opens, as shown in the following image.

🛃 Import	
Import Projects Select a directory to search for existing Eclipse projects.	
Select root directory: Select archive file: Projects:	Browse Browse
	Select All Deselect All R <u>e</u> fresh
Copy projects into workspace Working sets Add project to working sets Working sets:	Sglect
Working sets:	Select

4. Click Select archive file and then click Browse.

The Select archive containing the projects to import pane opens, as shown in the following image.



5. Select the *Hipaa_usr_samples_ilT_workspace.zip* file and click *Open*.

You are returned to the Import Projects pane, as shown in the following image.

d Import		
Import Projects Select a directory to sear	ch for existing Eclipse projects.	
 Select root directory: Select archive file: Projects: 	E:\IIT_Builds\JAN 16 2014\iIT-7.0.0\Hipaa_usı	Browse
V Hipaa_usr_sample	es_proj (Hipaa_usr_samples_proj)	Select All
Copy projects into wo Working sets	rkspace ing sets	Sglect
?	< Back Next > Einish	Cancel

6. Click Finish.

The HIPAA user samples are loaded into your iIT workspace, as shown in the following image.



The Integration Explorer tab on the left pane displays a hierarchy of all the imported channel components (for example, Ebixes, listeners, outlets, preparsers, routes, process flows, and so on). The Console tab on the bottom provides a status as each channel component is imported.

Publishing iWay Integration Applications to the iWay Service Manager Registry

This section describes how to publish iWay Integration Applications (iIAs) to the iWay Service Manager (iSM) Registry.

Procedure: How to Publish iWay Integration Applications to the iWay Service Manager Registry

1. In the Integration Explorer tab, right-click *Hipaa_usr_samples_App*, select *Integration Tools* from the context menu, and then click *Publish to...*, as shown in the following image.

💋 Integrati 🖾 😇 iWay Ex	p 🛋 Library M	0				
		雪	~			
Hipaa_usr_samples_proj Adapters Applications Hipaa_usr_sample	s Ann	•	^			
Channels	New	5 I				
Emitters	Open Wat					
HipaaAckOul XmltoHipaaE XmltoHipaaE XmltoHipaaE Inlets HipaaToXML XmlToHipaa Listeners Kir HipaaToXML Kir AlipaaToXML Kir	Open With Copy Paste Duplicate Move Rename Import	-				
🖻 🗁 Outlets 🗷 🐎 HipaaAckOul –	Export	_				
Control Contro Control Control Control Control Control Control Control Control Co	Calidate Run As Debug As Team			Pror	perties	Ø Fr
🗷 🔁 XmltoHipaaE	Compare With	•		IT Mess	age Cop	cole
🕀 🗧 HipaaToXML_IB_	Replace With	 International 			age con	3010
🕀 🗧 XmlToHipaa_Q5_	Integration Tools	·	Publi	sh	Ctrl+I,	F
Ebixes Flows	Properties		Build	sh to	Ctrl+I,	Р
 説 HipaatoXML_pflow 説 XMLToHipaa_pFlow 一 20 Registers 	v_AckAgent w	6	Build Depl	oy	Ctrl+I,	D
			Crea	ice kepor	C	

The Publish Resource Wizard dialog opens, as shown in the following image.

A Publish	🛛	
Server Sel Please specif resource. Yo	ec tion y the server where you would like to publish this u can either type in the full SOAP url,	
Server URL:	http://192.168.128.145:9000	~
Password	••••	
?	Einish	Cancel

2. In the Server URL field, type the server IP number or computer name and then the port number (default port is 9000). For example:

http://111.111.111.000:9000

Type the iSM credentials (for example, user name: iway, password: iway).

3. Click Finish.

The Console tab on the bottom provides a status log that you can use for verification purposes, as shown in the following image.



Deploying iWay Integration Applications to iWay Service Manager

This section describes how to deploy iWay Integration Applications (iIAs) to iWay Service Manager (iSM).

Procedure: How to Deploy iWay Integration Applications to iWay Service Manager

1. Enter the following URL to access the iSM Administration Console:

```
http://[host]:[port]/ism
```

where:

host

Is the host machine where iSM is installed. The default value is *localhost*.

port

Is the port where iSM is listening. The default port is 9999.

2. After publishing the iWay Integration Application (Hipaa_usr_samples_App), you can find this iIA under the Management\Applications link in the iSM Administration Console, as shown in the following image.

iWay Service Ma Server Registry Do	nager ployments Tools		Management base	R
Application Management Deployments Applications	Applications Upload/Download/Delete app be deployed, started, stoppe	lications. iWay Integ d and deleted withou Actions	gration Application (IIA) is a ut affecting other IIAs.	n integration solution
Templates	Hinga usr samples Ann		amtev@AXTST145	01/23/14 18:07:29
Events	iWayXMLArchive		sh11355@sh11355-PC	02/28/11 10:17:47
Server Management	New Import	(copo)		

3. Click the *Deploy* icon in the application name under the Actions column, as shown in the following image.

iWay Service Ma Server Registry D	anager eployments Tools		Management base	Re
Application Management Deployments Applications	Applications Upload/Download/Delete app be deployed, started, stoppe Application	lications. iWay Integ d and deleted withou Actions	gration Application (IIA) is a it affecting other IIAs. Owner	n integration solution Version
Templates Events	Hipaa_usr_samples_App iWayXMLArchive		amtex@AXTST145 sh11355@sh11355-PC	01/23/14 18:07:29 02/28/11 10:17:47
Server Management	New Import	[00]107]		

The Deployments pane opens, as shown in the following image.

iWay Service M	anager reployments Tools	Management base
Application Management	Deployments - New Deplo Deploy an application	syment
Deployments	Deploy application Hipa	a_usr_samples_App (01/23/14 18:07:29)
Applications Templates Events	Deployment Name	Use an auto-generated name below or provide a custom n Hipaa_usr_samples_App
Server Management Servers Users	Deploy As Test Server	Selecting this option makes the deployment available as a
Server Roles Test Servers Remote Servers	Port	Port the console will listen on 10000
	Application Description	An automatic description is generated by default.
	< Back Deploy R	teset

- 4. Click Deploy.
- 5. From the Management drop-down list, select your deployed application (for example, *Hipaa_usr_samples_App [down]*), as shown in the following image.

iWay Service Manager Server Registry <u>Deployments</u> Tools			Managem	ent base Admin base	▼ Ø	About L		
Application Management	Deployments Monitor and manage deploye	ed application	s		Applications Hipaa_usr_samples_App Templates Test	p (down)		
Deployments	Deployment	Actions	State	Since	Application	Template	Source	
Templates	Hipaa_usr_samples_App	• * X	٢	06/13/14 06:26:44	Hipaa_usr_samples_App	raw		
Server Sources Properties General Properties	Monitoring Tools General Properties Listed below are the genera	al properties f	or the Hipe	aa_usr_samples_App a	Restart	Licenses	About L	
General Properties								
sore riopendes	General							
Settings	Application		Hipaa_usr_samples_App					
General Settings	Name / Home	n/a - C:/PROGRA~2/iway7/						
Console Settings	Version		7.0.0.1304					
Java Settings	Build Date PLATO Januar			PLATO January 22 2014 1837				
Register Settings	Settings							
Trace Settings	Name	Name Hinaa usr samples Ann C:/PROGRA~2/iway7/confin/Hinaa usr samples Ann						
Log Settings Path Settings	s Status Server is down							

- 6. Click Server in the top menu and then *Register Settings* in the left pane for the *Hipaa_usr_samples_App [down]* application.
- 7. Click Add to create all required registers (HIPAA_Installdir, HIPAA_Input, HIPAA_Output, and ValidateHIPAA) for the Hipaa_usr_samples_App [down] application.

For more information, see Setting Registers in the iWay Service Manager Administration Console on page 57.

8. In the State column, click the *Deployment State* icon to start the deployed Application.

iWay Service	Manager Deployments Tools			Management <mark>base</mark>	e 💌 Resta	 Icenses 	7 0.0-SNAP
Application Management	Deployments Monitor and manage deploye	ed application:	s				
Deployments	Deployment	Actions	State	Since	Application	Template	Source
Applications Templates	Hipaa_usr_samples_App	 Sections 		01/23/14 18:29:32	Hipaa_usr_samples_App	raw	Jource
Events	New Application is down (click to start)						

9. When the Message from webpage window appears, click OK to proceed.



- 10. Once the application has successfully started, place your input data into the input location that is configured for the application.
- 11. Select the *Hipaa_usr_samples_App [down]* application from the Management drop-down list.

Management	base 🔻	
	Admin base	ises About Logout
	Applications Hipaa_usr_samples_App [down]	
	Templates Test	

12. Click the *Monitoring* link and observe the page. The deployed application channels *HipaaToXML_IB_QS_AckRpt_Pflow_Channel* and *xmIToHipaa_QA_Channel* are displayed, as shown in the following image.

iWay Service Manager Nanagement				Hipaa_usr_samples_App M Admin base		art Licon	art Licenses About Logout		
Monitoring	Channels Monitor, start and stop application channels			Application Templates Test	isr_samples_App				
				Mess		sages	ages		
	Name	Type	State	Active	Completed	Successful	Failed	Description	
	HipaaToXML_IB_QS_AckRpt_Pflow_Channel	FILE	0					Hipaa to XML file listener	
	XmlToHipaa_QS_Channel	FILE	0					XML to Hipaa file listener	

The following image shows the inbound and outbound channels that are running in iSM. You can stop either channel and have only one channel running at a time as required.

iWay Service Manager Server Sources <u>Honitorina</u> Tools			Mai	agement	Hipaa_usr_sample	ts_App ▼ estart Licen	🧭 🧭 ses Aboi	7.0.0.1304 t Logout
Monitoring Channels	Channels Monitor, start and stop application channels							
			Type State		Messages			
	Name	Туре		Active	Completed	Successful	Failed	Description
	HipaaToXML_IB_QS_AckRpt_Pflow_Channel	FILE	0					Hipaa to XML file listener
4	XmlToHipaa_QS_Channel	FILE	0					XML to Hipaa file listener

Setting Registers in the iWay Service Manager Administration Console

This section describes how to set Registers in the iWay Service Manager (iSM) Administration Console.

Procedure: How to Set Registers in the iWay Service Manager Administration Console

1. From the iSM Administration Console, select the *Hipaa_usr_samples_App [down]* application from the Management drop-down list. Click *Server* in the top menu and then *Register Settings* in the left pane.

iWay Service	Manager Monitoring	Tools	Management	Hipaa_usr_samples_App [dov v 🕢 🕢 Restart Licenses	O 2 7.0.0
Properties General Properties Java Properties	Regist Specia becom restart	er Settings I registers are named variable e available to all components ed/redeployed. Listed below a cial Registers	is that reference values which are carried th of the system. Any changes to the register s re the register settings for the Hipaa_usr_sa	roughout the system. Once defined, t ettings do not take effect until the se mples_App configuration of this serve	hese variables rver is er.
General Settings		Name	Value	Description	Туре
Console Settings		iwayversion	unavailable	system defined (readonly)	string
Java Settings		iwayhome	unavailable	system defined (readonly)	string
Trace Settings		iwaydata	unavailable	system defined (readonly)	string
	8	Hipaa_XML.ValidationRepor	sreg(HIPAA_Output)/OB_Report		string
	Add	Delete			

2. Click Add.

3. Add *HIPAA_Installdir* and provide the appropriate values in the fields, as shown in the following image. Click *Finish*.

Cattings	Special Register Def	finition					
Settings	Name *	Enter the name of the special register to add.					
Console Settings		HIPAA_Installdir					
Register Settings	Type Select a type for the value of this special register.						
Trace Settings Log Settings Path Settings		string					
Data Settings	Value *	Enter a value for this special register. The value can be a constant or a call to the evaluation functions.					
Backup Settings		C:/HIPAA_Accelerator					
Providers		~					
Data Provider							
Services Provider	Description	Enter a description for this special register.					
Directory Provider							
Security Provider							
XML Namespace Map Provider		~					
Pooling Providers							
Authentication Realms	Finish						

4. Add *HIPAA_Input* and provide the appropriate values in the fields, as shown in the following image. Click *Finish*.

	Special Register Definition							
Settings	Name *	Enter the name of the special register to add.						
General Settings Console Settings		HIPAA_Input						
Java Settings	Ture	Colort a time for the value of this appendix register						
Register Settings	туре	Select a type for the value of this special register.						
Trace Settings		string						
Log Settings								
Path Settings		Taken a salah da bila ana dalam sister. Tha salah ana ka a salahak ana salih kuka salah disa da sibara						
Data Settings	Value *	Enter a value for this special register. I ne value can be a constant or a call to the evaluation functions.						
Backup Settings		sreg(HIPAA_Installdir)\HIPAA_in						
Providers								
Data Provider								
Services Provider	Description	Enter a description for this special register.						
Directory Provider								
Security Provider								
XML Namespace Map Provider		×						
Pooling Providers								
Authentication	Finish							

5. Add *HIPAA_Output* and provide the appropriate values in the fields, as shown in the following image. Click *Finish*.

0.11	Special Register De	efinition
General Settings	Name *	Enter the name of the special register to add.
Console Settings		HIPAA_Output
Java Settings	-	
Register Settings	Type	select a type for the value of this special register.
Trace Settings		string
Log Settings		
Path Settings		
Data Settings	Value *	Enter a value for this special register. The value can be a constant or a call to the evaluation functions.
Backup Settings		sreg(HIPAA_Installdir)\HIPAA_out
Providers		· · · · · · · · · · · · · · · · · · ·
Data Provider		
Services Provider	Description	Enter a description for this special register.
Directory Provider	10-00-00 • 1-00-00-0	
Security Provider		
XML Namespace Map Provider		<u></u>
Pooling Providers		
Authentication	Finish	

6. Add *ValidateHIPAA* and provide the appropriate values in the fields, as shown in the following image. Click *Finish*.

Collinso	Special Register D	efinition
Seconds	Name *	Enter the name of the special register to add.
Console Settings		ValidateHIPAA
Java Settings	Type	Select a type for the value of this special register.
Register Settings		
Trace Settings		string
Log Settings		
Path Settings	Value *	Enter a value for this appaid register. The value can be a constant or a call to the subvalian functions
Data Settings	value	Enter a value for this special register. The value can be a constant or a call to the evaluation functions.
Backup Settings		true
Providers		
Data Provider		
Services Provider	Description	Enter a description for this special register.
Directory Provider		
Security Provider		
XML Namespace Map Provider		M.
Pooling Providers		
Authentication	Finish	

Secure Shell Provider	HIPAA_Input	sreg(HIPAA_Installdir)/HIPAA_in	string
Schedule Provider	HIPAA_Installdir	C:/HIPAA_Accelerator	string
SNMP Provider	HIPAA_Output	sreg(HIPAA_Installdir)/HIPAA_out	string
Facilities	Hipaa.Ack	sreg(HIPAA_Input)/OB_Output	string
Activity Facility Correlation Facility	Hipaa Archive	sreg(HIPAA_Input)/IB_Archive	string
	Hipaa BadOutput	sreg(HIPAA_Input)/IB_Error	string
	Hipaa.GoodOutput	sreg(HIPAA_Input)/IB_Output	string
	Hipaa Input	sreg(HIPAA_Input)	string
	Hipaa Rpt	sreg(HIPAA_Input)/IB_Report	string
	Hipaa_XML.Archive	sreg(HIPAA_Output)/OB_Archive	string
	Hipaa_XML.Error	sreg(HIPAA_Output)/OB_Error	string
	Hipaa_XML.Input	sreg(HIPAA_Output)	string
	Hipaa_XML.Output	sreg(HIPAA_Output)/OB_Output	string
	Hipaa_XML.ValidationReport	sreg(HIPAA_Output)/OB_Report	string
	ValidateHIPAA	true	string
	und much		
	Add Delete		

The following image shows the summary of defined Registers.

Note: If any changes are made to Registers after an application has started, you must restart that application for these changes to be applied.

Stopping Inbound (HIPAA to XML) and Outbound (XML to HIPAA) Processing

This section describes how to stop inbound (HIPAA to XML) and outbound (XML to HIPAA) processing.

Procedure: How to Stop Inbound (HIPAA to XML) Processing

Click the State icon adjacent to the inbound application channel (HipaaToXml_IB_QA_AckRpt_Pflow_Channel) under Management\Monitoring and click *OK*, as shown in the following image.

Mar Mon	ager toring Tools	Ma	nageme	nt Hipaa_usr_	samples_App R	💌 🙆 🧭 estart Licen	0 🕜 7.0.0 ses Abo	-SNAPSHOT 1256 Hut Logout	
	Channels Monitor, start and stop application channels				Mes	sages			
	Name	Туре	State	Active	Completed	Successful	Failed	Description	_
	HipaaToXML_IB_QS_AckRpt_Pflow_Channel	FILE	0	Message from webpage Are you sure you want to stop the listener 'HipaaToXML_IB_QS_AckRpt_Pflow_Chann					v_Channel?
	XmlToHipaa_QS_Channel	FILE	0			ок (Cancel)	

The inbound application channel will be stopped, as shown in the following image.

Nanager Monitoring Tools	Ma	nagemen	Hipaa_usr_	_samples_App R	💌 🙆 🥝 lestart Licen	0 🕜 7.0.0 ses Abo	-SNAPSHOT.1256 ut Logout
Channels Monitor, start and stop application channels							
			Messages				
Name	Туре	State	Active	Completed	Successful	Failed	Description
HipaaToXML_IB_QS_AckRpt_Pflow_Channel	FILE	0					Hipaa to XML file listener
XmIToHipaa_QS_Channel	FILE	0					XML to Hipaa file listener

Procedure: How to Stop Outbound (XML to HIPAA) Processing

Click the State icon adjacent to the outbound application channel (XmIToHipaa_QS_Channel) under Management\Monitoring and click *OK*, as shown in the following image.

anager onitoring Tools	Ma	nagemer	nt Hipaa_usr_	_samples_App	💌 🕢 🧭	0 🕜 7.0.0	SNAPSHOT 1258	
Channels Monitor, start and stop application channels								
				Me	ssages			
Name	Туре	State	Active	Completed	Successful	Failed	Description	
HipaaToXML IB QS AckRot Pflow Channel	FILE						Hipaa to XMI_file	
			Message f	from webpage				
XmlToHipaa_QS_Channel	FILE	0	Are you sure you want to stop the listener %mlToHipaa_Q5_Chann					
					OK Can	cel		

The outbound application channel will be stopped, as shown in the following image.

Ma ™⊡	nager nitoring Tools	Ma	nagemen	t Hipaa_usr_	samples_App R	💌 🧟 🧭 estart Licen	7.0.0 ses Abo	-SNAPSHOT.1256 ut Legout
	Channels Monitor, start and stop application channels							
				Messages				
	Name	Туре	State	Active	Completed	Successful	Failed	Description
	HipaaToXML_IB_QS_AckRpt_Pflow_Channel	FILE	0					Hipaa to XML file listener
	XmlToHipaa_QS_Channel	FILE	0					XML to Hipaa file listener

Testing the Sample HIPAA Applications

This section describes how to test the sample inbound (HIPAA to XML) and outbound (XML to HIPAA) applications.

Procedure: How to Test the Sample Inbound (HIPAA to XML) Application

1. Copy the input test data to the following directory:

Hipaa_Accelerator\HIPAA_in

For example:

🕒 😳 🦾 🔹 Computer 🔹 OS (C	:) • HIPAA_Accelerator • HIPAA_in •			
Organize 👻 📄 Open New folder				
🚖 Favorites	Name *	Date modified	Туре	Size
E Desktop	3 IB_Archive	6/13/2014 12:19 PM	File folder	
Downloads	IB_Error	2/3/2014 11:51 PM	File folder	
20 Recent Places	JB_Output	6/13/2014 12:19 PM	File folder	
	3 IB_Report	6/13/2014 12:19 PM	File folder	
Libraries	3 IB_TransformGood	2/3/2014 11:51 PM	File folder	
Documents	B OB_Archive	2/3/2014 11:51 PM	File folder	
Pichares	B_Error	2/3/2014 11:51 PM	File folder	
Subversion	B_Output	6/13/2014 12:19 PM	File folder	
Videos	B OB_Report	2/3/2014 11:51 PM	File folder	
	B OB_TransformGood	2/3/2014 11:51 PM	File folder	
P Computer	h837d5010x224a2c01_001.data	1/17/2014 7:48 AM	DATA File	210
💒 OS (C:)	h837dp5010x225a2c01_001.data	1/17/2014 7:44 AM	DATA File	8 13
C on THIRU	h837/5010x223a2c01_001.data	1/17/2014 7:44 AM	DATA File	14 Ki

2. Observe the transformed XML output in the following directory:

Hipaa_Accelerator\HIPAA_in\IB_Output

For example:

IB_Output								
GO V Komputer + OS (C:) + HIPAA_Acc	elerator • HIPAA_in • IB_Output							
Organize 💌 Include in Ibrary 💌 Share with 💌 New folder								
> Favorites	Name	Date modified ~	Туре	Size				
E Desktop	h837dp5010x225a2c01_001_2014-06-13T16_19_30_1382001.xml	6/13/2014 12:19 PM	XML File	64 KB				
Downloads	h837/5010x223a2c01_001_2014-06-13T16_19_31_4902001.xml	6/13/2014 12:19 PM	XML File	106 KB				
Secent Places	h837d5010x224a2c01_001_2014-06-13T16_19_29_3322001.xml	6/13/2014 12:19 PM	XML File	14 KB				

3. Observe the Reports in the following directory:

Hipaa_Accelerator\HIPAA_in\IB_Report

For example:

IB_Report G →	_Accelerator + HIPAA_in + IB_Report							
Organize Include in library Share with New folder								
Favorites	Name	Date modified -	Туре	Size				
Desktop	h837dp5010x225a2c01_001_2014-06-13T16_19_30_1322001.xml	6/13/2014 12:19 PM	XML File	72 KB				
Downloads	h837/5010x223a2c01_001_2014-06-13T16_19_31_494Z001.xml	6/13/2014 12:19 PM	XML File	120 KB				
M Recent Places	h837d5010x224a2c01_001_2014-06-13T16_19_29_334Z001.xml	6/13/2014 12:19 PM	XML File	16 KB				

4. Observe the Acknowledgement in the following directory:

Hipaa_Accelerator\HIPAA_in\OB_Output

For example:



5. If any Error occurs in the input test data then observe Error data in the following directory: Hipaa_Accelerator\HIPAA_in\IB_Error

For example:

B_Error				
🌀 🔾 🖉 🔸 Computer 🔹 OS (C:) • HIPAA_Accelerator • HIPAA_in • IB_Error			
Organize 👻 Include in library 👻	Share with Vew folder			
😤 Favorites	Name ~	Date modified	Туре	Size
E Desktop				This folder is em

6. After processing the input data that you place for transformation, a copy of input data will get stored in the following directory:

Hipaa_Accelerator\HIPAA_in\IB_Archive

For example:

🕌 IB_Archive				
🔆 🔁 🗸 🗸 - Computer - OS (C:)	HIPAA_Accelerator HIPAA_in IB_Archive			
Organize 🝷 🍃 Open 🛛 Include in libr	ary Share with New folder			
🔆 Favorites	Name *	Date modified	Type	Size
E Desktop	🍓 4010_hipaa	5/30/2014 6:49 AM	File folder	
Downloads	🍌 5010_hipaa	5/30/2014 6:49 AM	File folder	
Recent Places	h837d5010x224a2c01_001.data	6/13/2014 12:19 PM	DATA File	2 KB
1 brarian	h837dp5010x225a2c01_001.data	6/13/2014 12:19 PM	DATA File	8 KB
Documents	h837i5010x223a2c01_001.data	6/13/2014 12:19 PM	DATA File	14 KB

Procedure: How to Test the Sample Outbound (XML to HIPAA) Application

1. Copy the input test data to the following directory:

Hipaa_Accelerator\HIPAA_out

For example:

Organize Include in library	Share with Vew folder			
🔆 Favorites	Name ~	Date modified	Туре	Size
E Desktop	IB_Archive	2/3/2014 11:51 PM	File folder	
Downloads	B_Error	2/3/2014 11:51 PM	File folder	
Recent Places	IB_Output	2/3/2014 11:51 PM	File folder	
1 braries	JB_Report	2/3/2014 11:51 PM	File folder	
Documents	IB_TransformGood	2/3/2014 11:51 PM	File folder	
J Music	OB_Archive	5/30/2014 7:03 AM	File folder	
Pictures	B_Error	2/3/2014 11:51 PM	File folder	
Subversion	B_Output	5/30/2014 7:03 AM	File folder	
Videos	B_Report	5/30/2014 7:03 AM	File folder	
Computer	OB_TransformGood	2/3/2014 11:51 PM	File folder	
A OS (C:)	h837d4010x097a1c01_01.xml	5/30/2014 7:03 AM	XML File	81 K
C on THIRU	h837d4010x097a1c01_ta1.xml	5/30/2014 7:03 AM	XML File	81 K
D on THIRU	h837p4010x098a1c01_01.xml	5/30/2014 7:03 AM	XML File	22 10

2. Observe the transformed XML output in the following directory:

Hipaa_Accelerator\HIPAA_out\OB_Output

For example:

B_Output			
Computer + OS (C:) + HIP	AA_Accelerator + HIPAA_out + OB_Output		
Organize 💌 Indude in library 💌 Share w	ith 🔻 New folder		
🚖 Favorites	Name	Date modified - Type	Size
Desktop	h837d4010x097a1c01_01_2014-05-3	0T11_03_11_059Z001.hipaa 5/30/2014 7:03 AM HIPAA F	ile 6 KB
Downloads	h837d4010x097a1c01_ta1_2014-05-	30T11_03_11_2882001.hipaa 5/30/2014 7:03 AM HIPAA F	ile 6 KB
🕍 Recent Places	h837p4010x098a1c01_01_2014-05-3	0T11_03_11_5792001.hipaa 5/30/2014 7:03 AM HIPAA F	He 2 KB

3. Observe the Reports in the following directory:

Hipaa_Accelerator\HIPAA_out\OB_Report

For example:

B_OB_Report Computer → OS (C:) → HIPAA_Acc	referator + HIPAA_out + OB_Report			
Organize Include in library Share with	New folder			
😤 Favorites	Name	Date modified ~	Туре	Size
E Desktop	validationh837d4010x097a1c01_01_2014-05-30T11_03_11_0882001.xml	5/30/2014 7:03 AM	XML File	7 KB
bownloads 2	validationh837d4010x097a1c01_ta1_2014-05-30T11_03_11_3522001.xml	5/30/2014 7:03 AM	XML File	7 KB
Recent Places	validationh837p4010x098a1c01_01_2014-05-30T11_03_11_5992001.xml	5/30/2014 7:03 AM	XML File	3 KB

4. If any Error occurs in the input test data then observe Error data in the following directory:

Hipaa_Accelerator\HIPAA_out\OB_Error

For example:

B_Error				
- Computer + OS (C:)	HIPAA_Accelerator HIPAA_out OB_Error			
Organize 👻 Include in library 👻 S	ihare with 👻 New folder			
★ Favorites	Name ^	Date modified	Туре	Size
Desktop				This folder is empty.

5. After processing the input data that you place for transformation, a copy of input data will get stored in the following directory:

Hipaa_Accelerator\HIPAA_out\OB_Archive

For example:

B_Archive				
🔄 🔾 🗸 • Computer • OS (C:) • H	IPAA_Accelerator • HIPAA_out • OB_Archive •			
Organize 👻 Include in library 👻 Share	with 🔻 New folder			
☆ Favorites	Name ~	Date modified	Туре	Size
E Desktop	3 4010_xml	5/30/2014 6:49 AM	File folder	
Downloads	🎉 5010_xml	5/30/2014 6:49 AM	File folder	
Recent Places	h837d4010x097a1c01_01.xml	5/30/2014 7:03 AM	XML File	81 KB
Tibraries	h837d4010x097a1c01_ta1.xml	5/30/2014 7:03 AM	XML File	81 KB
Documents	h837p4010x098a1c01_01.xml	5/30/2014 7:03 AM	XML File	22 KB



Inbound Processing: HIPAA to XML

The iWay Integration Solution for HIPAA includes iWay Service Manager. iWay Service Manager converts a document from HIPAA format to XML format, and validates it based on implementation guides published by HIPAA.

This chapter provides the information you need to understand and implement a basic inbound message flow.

- □ The **inbound processing overview** describes the iWay business components and the processing steps in the basic inbound message flow.
- □ The **sample configuration** contains detailed instructions for configuring the basic inbound message flow. This topic guides you through each step of the configuration procedure.

In this chapter:

- HIPAA Inbound Processing Overview
- Sample Configuration for Inbound Processing: HIPAA to XML

HIPAA Inbound Processing Overview

The inbound process converts a HIPAA formatted document to an XML document.

In a basic message flow, inbound processing consists of the following components and steps. For an illustration of the components available in the construction of a message flow, see *Using a Channel to Construct a HIPAA Message Flow* on page 23. You will define the components in the configuration instructions in *Sample Configuration for Inbound Processing: HIPAA to XML* on page 69.

Inlet

- **I** The **listener** picks up the incoming HIPAA document.
- □ The **preparser** obtains the message type and version from the HIPAA document, in order to select the appropriate transformation template name. The transformation template converts the original HIPAA document to an XML representation of that document.

The preparser ensures that the document is converted to a structurally correct HIPAA XML document. The transformation templates that are provided in the **Ebix** are used to transform the structure of the document.

Validation

❑ The iWay Integration Solution for HIPAA ensures the validity of the HIPAA format and content of the XML document. The HIPAA level 1-5 validation tests are performed. The integration solution uses the rules provided in the **Ebix** for each transaction to apply rules in compliance with the HIPAA implementation guides to ensure that the resulting XML is HIPAA compliant.

For example, here is a typical date segment in an inbound HIPAA document:

DTM*001*20080701

The value in DTM01 ("001") is validated against an allowed code list. The value in DTM02 ("20080701") is validated as a properly formatted date.

In addition, the following business rule is applied: DTM02 is required if DTM01 is present (if there is a qualifier, there must be data).

Route

After validation, you can apply any additional business logic to the document. You can use a single service or multiple services, passing the output of one service to the input of the next.

In our basic message flow, the copy service redirects the output document to the destination.

For details on available services, see the *iWay* Service Manager User's Guide.

□ The **acknowledgement service** creates a functional acknowledgement (997 or 999) for the inbound document. The acknowledgement indicates that the document was received and validated for structure.

Outlets

Outlets define how messages leave a channel at the end of a process. In our basic sample channel, two outlets are configured:

- One outputs the XML format of the document. In a real case scenario, it would output the result of your business logic.
- □ The other outputs the functional acknowledgement. A functional acknowledgement is typically returned to the sender of the document.

Sample Configuration for Inbound Processing: HIPAA to XML

This topic provides step-by-step instructions on how to configure a basic inbound message flow for the iWay Integration Solution for HIPAA. This message flow represents the movement and tasks that are performed during the conversion of a message from HIPAA format to XML format and an acknowledgement of the message.

The inbound configuration that is described in this topic represents the simplest possible route and is not equivalent to the inbound configuration that is described in *Working With HIPAA Inbound and Outbound Applications Using iWay Integration Tools (iIT)* on page 41. The inbound configuration in the quick start topic processes different HIPAA versions. The inbound configuration in this topic processes a single HIPAA version.

If you plan to modify the message flow that is described in this section and want more information on the supported iWay business components that you can use during the construction of a channel, see the *iWay Service Manager User's Guide*.

Accessing the iWay Service Manager Administration Console

To access the iWay Service Manager Administration Console, you must first ensure that the iWay Service Manager service is running.

For instructions on starting iWay Service Manager, see the iWay Service Manager User's Guide.

Procedure: How to Access the iWay Service Manager Administration Console on Windows

1. From the Windows desktop, select Start, All Programs, iWay 8.0 Service Manager, and Console.

or,

from a browser such as Microsoft Internet Explorer, enter the following URL,

http://host:port

where:

host

Is the host machine on which iWay Service Manager is installed. The default value is localhost.

port

Is the port number on which iWay Service Manager is listening. The default value is 9999.

The following image shows the URL with the default values.



2. When prompted, enter your user name and password, and click OK.

Note: The default user name and password is *iway*.

The iWay Service Manager Administration Console opens, as shown in the following image.

iWay Service Man	ager	Management base 🔹 🙆 🙆 🕐 7.0.7.3672
Server Registry Deploy	ments Tools	
Properties General Properties	General Properties Listed below are the general prope	rties for the base configuration of this server.
Java Properties	General	
Settings	Name / Home	INFORMA-Q8T67IU\$ - C:/PROGRA~2/iway7/
General Settings	Version	7.0.7.3672
Console Settings	Build Date	ASGARD March 3 2017 1943
Java Settings	Usage	Live
Register Settings	Configuration	
Trace Settings	Name	base - C:/PROGRA~2/iway7/config/base
Log Settings	Status	Server Uptime: 37 minutes
Path Settings	User Security Access	Read / Write
Data Settings	Environment	
Backup Settings	OS / Hardware	Windows 10 (service) / amd64, CPUs: 2
Providers	Java Info	25.111-b14 Oracle Corporation Java HotSpot(TM) 64-Bit Server VM
Data Provider	Java Memory	222.91 MB of 1799.50 MB (12.4%) used
Services Provider	Classpath	
Directory Provider		[1] C: \PROGRA~2 \Way/ \conng\base \ib
Security Provider	Language and Locale	
Provider	Locale / Timezone	en / America/New_York; time zone offset is -4 hours
Pooling Providers	Language	English 🔻 Save
Authentication Realms		The server has to be stopped, and started for the language change to take effect.
Data Quality Providers		
TCP Connection Provider		
Token Store Provider		
Schedule Provider		
Calendar Provider		
SNMP Provider		
Secure Shell Provider		

Adding an Ebix to the Registry

The iWay e-Business Information Exchange (Ebix) framework supplies several Ebix files for the iWay Integration Solution for HIPAA.

An Ebix file for HIPAA is named HIPAA_Version.ebx, where Version is the HIPAA version number. For example, the Ebix file for HIPAA version 4010A1 is named HIPAA_4010A1.ebx.

For details on the supported HIPAA versions and transaction sets, see *Ebix-Supported Transaction* Sets on page 147.

This topic describes how to add an Ebix to the Registry on Windows and UNIX.

Procedure: How to Add an Ebix to the Registry on Windows

1. To access the Registry, select the *Registry* option in the blue shaded area below the iWay Service Manager banner, as shown in the following image.

iWay Service Man	ager	Management bas	se 🗸	🖉 🥺 😨 7.0.0-CFR.1190
Server Registry Deploy	yments Tools			Licenses About Logout
Properties General Properties	General Properties Listed below are the general properties for t	he base configuration of this server.		
Java Properties	General			
Settings	Name / Home	ac11698 - C:/PROGRA~2/iway7/		
General Settings	Version	7.0.0-CFR.1190		
Console Settings	Build Date	PLATO December 20 2013 1658		
Java Settings	Configuration			
Register Settings	Name	base - C:/PROGRA~2/iway7/config/	/base	
Trace Settings	Status	Server Uptime: 1 hours, 49 minutes	;	
Log Settings	User Security Access	Read / Write		
Data Settings	Environment			
Backup Settings	OS / Hardware	Windows 7 (process) / x86		
	Java Info	23.7-b01 Oracle Corporation Ja	ava HotSpot(TM) Client VM	
Providers	Java Memory	36.28 MB of 247.50 MB (14.7%) use	ed	
Data Provider Socioso Provider	Classpath	[1] C:\PROGRA~2\iway7\config\base	\\ib*	\checkmark
Directory Provider		[[-]		
Security Provider	Language and Locale		effect to f because	
XML Namespace Map	Locale / Timezone	en / America/New_York; time zone	offset is -5 nours	
Provider	Language	English V Save	started for the longuage ob	ange te teke offest
Pooling Providers		The server has to be stopped, and s	statted for the language cha	ange to take ellect.
Authentication Realms				
Secure Shell Providers				
Schedule Provider				
SNMP Provider				
Facilities				
Activity Facility				
Correlation Facility				

2. Under Components in the left pane of the Registry, select *Ebix*.

Components
Adapters
Decryptors
Ebix
Emitters 🖑

The Ebix pane opens, as shown in the following image.

e n	bix Bus ness	ine ag	ss Information Exch e flows related to eB	ange (ebix) are collections of sta Susiness.	ndard metadata, templates, rules an	d schemas that are typically used by
ſ	Eb	ix-				
		F	ilter By Name Where	Name 文 Equals 文		
	E		Name	Download	References	Description
			No data matching th	he selection criteria was found.		
L	_					
	Ad	d	Delete R	ename Copy		

3. Click *Add* to add a new Ebix. The New Ebix pane opens.

HIPAAtoXML_Channel 🕼 regs ebix 🖝	Channel for HIPAA to XML inbound processing.
----------------------------------	--

- 4. Browse to the directory in which the Ebix is located and select the name of the file, for example, *HIPAA_5010X212_pipeline.ebx*.
- 5. Once you have selected the Ebix, click Next.

You are prompted for the name of the Ebix and an optional description.

New Ebix	
Name *	Name of the new ebix
	HIPAA_5010X212_pipeline
Description	Description for the new ebix
	Hipaa 5010 pipeline ebix transactions 276 - 277

- 6. Enter a name for the Ebix, for example, *HIPAA_5010X212_pipeline*, and an optional description.
- 7. Click Finish.

On the Ebix pane, you will see that the Ebix was successfully added. Later you will associate it with the channel for inbound processing.

Note: This procedure must be repeated for each HIPAA message set that will be processed by this channel. For example, if HIPAA 997 messages are packaged in the HIPAA_5010_pipeline.ebx file and if your channel will be processing 997 messages, then this Ebix file must be added to the Registry.
Procedure: How to Add an Ebix to the Registry on UNIX

Depending on your system configuration, there are two methods that you can use to add an Ebix to the Registry on UNIX.

- If you have a web browser on the UNIX machine, follow the instructions for Windows.
- ❑ Use FTP to download the Ebix from the *iWay7/etc/manager/packages* directory to your Windows machine and follow the instructions for Windows.

Adding Special Register Sets

In iWay Service Manager, a special register is a name-value pair that defines a variable that is carried throughout the system. Once defined, this variable is available to all components of the system. Within the HIPAA components, a best practice is to use special registers to define inputs and outputs. When packages containing channels are migrated between systems, the only changes required to deploy in the new location is to modify these special registers and build the channel. Channels may have many locations and this practice will minimize the effort required to migrate. For a complete list of system special registers that are provided, see the *iWay Service Manager Programmer's Guide*. For more information on defining a special register of your own, see the *iWay Service Manager User's Guide*.

The sample inbound channel uses a set of special registers defined as HIPAA. For example:

Registers / Hipaa

Register name/value sets to be used by various conduits.

	Name	Туре	¥alue	Description
	Ack	string 💌	C:\file_out\hipaa\ack	Output directory for 997 <
	Ack999	string 🔽	C:\file_out\hipaa\ack	Output directory for 999 🥁
	Archive	string 💌	C:\file_out	Archive of transformed X12 files
	BadOutput	string 💌	c:\file_out\hipaa\bad	XML where ack status is 🔨 not equal to A(accept)
	Error	string 💌	C:\file_out	
	GoodOutput	string 💌	c:\file_out\hipaa\good	XML where ack status equal to A (accept)
	Input	string 💌	C:\file_in\hipaa	X12 inbound flow scans this directory for EDI
Add		string 🔽		

Procedure: How to Add a Special Register Set to Your Channel

To add a special register set to your channel:

1. In the left console pane of the Registry menu, select Channels.

The Channels pane opens.

- In the row for your channel, click *Regs* for the channel you want to modify. The Assign register pane opens.
- 3. Select a register and click Finish.
- 4. Click *Back* to return to the Channels pane.

Defining an Inlet

An inlet defines how a message enters a channel. It typically contains a:

Listener. A listener is a component that picks up input on a channel from a configured end point.

- Decryptor. A decryptor is a component that applies a decryption algorithm to an incoming message and verifies the security of the message. The configuration example in this topic does not include a decryptor, which is an optional component.
- One or more preparsers. A preparser is a component that converts incoming messages into processable documents. Typically a preparser converts a document into XML format. Other preparsers may perform data decryption or reformatting.

Procedure: How to Add a Listener

- 1. From the Registry menu options on the left pane, select *Listeners* under Components.
- 2. On the Listeners pane on the right, click *Add* to add a new listener.
- 3. For the purpose of this example, we will show the configuration with a File listener. For details on supported protocols, see the *iWay Service Manager Protocol Guide*.

Select File from the Type drop-down list and click Next.

The configuration parameters pane opens.

Input Path *	Directory in which input messages are received. A specific file name or DOS-style pattern) can be used. Do not use file suffix.
	sreg(Hipaa.Input) Browse
Destination *	Directory into which output files are stored. Specific file name is optional. Use * in file name to be replaced by timestamp, # by sequential counter
	sreg(Hipaa.GoodOutput) Browse
Removal Destination	Full path file pattern asserting where input files will be moved. Use * in file name to be replaced by timestamp, # by sequential counter
	sreg(Hipaa.Archive) Browse
Suffix In	Limits input files to those with these extensions. Ex: XML, in Do not use `;; - mean no extension, ' means any
	*
Scan subdirectories	If true, all subdirectories will be scanned for files to process
	false
	Pick one
Do not unzip ZIP files	Pass ZIP files as a single file for processing (requires ACCEPT FLAT turned on)
	false
	Pick one
Suffix Out	Extension for output files (name is same as input file unless specified in destination parameter)

4. Supply configuration parameters for the new File listener as follows. An asterisk indicates that a parameter is required. For parameters not listed in the following table, accept the default value.

Parameter	Value
Input Path *	sreg(Hipaa.Input)
	This value is a special register that uses a defined directory in which input messages are received.
	Make sure that you have created this directory; otherwise, errors will occur during deployment.
Destination *	sreg(Hipaa.GoodOutput)
	This value is a special register that uses a defined directory in which output files are stored after transformation.
	Make sure that you have created this directory; otherwise, errors will occur during deployment.
Removal Destination	sreg(Hipaa.Archive)
	This value is a special register that uses a defined directory to which input messages are moved if they fail during transformation.
	Make sure that you have created this directory; otherwise, errors will occur during deployment. It is recommended to configure a removal destination when you are constructing a basic channel.
Suffix In	*
	Input files with any file extension are allowed.
Suffix Out	xml
	The extension for output files is .xml.

5. Click Next.

You are prompted for the name of the listener and an optional description.

Listeners Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.			
Select listener typ	e		
Name *	Name of the new listener HipaaToXML_Ebix		
Description	Description for the new listener Hipaa to XML file listener		
< Back Finis	h		

 On the Listeners pane, enter the name of the new listener, *HipaaToXML_Ebix*, and an optional description. Then click *Finish* to add the listener. In a later step, you will associate this listener with the inlet.

Procedure: How to Add a Preparser

- 1. From the Registry menu options, select *Preparsers* under Components.
- 2. On the Preparsers pane, click *Add* to add a new preparser. You are prompted for the type of preparser.

Preparsers A logical process that handles documents before they are parsed by the system. Usually used to convert from non-XML to xml.				
Select the type for the new Preparser object definition				
Туре *	Available Preparser types			
	com.ibi.preparsers.HIPAASplitPP			
< Back Next	>>			

3. Select *HipaaSplitterPreParser (com.ibi.preparsers.HIPAASplitPP)* from the Type drop-down list.

The HipaaSplitterPreParser parses a HIPAA input file with one or more ISAs and multiple transaction sets (STs), and creates *multiple* XML output files. One XML output file is produced for each transaction set. You can also use the HipaaSplitterPreParser if there is only one transaction set in an ISA.

4. Click Next.

The Preparsers configuration parameters pane opens.

The following table lists and describes the available configuration parameters for the preparser:

Parameter	Description
Template	Used to locate the template in the Ebix used in the transformation from HIPAA format to XML format.
Debug	If enabled, the transformation components are written to files in the local directory. This parameter is set to False by default.
Segment Terminator	The control character that marks the end of a specific variable-length segment.
	To view a list of segment terminator characters, see <i>Using HIPAA Separators and Terminators</i> on page 175.
Element Delimiter	The control character used to separate elements in a segment. It follows the segment identifier and each data element in a segment except the last.
	To view a list of element delimiter characters, see <i>Using HIPAA Separators and Terminators</i> on page 175.
Component Element Delimiter	The control character used to separate sub- elements/components in a composite element.
	To view a list of component element delimiter characters, see <i>Using HIPAA Separators and Terminators</i> on page 175.
Escape Character	The escape character is necessary if any of the HIPAA document separators is part of the actual value of an attribute.

Parameter	Description
Timestamp	Disabled by default, this option writes a timestamp to the log file. When enabled, the log file will display the start and end time of the file transformation and the input file name that is used. This feature is useful in development or debugging environments when processing batches of files. When the transaction log is not in use (for example, in a production mode) then this information is available in the Activity Log.
	Note: To use this feature, logging must be enabled in the <i>Log Settings</i> section of the iWay Service Manager Administration Console.
XML Transformer	Enabled by default, this parameter sets the HipaaSplitterPreParser to transform the individual documents that are split from the incoming message into XML format.
	Note: Use any of the following standalone EDI batch splitter preparsers if you do not require an XML transformation to be called:
	HIPAA Batch Splitter (com.ibi.preparsers.XDHIPAABatchSplitter)
	EDI Batch Splitter (com.ibi.preparsers.XDEDIBatchSplitter)
Insert Group Loop	Inserts a Group loop tag in the XML document. Group loop tags are displayed in activity logs and validation processing reports.
	Note: Ensure that this parameter is set to <i>false</i> . By default, this parameter is set to <i>true</i> .

Parameter	Description
Node 'delimiters'	If set to <i>true</i> , node delimiters are added to the generated XML document. For example:
	<pre><?xml version="1.0" encoding="ISO-8859-1" ?> <hipaa271> <delimiters> <_01_Element_Terminator>2A<!-- _01_Element_Terminator--> <_02_Segment_Terminator>0D0A<!-- _02_Segment_Terminator--> </delimiters> <isa></isa></hipaa271></pre>
	By default, this parameter is set to false.

- 5. In the template field, enter the following template mask, based on the HIPAA version you are processing:
 - □ For HIPAA version 4010, enter:

%_^_HIPAA_XML.xch

□ For HIPAA version 5010, enter:

HIPAA_%_^toXML.xch

The preparser obtains the message type and version information from the HIPAA input document. In the parameter, the character "%" represents the message type, and the character "^" represents the version.

For example, if the message type of the HIPAA input document is 835 and the version is 004010x091A1, then the constructed template name is:

835_004010X091A1_HIPAA_XML.xch

If the message type of the HIPAA input document is 276 and the version is 005010X212, then the constructed template name is:

HIPAA_276_005010X212toXML.xch

6. Click Next.

You are prompted for a name and optional description for the new preparser.

Preparsers A logical process that handles documents before they are parsed by the system. Usually used to convert from non-XML to xml.			
Provide a name and description for the new Preparser object definition			
Name *	Name of the new Preparser object definition HipaaSplitter		
Description	Description for the new Preparser object definition Hipaa Splitter Preparser		
< Back Finis			

- 7. Enter a name for the new preparser, for example, *HipaaSplitter*, and an optional description.
- 8. Click *Finish* to add the preparser. In the next procedure, you will associate this preparser with an inlet.

Procedure: How to Define an Inlet

Now that you have added a File listener and splitter preparser to the Registry, you are ready to add and define an inlet. You will associate the previously created listener and preparser with the inlet.

- 1. From the Registry menu options, select *Inlets* under Conduits.
- 2. On the Inlet Definitions pane, click *Add* to add an inlet.
- 3. On the New Inlet Definition pane, enter the name of the new inlet and an optional description, as shown in the following table. Then click *Finish* to add the inlet.

Parameter	Value
Name *	HipaaToXML_Ebix
Description	The file inlet contains the file listener and edi splitter preparser for Hipaa to XML processing

4. On the Construct Inlet pane, click *Add* to associate the listener and preparser with the inlet.

The next pane prompts you for the component type.

Inlets / HipaaToXML_Ebix Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.				
Sel	ect component type –			
	Component Types	Description		
۲	Listener	Listeners are protocol handlers, that receive input for a channel from a configured endpoint.		
0	Decryptor	Decrypts the document.		
0	Preparser	A logical process that handles documents before they are parsed by the system. Usually used to convert from non-XML to xml.		
<<	Back Next >>			

5. Select Listener and click Next.

The next pane prompts you to select a listener.

6. Select *HipaaToXML_Ebix*, which is the listener you added earlier, and click *Finish*.

The listener is associated with the inlet. Now you need to associate the preparser created earlier with the inlet.

7. On the Construct Inlet pane, click Add.

The next pane prompts you for the component type.

In In	lets lets	HipaaToXML_Ebix are conduits which represent to the commence time.	sent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.	
	Sen	ect component type		l
		Component Types	Description	
	0	Decryptor	Decrypts the document.	
	•	Preparser	A logical process that handles documents before they are parsed by the system. U sually used to convert from non-XML to xml.	
	<<	Back Next >>		

8. Select Preparser and click Next.

On the next pane, you are prompted to select a preparser.

Inlets / HipaaToXML_Ebix Inlets are conduits which represent the entry into a channel. Inlets contain a Listener, Decryptor, and Preparsers.					
Г	Select one or more preparser definitions				
	Filter By Name Where Name Equals				
		Name	Туре	Description	
		Name HipaaSplitter	Type com.ibi.preparsers.HIPAASplitPP	Description Hipaa Splitter Preparser	

9. Select HipaaSplitter, which is the preparser you added earlier, and click Finish.

You have now successfully completed definition of the inlet.

Defining a Route

For this sample channel configuration, you will define a route that will invoke the HIPAA to XML validation process flow. The outcome of the validation process flow will place valid transformed XML data in a defined output folder. Invalid transformed data will be routed to an errors folder. A HIPAA functional acknowledgement and a validation report will be sent to their designated output folder defined in the sample channel.

iWay Integration Tools (iIT) Designer (previously known as iWay Designer) is a GUI tool that is delivered as a plugin with iIT. For more information, see the *iWay Integration Tools Designer User's Guide*.

This section describes how to create a validation process flow using iIT Designer and bind it to a sample inbound channel as a route.

Procedure: How to Create a New Project and Start the Process Flow

To create a new project and start the process flow using iIT Designer:

- 1. Open iIT.
- 2. Right-click on the Integration Explorer tab, select *New*, and then click *Integration Project* from the context menu.
- 3. In the Name field, provide a valid integration name (for example, *Test*), and then click *Finish*, as shown in the following image.



4. Right-click the *Flows* folder, select *New*, and then click *Process Flow* from the context menu, as shown in the following image.



🔬 New Process Flow Wizard		
General Properties 8 The name field is required.		1
Project Folder	/Test/Flows	Browse
Name	1	
Description		
Target Server Version	7.0.5-SNAPSHOT	-
	Install additional Target Server Version	
	Create in current folder	
	Enable taps	
•	Finish	Cancel

The New Process Flow Configuration Wizard opens, as shown in the following image.

- In the Name field, type *HIPAAtoXML_pflow_AckRpt* as the process flow name.
 In the Description field, type a brief description (optional).
- 6. Click Finish.

The new HIPAAtoXML_pflow_AckRpt node appears under the Flows folder, and the workspace displays a Start and End object with a relation established in between.



You are ready to build the HIPAAtoXML_pflow_AckRpt validation process flow by configuring objects to it and specifying their relationships.

Procedure: How to Configure Objects for the Process Flow

To configure objects for the process flow using iIT Designer:

1. Drag and drop the Service object from the toolbar to the workspace.

The New Service Object dialog box opens.

2. In the Name field, type *ValidationReport*, and a brief description (optional) in the Description field and click *Next*.

The Service Type dialog box opens.

- 3. Select *Class Name* and enter *com.ibi.agents.XDHIPAAValidationReportAgent* and click *Next*. The Properties dialog box opens.
- 4. Click Finish.

The new Service object (ValidationReport) appears in the workspace.

5. Select the *Start* object, right-click the *ValidationReport* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

6. From the Event drop-down list, select *OnCompletion* and click *OK*.

This option indicates that there are no conditions that affect the path, and that the path between the two objects will always be followed.

A line appears between the objects to indicate that a relationship has been established.



7. Drag and drop the File object from the toolbar to the workspace.

The New File Object dialog box opens.

8. In the Name field, type *Write to Rpt Dir*, and a brief description (optional) in the Description field and click *Next*.

The File Type dialog box opens.

9. From the Type drop-down list, select *File Write* and click *Next*.

The Properties dialog box opens.

New File Object 'Write to Rpt Dir' of Type 'File Write'				
Properties Enter the properties for the File object you wish to define.				
Name	Value	Description I		
Source of		Source of data to write. If omit		
* Target Dir	sreg(HIPAA.ValidRpt)	The target output directory		
* File Pattern	sreg(basename)_rpt xml 💉	The output file name, which c		
Avoid Pre	true	Should any preemitter be avoi		
Return	status	'status': status document will b		
Base64 D	false	If set, the value is assumed to 🥁		
< <u>B</u> ack Finish Cancel Help				

- 10. For the Target Directory parameter, enter a location where error data will be written, for example, *sreg*(*HIPAA*.*ValidRpt*).
- 11. For the File Pattern parameter, enter sreg(basename)_rpt.xml.

12. For the Return parameter, select *status* from the drop-down list and click *Finish*.

The new File object (Write To Rpt Dir) appears in the workspace.

13. Select the *ValidationReport* object, right-click the *Write To Rpt Dir* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

14. From the Event drop-down list, select OnCompletion and click OK.

A line appears between the objects to indicate that a relationship has been established.



15. Drag and drop the End object from the toolbar to the workspace.

The End Name and Description dialog box opens.

16. In the Name field, type *End_Rpt*, and a brief description (optional) in the Description field and click *Next*.

The End Name Schema dialog box opens.

17. Since no schemas are used in this processing path (that is, the process flow will not be exposed as a web service), from the Schema drop-down list, select *None* and click *Next*.

The Properties dialog box opens.

18. Click *Finish* to accept the default values and close the dialog box.

The new End_Rpt object appears in the workspace.

19. Select the *Write To Rpt Dir* object, right-click the *End_Rpt* object, and select *Relation* from the drop-down list.

The Line Configuration dialog box opens.



A line appears between the objects to indicate that a relationship has been established.

21. Drag and drop the Service object from the toolbar to the workspace.

The New Service Object dialog box opens.

22. In the Name field, type *HIPAAAckAgent*, and a brief description (optional) in the Description field and click *Next*.

The Service Type dialog box opens.

23. Select Class Name and enter com.ibi.agents.XDHIPAAAckAgent and click Next.

The Properties dialog box opens. The configuration parameters for HIPAAAckAgent are displayed. The following table lists and describes the configuration parameters.

Parameter	Description
Protocol	Protocol on which to make acknowledgment copies. Select one of the following options from the drop-down list: NONE FILE
Location	Location for acknowledgment copies.
End Tag	The surrounding XML tag.
Preemitter	Determines whether the preemitter should be run on acknowledgment output.
Error	Determines whether to send an error.

Parameter	Description
ISA Control Number	Element location of ISA control number. Select one of the following locations from the drop- down list:
	Input Document
	□ _SReg(edi.ICN)
GS Control Number	Element location of GS control number. Select one of the following locations from the drop- down list:
	Input Document
	SReg(edi.GCN)
ST Control Number	Element location of ST control number. Select one of the following locations from the drop-down list:
	Input Document
	SReg(edi.MCN)
Stream Acknowledgment	Determines the level of acknowledgment information to return. Select one of the following acknowledgment levels from the drop-down list:
	Group. Returns acknowledgment information at the Group level.
	Interchange. Returns acknowledgment information at the Interchange level.
	□ Transaction. Returns acknowledgment information at the Transaction level.

Parameter	Description
Acknowledgment Type	The functional acknowledgment message type to be returned (X12 997/HIPAA 999) after the incoming HIPAA document is validated. Select one of the following message types from the drop-down list: 997
	999
	Note: If the input document is HIPAA version 4010 version, then only a 997 functional acknowledgment message type is generated.

- 24. Configure the available parameters for HIPAAAckAgent according to your requirements.
- 25. Click Finish.

The new Service object (HIPAAAckAgent) appears in the workspace.

26. Select the *Start* object, right-click the *HIPAAAckAgent* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

27. From the Event drop-down list, select OnCompletion and click OK.

A line appears between the objects to indicate that a relationship has been established.



28. Drag and drop the Set object from the toolbar to the workspace.

The New Set Object dialog box opens.

29. In the Name field, type SetAckStatus, and a brief description (optional) in the Description field and click Next.

The Define Events dialog box opens.

Define Events Add one or more events which can be used to meet conditions defined in one or more Synch objects.				
				×
Name	Value	Usage	Description	
Event1	AckDone 🗸	user		
Click here to add				
<				>
< <u>B</u> ack Finish Cancel Help				

- 30. In the first row of the Name column, type *Event1*.
- 31. In the first row of the Name column, select *AckDone* from the drop-down list and click *Finish*.

The new Set object appears in the workspace.

32. Select the *HIPAAAckAgent* object, right-click the *SetAckStatus* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.



A line appears between the objects to indicate that a relationship has been established.

34. Drag and drop the End object from the toolbar to the workspace.

The End Name and Description dialog box opens.

35. In the Name field, type *End*, and a brief description (optional) in the Description field and click *Next*.

The End Name Schema dialog box opens.

36. Since no schemas are used in this processing path (that is, the process flow will not be exposed as a web service), from the Schema drop-down list, select *None* and click *Next*.

The Properties dialog box opens.

37. Click Finish to accept the default values and close the dialog box.

The new End_Success object appears in the workspace.

38. Select the SetAckStatus object, right-click the End object, and select Relation from the drop-down list.

The Line Configuration dialog box opens.



A line appears between the objects to indicate that a relationship has been established.

40. Drag and drop the Sync object from the toolbar to the workspace.

The New Sync Object dialog box opens.

41. In the Name field, type *SynchAckStatus*, and a brief description (optional) in the Description field and click *Next*.

The Sync Expression Builder dialog box opens.

42. Click the first row of the Name column.

Default values are entered in this row.

43. In the first row of the Name column, select *AckDone* from the drop-down list and click *Next*.

The Timeout dialog box opens.

44. Click Finish.

The new SynchAckStatus object appears in the workspace.

45. Select the *Start* object, right-click the *SynchAckStatus* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

This option indicates that there are no conditions that affect the path, and that the path between the two objects will always be followed.



A line appears between the objects to indicate that a relationship has been established.

47. Drag and drop the Decision Switch object from the toolbar to the workspace.

The New Decision Switch Object dialog box opens.

48. In the Name field, type *Check Ack Status*, and a brief description (optional) in the Description field and click *Next*.

The Switch Expression dialog box opens.

49. Type the following value in the field:

sreg(hipaa.ackstatus)

50. Click Next.

The Switch Cases dialog box opens.

Switch Cases Create a list of expected case values that you will wish to create custom events for.		
	XAV	
Case	Description	
null	Expression returns with no nodes	
empty	Expression returns with node(s) with no value	
A	HIPAA Good	
R	HIPAA Reject	
E	HIPAA Error	
P	HIPAA Partial	
Click here to add		
< <u>B</u> a	ack Finish Cancel Help	

51. In the Case Column, enter the following lines in separate rows:

Case	Description
A	HIPAA Good
R	HIPAA Reject
E	HIPAA Error
Р	HIPAA Partial

52. Click Finish.

The new Decision Switch object appears in the workspace.

53. Select the SynchAckStatus object, right-click the Check Ack Status object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

54. From the Event drop-down list, select OnCompletion and click OK.

This option indicates that there are no conditions that affect the path, and that the path between the two objects will always be followed.

A line appears between the objects to indicate that a relationship has been established.



55. Drag and drop the File object from the toolbar to the workspace.

The New File Object dialog box opens.

56. In the Name field, type Good File, and a brief description (optional) in the Description field and click Next.

The File Type dialog box opens.

57. Select File Write from the drop-down list and click Next.

The Properties dialog box opens.

58. For the Target Directory parameter, enter the following location where valid data will be written:

sreg(HIPAA.GoodOutput)

59. For the File Pattern parameter, enter the following:

sreg(basename)_*.xml

- 60. For the Respect Transactionality parameter, select *true* from the drop-down list.
- 61. Accept the default values for the remaining parameters and click Finish.

The new File object (Good File) appears in the workspace.

62. Select the *Check Ack Status* object, right-click the *Good File* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

- 63. From the Event drop-down list, select OnCustom.
- 64. In the Case of section, select case A (HIPAA Good).
- 65. Click OK.

A line appears between the objects to indicate that a relationship has been established.



66. Drag and drop the File object from the toolbar to the workspace.

The New File Object dialog box opens.

- 67. In the Name field, type Bad File, and a brief description (optional) in the Description field.
- 68. Click Next.

The File Type dialog box opens.

69. Select File Write from the drop-down list and click Next.

The Properties dialog box opens.

70. For the Target Directory parameter, enter the following location where valid data will be written:

sreg(HIPAA.BadOutput)

71. For the File Pattern parameter, enter the following:

sreg(basename)_*.xml

- 72. For the Respect Transactionality parameter, select true from the drop-down list.
- 73. Accept the default values for the remaining parameters.
- 74. Click Finish.

The new File object (Bad File) appears in the workspace.

75. Select the *Check Ack Status* object, right-click the *Bad File* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

- 76. From the Event drop-down list, select OnCustom.
- 77. In the Case of section, select cases *R* (*HIPAA Reject*), *E* (*HIPAA Error*), and *P* (*HIPAA Partial*).
- 78. Click OK.



A line appears between the objects to indicate that a relationship has been established.

79. Select the Good File object, right-click the End object, and select Relation from the dropdown list.

The Line Configuration dialog box opens.

- 80. From the Event drop-down list, select OnCompletion and click OK.
- 81. Select the *Bad File* object, right-click the *End* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.



A line appears between the objects to indicate that a relationship has been established.

The process flow is now complete.

83. To save the process flow, right-click the *HIPAAtoXML_pflow_AckRpt* node in the left pane and select Save from the context menu.

Now you need to validate the process flow and publish it to the Registry of the iWay Service Manager Administration Console for use in the route of a channel for outbound processing.

Validating a process flow ensures that its structure is correct. Publishing a process flow makes it available in the Registry for use in a channel configuration. For instructions on validating and publishing the process flow, see the *iWay Integration Tools Designer User's Guide*.

84. Close iIT Designer.

Your next step is to add a new route to the Registry using the iWay Service Manager Administration Console and associate the process flow with it.

Procedure: How to Define a Route and Associate the Process Flow With the Route

To define a route and associate the process flow with It:

- 1. From the Registry menu options in the iWay Service Manager Administration Console, click *Routes*.
- 2. On the Route Definitions pane, click *Add* to add a route.
- 3. On the New Route Definition pane, enter a name for the route and an optional description, as shown in the following table.

Parameter	Value
Name *	HIPAAtoXML_Route
Description	This route will invoke the HIPAA to XML validation process. The outcome of this process will place valid HIPAA transformed data in your valid inbound folder. Invalid HIPAA transformed data will be routed to its appropriate folder. A validation report will also be generated and sent to its appropriate folder.

- 4. Click Finish.
- 5. On the Construct Route pane, click Add.

You are prompted for the type of component to associate with the route.

- 6. Select Process and click Next.
- 7. The next pane prompts you to select a process. Select the process flow you created earlier with iIT Designer, *HIPAAtoXML_pflow_AckRpt*, and click *Finish*.

The route, with its associated process flow, has been successfully defined.

Defining the Outlets

An outlet defines how a message leaves a channel. An emitter is a transport protocol that sends a document to its recipient. In the sample configuration, we will use a File emitter. For details on supported protocols, see the *iWay Service Manager Protocol Guide*.

For the channel in this example, you will add three emitters to the Registry. Then you will define three outlets, associating one emitter with each outlet.

When you associate the outlets with the channel in later steps, you will apply a condition to each one to dynamically direct the flow of the output document based on its content.

In the example, you will add:

- ❑ An emitter for the XML output data. When you add the XML outlet to the channel, you will set the condition _isXML(). This condition tests the output data for XML format. If it is in XML format, it is routed to the specified destination.
- ❑ An emitter for the acknowledgement data. In the example, the data for the functional acknowledgement (transaction 997) is in EDI flat file (non-XML) format. When you add the acknowledgement outlet to the channel, you will set the condition _isFLAT(). This condition tests the output data for flat file (non-XML) format. If the data is in flat file (non-XML) format, it is routed to the specified destination.
- □ An emitter for data that failed rules validation.

Procedure: How to Add an Emitter for Valid XML Output

- 1. From the Registry menu options, select *Emitters*.
- 2. On the Emitters pane, click *Add* to add an emitter. The next pane prompts you for the emitter type.
- 3. For this example, select *File* from the drop-down list and click *Next*.

The configuration parameters pane opens.

Configuration parameters for new emitter of type File			
Destination *	path to file, * replaced with timestamp		
	<pre>sreg(Hipaa.GoodOutput)/SREG(basename)_*.xml</pre>		
Create Directory	ctory Create directory if it doesn't exist		
	false		
	Pick one	~	

4. Supply configuration parameters for the new File emitter as follows, then click Next.

Parameter	Value
Destination *	sreg(Hipaa.GoodOutput)/SREG(basename)_*.xml
	This value is the directory where the valid XML output is placed.
	sreg(Hipaa.GoodOutput) is a special register value that uses a defined directory in which valid output files are stored after transformation.
	Make sure that you have created this directory; otherwise, errors will occur during deployment.
	SREG(basename) is a special register reserved by iWay Service Manager, which uses a file name without an extension. For more information on how to configure a special register (SREG), see the <i>iWay</i> <i>Service Manager User's Guide</i> .
	On output, an asterisk (*) in the destination file name is replaced by a date and time stamp.
	Since you are using the HipaaSplitterPreParser, include an asterisk in the file name to create each XML output file with a unique name.
Create Directory	false

5. On the Emitters pane, enter the name of the new emitter and an optional description, as shown in the following table. Then click *Finish* to add the emitter.

Parameter	Value
Name *	HipaaFileOut
Description	Hipaa file output

Procedure: How to Add an Emitter for Acknowledgement Output

- 1. On the Emitters pane, click *Add* to add another emitter. The next pane prompts you for the emitter type.
- 2. For this example, select *File* from the drop-down list and click *Next*.

The configuration parameters pane opens.

Configuration parameters for new emitter of type File		
Destination *	path to file, * replaced with timestamp	
	sreg(Hipaa.Ack)/SREG(basename)*.txt	
Create Directory	Create directory if it doesn't exist	
	true	
	Pick one	*

3. Supply configuration parameters for the second File emitter as follows, then click Next.

Parameter	Value
Destination *	sreg(Hipaa.Ack)/SREG(basename)*.txt
	This value is the directory where the acknowledgement output is placed. You can use an extension other than .txt, for example, .edi or .data.
	sreg(Hipaa.Ack) is a special register value that uses a defined directory in which output files are stored after transformation.
	Make sure that you have created this directory; otherwise, errors will occur during deployment.
	On output, an asterisk (*) in the destination file name is replaced by a date and time stamp. For details on the special register (SREG) used in the preceding file name, see the <i>iWay Service Manager User</i> 's <i>Guide</i> .
Create Directory	false

4. On the Emitters pane, enter the name of the new emitter and an optional description, as shown in the following table. Then click *Finish* to add the emitter.

Parameter	Value
Name *	HipaaAckOut
Description	Hipaa Acknowledgement Out

Procedure: How to Add an Emitter for Invalid XML Output

- 1. From the Registry menu options, select *Emitters*.
- 2. On the Emitters pane, click *Add* to add an emitter. The next pane prompts you for the emitter type.
- 3. For this example, select *File* from the drop-down list and click *Next*.

The configuration parameters pane opens.

Configuration parameters for new emitter of type File	
Destination *	path to file, * replaced with timestamp
	<pre>sreg(Hipaa.BadOutput)/SREG(basename)_*.xml</pre>
Create Directory	Create directory if it doesn't exist
	false
	Pick one

4. Supply configuration parameters for the new File emitter as follows, then click Next.

Parameter	Value
Destination *	sreg(Hipaa.BadOutput)/SREG(basename)_*.xml
	This value is the directory where the invalid XML output is placed.
	sreg(Hipaa.BadOutput) is a special register value that uses a defined directory in which invalid output files are stored after transformation.
	Make sure that you have created this directory; otherwise, errors will occur during deployment.
	SREG(basename) is a special register reserved by iWay Service Manager, which uses a file name without an extension. For more information on how to configure a special register (SREG), see the <i>iWay</i> <i>Service Manager User's Guide</i> .
	On output, an asterisk (*) in the destination file name is replaced by a date and time stamp.
	Since you are using the HipaaSplitterPreParser, include an asterisk in the file name to create each XML output file with a unique name.
Create Directory	false

5. On the Emitters pane, enter the name of the new emitter and an optional description, as shown in the following table. Then click *Finish* to add the emitter.

Parameter	Value
Name *	HipaaBadFileOut
Description	Hipaa Bad Output File

Procedure: How to Define an Outlet for Acknowledgement Output

Now that you have added the required emitters to the Registry, you are ready to add and define the outlets. You will associate the previously created emitters with the outlets (one emitter for each outlet).

- 1. From the Registry menu options, select Outlets.
- 2. On the Outlet Definitions pane, click *Add* to add an outlet.
- 3. On the New Outlet Definition pane, enter the name of the new outlet and an optional description, as shown in the following table. Then click *Finish* to add the outlet.

Parameter	Value
Name *	HipaaAckOut
Description	Outlet containing Hipaa Acknowledgement to be sent to your trading partner

4. On the Construct Outlet pane, click *Add* to associate the acknowledgement emitter with the acknowledgement outlet.

The next pane prompts you for the component type.

Outlets / HipaaAckOut

Outlets are conduits which contain Preemitters, Encryptors, and an Emitter

	Component Types	Description
۲	Emitter	Emitters are protocol handlers, that drive the output of a channel to a configur ed endpoint.
0	Preemitter	A logical process that handles documents immediately prior to transmission. Usually this converts from XML to non-xml.
0	Encryptor	Encrypts the document.

5. Select Emitter and click Next.

The next pane prompts you to select an emitter.

6. Select *HipaaAckOut*, which is the acknowledgement emitter you added earlier, and click *Finish*.

The acknowledgement emitter is associated with the acknowledgement outlet.
Procedure: How to Define an Outlet for Invalid XML Output

- 1. From the Registry menu options, select *Outlets*.
- 2. On the Outlet Definitions pane, click *Add* to add an outlet.
- 3. On the New Outlet Definition pane, enter the name of the new outlet and an optional description, as shown in the following table. Then click *Finish* to add the outlet.

Parameter	Value
Name *	HipaaBadFileOut
Description	Outlet containing converted Hipaa data that failed rules validation

4. On the Construct Outlet pane, click *Add* to associate the invalid XML output emitter with the invalid XML output outlet.

The next pane prompts you for the component type.

5. Select Emitter and click Next.

The next pane prompts you to select an emitter.

6. Select *HipaaBadFileOut*, which is the XML emitter you added earlier, and click *Finish*.

The invalid XML output emitter is associated with the invalid XML output outlet.

Procedure: How to Define an Outlet for Valid XML Output

- 1. From the Registry menu options, select *Outlets*.
- 2. On the Outlet Definitions pane, click *Add* to add an outlet.
- 3. On the New Outlet Definition pane, enter the name of the new outlet and an optional description, as shown in the following table. Then click *Finish* to add the outlet.

Parameter	Value
Name *	HipaaFileOut
Description	Outlet containing converted Hipaa data into XML format

4. On the Construct Outlet pane, click *Add* to associate the valid XML output emitter with the valid XML output outlet.

The next pane prompts you for the component type.

5. Select Emitter and click Next.

The next pane prompts you to select an emitter.

6. Select *HipaaFileOut*, which is the XML emitter you added earlier, and click *Finish*.

The valid XML output emitter is associated with the valid XML output outlet.

Defining a Channel

Now that you have defined the inlet, route, and outlets for the channel, you are ready to add the channel to the Registry and associate the conduits with it.

Procedure: How to Define a Channel

- 1. From the Registry menu options, select *Channels* under Conduits.
- 2. On the Channel Definitions pane, click *Add* to add a channel.
- 3. On the New Channel Definition pane, enter the name of the new channel and an optional description, as shown in the following table. Then click *Finish* to add the channel.

Parameter	Value
Name *	HipaaToXML_Ebix
Description	Processing channel - Hipaa inbound data to XML. Channel uses SREG (Special Registers) to define destination paths. Ebix files should be attached to this channel before deployment.

4. On the Construct Channel pane, click *Add* to associate the inlet, route, and outlets defined previously with the channel.

You are prompted to associate components with the channel.

- 5. Select *Inlet* and click *Next*. The next pane prompts you to select an inlet.
- 6. Select *HipaaToXML_Ebix*, which is the inlet you defined earlier, and click *Finish*.

The inlet is added to the channel. Now you need to associate the route defined earlier with the channel.

7. On the Construct Channel pane, click Add.

The next pane prompts you for the component type.

8. Select *Route* and click *Next*.

On the next pane, you are prompted to select a route.

- 9. Select *HipaaToXML_Ebix*, which is the route created earlier, and click *Finish*.
- 10. On the Construct Channel pane, click the *minus sign* (-) under Conditions next to the name of the route to set it as the default.

Туре	Conditions	Move	D
Inlet			
Route	نه مالی		
Add	Dele <mark>Click to</mark>	set defau	IF.

- 11. On the Construct Channel pane, click *Add* to add the outlets.
- 12. On the next pane, select Outlet and click Next.
- 13. Select the three outlets defined earlier HipaaAckOut, HipaaBadFileOut, and HipaaFileOut.
- 14. Click Finish.
- 15. To set a condition for the outlets, on the Construct Channel pane, click the *plus sign* (+) under Conditions for the specific outlet.

Туре	Conditions
Inlet	
Route	46
Outlet	

The Set Condition pane opens.

In the Condition input field, enter the appropriate conditional expression, and click Update.
 The following table lists the expression that must be entered for each outlet.

Outlet	Expression
HipaaAckOut	_isflat()
HipaaBadFileOut	_isXML() AND COND(SREG(edi.ackstatus),NE,'A')

Outlet	Expression
HipaaFileOut	_isXML() AND COND(SREG(edi.ackstatus),EQ,'A')

For details on supported conditions, see the topic on using functions in the *iWay* Service *Manager* User's Guide.

Procedure: How to Build the Channel

- 1. From the Registry menu options on the left pane, select *Channels* under Conduits.
- 2. On the Channel Definitions pane, select the channel defined previously, *HipaaToXML_Ebix*, and click *Build*.

The results of the build are displayed on the right pane.

Channels

Channels are the pipes through which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

Build result for chan	nel
Message level	Message
Info	Start
Info	Validating Channel
Info	Channel is valid
Info	Validating Inlet
Info	Inlet is valid
Info	Validating Routes
Info	Routes are valid
Info	Validating Outlets
Info	Outlets are valid
Info	Build Successful
Info	End
Info	Channel archive C:\PROGRA~1\iway7\etc\repository\manager\car\HipaaToXML_Ebix\ HipaaToXML_Ebix.2\HipaaToXML_Ebix.car has been created/updated

<< Back

3. Review the results of your build and then click *Back*.

If an error or errors are displayed in the Message column, take the appropriate action as instructed.

Procedure: How to Deploy the Channel

Deployment is the mechanism by which a channel moves from being stored in the Registry to becoming active in iWay Service Manager. For more information on deployment, see the *iWay* Service Manager User's Guide.

- 1. Select the *Deployments* option in the blue shaded area below the iWay Service Manager banner.
- 2. On the Channel Management pane, click Deploy.

Ava This cha pag	ilable Channels s is a list of channels r nnel that has already t e.	eady for deployment into the seen deployed in to the sele	selected Managed Serve cted Managed Server. To	er. Select th Undeploy c	e channels and click deploy. You can not deploy a r Redeploy a channel, go back to the previous
	Channel Name	Build Date	Built On	Version	Description
۲	HipaaToXML_Ebix	May 12 2009 03:01 PM	http://IS11068-05068	2	Processing channel - Hipaa inbound data to XML. Channel uses SREG (Special Regis ters) to define destination paths . EBIXES should be attached to this channel before deployment. fore deployment.

3. On the Available Channels pane, select the channel you defined previously, *HipaaToXML Ebix*, and click *Deploy*.

The Channel Management pane reopens.

4. Select HipaaToXML_Ebix and click Start.

The red X under Status changes to a green check mark to indicate that the channel has been started. If an error or errors are displayed, take the appropriate action as instructed.

HipaaToXML_Ebix file	May 12 2009 03:06 PM 2	~	~	0 - 0 - 0 - 0
Deploy Undeploy R	edeploy Repair Start Stop			

Procedure: How to Verify the Channel

To make sure that the channel is working as expected, perform the following steps.

For more information on obtaining HIPAA sample files for testing purposes, see *Extracting HIPAA User Samples* on page 42.

1. Place a HIPAA document as test data in the C:\file_in\hipaa directory. This is the path in which HIPAA messages are received, which you specified for the listener associated with the inlet for the channel.

2. Check for the XML file in the c:\file_out\hipaa\good directory and the functional acknowledgement in the C:\file_out\hipaa\ack directory. This is the destination path you specified for the emitters associated with the outlets for the channel. The listener will detect the presence of the file in the input directory, and the copy service will copy it to the output directory, replacing the asterisk in the file name with a time stamp.

Reusing Your Channel Configuration

Using the Archive Manager feature of iWay Service Manager, you can archive your channel configuration with its associated components and import them into another Registry. They will then be available from that Registry for modification or reuse.

For details on this feature, see the *iWay* Service Manager User's Guide.

Chapter

Outbound Processing: XML to HIPAA

The iWay Integration Solution for HIPAA includes iWay Service Manager. iWay Service Manager validates an XML document based on HIPAA's published implementation guides and converts it to a document in HIPAA format.

This chapter provides the information you need to understand and implement a basic outbound message flow.

- □ The **outbound processing overview** describes the iWay business components and the processing steps in the basic outbound message flow.
- □ The **sample configuration** contains detailed instructions for configuring the basic outbound message flow. This topic guides you through each step of the configuration procedure.

In this chapter:

- HIPAA Outbound Processing Overview
- Sample Configuration for Outbound Processing: XML to HIPAA

HIPAA Outbound Processing Overview

The standard outbound process converts an XML representation of a HIPAA document to a HIPAA-formatted document.

The input document that is sent to the channel may not be in XML format. It can be any input document that should first be converted to the XML representation of the HIPAA document, which is then processed by the channel and transformed to a HIPAA document.

In a basic message flow, outbound processing consists of the following components and steps. For an illustration of the components available in the construction of a message flow, see *Using a Channel to Construct a HIPAA Message Flow* on page 23. You will define the components in the configuration instructions in *Sample Configuration for Outbound Processing: XML to HIPAA* on page 116.

Inlet

□ The **listener** picks up the input document.

Route / Process Flow

□ A process flow guides the XML-formatted HIPAA document through the next stages of the process.

Rules processing runs against the XML-formatted HIPAA document to validate its structure and content. The published HIPAA standards and user implementation guides define element types (for example, numeric, alpha, or date) and describe business rules to apply for validation.

The *XMLToHIPAATransformationAgent* obtains the message type and version from the XMLformatted HIPAA document. The appropriate transformation template is applied from the Ebix. The transformation converts the XML-formatted HIPAA document to HIPAA format.

The *XDHIPAAValidationReportAgent* creates a report (an XML document) containing the XMLformatted HIPAA document and resulting HIPAA formatted data, as well as the validation status.

If the HIPAA document did not contain any errors during the rules processing stage, it is emitted and continues to its next destination. The validation report is always emitted. In the sample process flow that is described later in this chapter, *good* validation reports are written with a file name prefix of *validation*. All other validation reports are written with a file name prefix of *error*. Information in the *error* validation reports can be routed accordingly for repair and reprocessing.

Outlet

I The HIPAA document is passed to the next step in the integration process.

Sample Configuration for Outbound Processing: XML to HIPAA

This topic provides step-by-step instructions on how to configure a basic outbound message flow for the iWay Integration Solution for HIPAA. This message flow represents the movement and tasks that are performed during the conversion of a message from XML format to HIPAA format.

The outbound configuration that is described in this topic represents the simplest possible route and is not equivalent to the outbound configuration that is described in *Working With HIPAA Inbound and Outbound Applications Using iWay Integration Tools (iIT)* on page 41. The outbound configuration in the quick start topic processes different HIPAA versions. The outbound configuration in this topic processes a single HIPAA version.

If you plan to modify the message flow that is described in this section and want more information on the supported iWay business components that you can use during the construction of a channel, see the *iWay Service Manager User's Guide*.

Accessing the iWay Service Manager Administration Console

For instructions, see Accessing the iWay Service Manager Administration Console.

Adding an Ebix to the Registry

The iWay e-Business Information Exchange (Ebix) framework supplies several Ebix files for the iWay Integration Solution for HIPAA.

An Ebix file for HIPAA is named HIPAA_Version.ebx, where Version is the transaction set number. For example, the Ebix file for HIPAA version 4010A1 is named HIPAA_4010A1.ebx.

For details on the supported HIPAA versions and transaction sets, see *Ebix-Supported Transaction* Sets on page 147.

This topic describes how to add an Ebix to the Registry on Windows and UNIX.

Tip: If you already added an Ebix to the Registry as described in *Adding an Ebix to the Registry* on page 70, you do not need to add it again for outbound processing. You can go directly to *Defining an Inlet* on page 121.

Procedure: How to Add an Ebix to the Registry on Windows

1. To access the Registry, select the *Registry* option in the blue shaded area below the iWay Service Manager banner, as shown in the following image.

iWay Service Man	ager	Management base	🔕 🥺 😨 7.0.0-CFR.1190
<u>Server</u> Registry Deplo	yments Tools		Licenses About Logout
Properties	General Properties		
General Properties	Listed below are the general properties f	or the base configuration of this server.	
Java Properties	General		
Settinas	Name / Home	ac11698 - C:/PROGRA~2/iway7/	
General Settings	Version	7.0.0-CFR.1190	
Console Settings	Build Date	PLATO December 20 2013 1658	
Java Settings	Configuration		
Register Settings	Name	base - C:/PROGRA~2/iway7/config/base	
Trace Settings	Status	Server Uptime: 1 hours, 49 minutes	
Log Settings	User Security Access	Read / Write	
Path Settings	Environment		
Backup Settings	OS / Hardware	Windows 7 (process) / x86	
	Java Info	23.7-b01 Oracle Corporation Java HotSpot(TM) Client VI	Λ
Providers	Java Memory	36.28 MB of 247.50 MB (14.7%) used	
Data Provider	Classpath	[1] C(\DDOCDA_2)\impy7\config\baca\lib*	
Services Provider	•		•
Directory Provider	Language and Locale		
XML Namespace Man	Locale / Timezone	en / America/New_York; time zone offset is -5 hours	
Provider	Language	English V Save	
Pooling Providers		The server has to be stopped, and started for the language cl	nange to take effect.
Authentication Realms			
Data Quality Providers			
Secure Shell Provider			
Schedule Provider			
SNMP Provider			
Facilities			
Activity Facility			
Correlation Facility			

2. Under Components in the left pane of the Registry, select Ebix.

Components		
Adapters		
Decryptors		
Ebix		
Emitters 🖑		

The Ebix pane opens, as shown in the following image.

Ebix eBusiness Information Exchange (ebix) are collections of standard metadata, templates, rules and schemas that are typically used by message flows related to eBusiness.									
	- Ebix								
		Filter By Name Where	Name 🔽 Equals 🗹						
		Name	Download	References	Description				
		No data matching t	he selection criteria was found.						
ſ	Add Delete Rename Copy								

3. Click *Add* to add a new Ebix. The New Ebix pane opens.

Ebix eBusiness Information Exchange (ebix) are collections of standard metadata, templates, rules and schemas that are typically used by message flows related to eBusiness.							
New Ebix							
Ebix File *	Choose an ebix file on your local machine for uploading to the server. When you click Next >>, the file will get uploaded to the server						
	C:\Program Files\iway7\etc\manager\packages\HIPAA_4010A1.ebx Browse						
< Back Next	»						

4. Browse to the directory in which the Ebix is located and select the name of the file.

For example, *HIPAA_5010X212_pipeline.ebx*.

5. Once you have selected the Ebix, click Next.

You are prompted for the name of the Ebix and an optional description.

New Ebix		
Name *	Name of the new ebix	
	HIPAA_5010X212_pipeline	
Description	Description for the new ebix	
	Ripaa 5010 pipeline ebix transactions 276 - 277	

- 6. Enter a name for the Ebix, for example, *HIPAA_5010x212_pipeline*, and an optional description.
- 7. Click Finish.

On the Ebix pane, you will see that the Ebix was successfully added. Later you will associate it with the channel for inbound processing.

Note: This procedure must be repeated for each HIPAA message set that will be processed by this channel. For example, if HIPAA 997 messages are packaged in the HIPAA_5010_pipeline.ebx file and if your channel will be processing 997 messages, then this Ebix file must be added to the Registry.

Procedure: How to Add an Ebix to the Registry on UNIX

Depending on your system configuration, there are two methods that you can use to add an Ebix to the Registry on UNIX.

- □ If you have a web browser on the UNIX machine, follow the instructions for Windows.
- □ Use FTP to download the Ebix from the *iWay7/etc/manager/packages* directory to your Windows machine and follow the instructions for Windows.

Adding Special Register Sets

In iWay Service Manager, a special register is a name-value pair that defines a variable that is carried throughout the system. Once defined, this variable is available to all components of the system. Within the HIPAA components, a best practice is to use special registers to define inputs and outputs. When packages containing channels are migrated between systems, the only changes required to deploy in the new location is to modify these special registers and build the channel. Channels may have many locations and this practice will minimize the effort required to migrate. For a complete list of system special registers that are provided, see the *iWay Service Manager Programmer's Guide*. For more information on defining a special register of your own, see the *iWay Service Manager User's Guide*.

The sample outbound channel uses a set of special registers defined as HIPAA_XML. For example:

Registers / Hipaa_XML

Register name/value sets to be used by various conduits.

The table below lists the names and values of registers that belong to register set 'Hipaa_XML'.							
	Name	Туре	Value	Description			
	Archive	string 💌	C:\file_out\hipaa\archivedi	Archive of transformed 🔨			
	Error	string 💌	C:\file_out\hipaa\errordir	EDA errors (bad formed 🔨 🗙			
	Input	string 💙	C:\file_in\hipaa\inputdir	Hippa outbound flow scans this directory			
	Output	string 💌	C:\File_out\hipaa\outputdir	Hipaa outbound flow 🔨 writes X12 to this			
	ValidationReport	string 💌	c:\file_out\hipaa\outputdir	All validation reports			
Add		string 💌		< >			

Procedure: How to Add a Special Register Set to Your Channel

To add a special register set to your channel:

1. In the left console pane of the Registry menu, select *Channels*.

The Channels pane opens.

2. In the row for your channel, click Regs for the channel you want to modify.

The Assign register pane opens.

- 3. Select a register and click Finish.
- 4. Click *Back* to return to the Channels pane.

Defining an Inlet

You will add a listener to the Registry, then associate that listener with a new inlet.

Procedure: How to Add a Listener

- 1. From the Registry menu options, select Listeners.
- 2. On the Listeners pane, click *Add* to add a new listener.
- 3. For the purpose of this example, we will show the configuration with a File listener. For details on supported protocols, see the *iWay Service Manager Protocol Guide*.

Select File from the Type drop-down list and click Next.

The configuration parameters pane opens.

Input Path *	Directory in which input messages are received. A specific file name or DOS-style pattern) can be used. Do not use file suffix.
	sreg(Hipaa_XML.Input) Browse
Destination *	Directory into which output files are stored. Specific file name is optional. Use * in file name to be replaced by timestamp, # by sequential counter
	sreg(Hipaa_XML.ValidationReport)\validation_sreg(basename)_*.xi Browse
Removal Destination	Full path file pattern asserting where input files will be moved. Use * in file name to be replaced by timestamp, # by sequential counter
	sreg(Hippa_XML.Archive) Browse
Suffix In	Limits input files to those with these extensions. Ex XML,in Do not use 💥 - mean no extension, * means any
	xml
Scan subdirectories	If true, all subdirectories will be scanned for files to process
	false
	Pick one
Do not unzip ZIP files	Pass ZIP files as a single file for processing (requires ACCEPT FLAT turned on)
	false
	Pick one
Suffix Out	Extension for output files (name is same as input file unless specified in destination parameter)

4. Supply configuration parameters for the new File listener as follows. An asterisk indicates that a parameter is required. For parameters not listed in the following table, accept the default value.

Parameter	Value
Input Path *	sreg(Hipaa_XML.Input)
	This value is a special register that uses a defined directory in which input messages are received.
	Make sure that you have created this directory; otherwise, errors will occur during deployment.

Parameter	Value
Destination *	sreg(Hipaa_XML.ValidationReport) \validationsreg(basename)_*.xml
	This value is a special register that uses a defined directory in which output messages are received.
	Note: The double underscore characters are used in the destination to escape the underscore.
	Make sure that you have created this directory; otherwise, errors will occur during deployment.
Removal	sreg(Hipaa_XML.Archive)
Destination	This value is a special register that uses a defined directory to which output messages are moved if they fail during transformation.
	Make sure that you have created this directory; otherwise, errors will occur during deployment. It is recommended to configure a removal destination when you are constructing a basic channel.
Suffix In	xml
	Input files with the extension .xml are allowed.
Suffix Out	x12
	In this example, the extension for output files is .x12.

- 5. Click Next.
- 6. On the Listeners pane, enter the name of the new listener and a brief description, as shown in the following table.

Parameter	Value
Name *	XmlToHipaa_Ebix
Description	XML to Hipaa file listener

7. Click *Finish* to add the listener.

Procedure: How to Define an Inlet

- 1. From the Registry menu options, select Inlets.
- 2. On the Inlet Definitions pane, click *Add* to add an inlet.
- 3. On the New Inlet Definition pane, enter the name of the new inlet and an optional description, as shown in the following table.

Parameter	Value
Name *	XmlToHipaa_Ebix
Description	The file inlet contains a file listener for XML to HIPAA processing.

- 4. Click *Finish* to add the inlet.
- 5. On the Construct Inlet pane, click *Add* to associate the listener with the inlet. The next pane prompts you for the component type.
- 6. Select Listener and click Next.

The next pane prompts you to select a listener.

7. Select *XmIToHipaa_Ebix*, which is the listener you added earlier for outbound processing, and click *Finish*.

The listener is added to the inlet.

Defining a Route

For this sample channel configuration, you will define a route that will invoke the XML to HIPAA validation process flow. The outcome of the validation process flow will place valid HIPAA data in a defined output folder. Invalid HIPAA data will be routed to an errors folder. A validation report will also be sent to the appropriate folder.

This section describes how to create a validation process flow using iIT Designer and bind it to a sample outbound channel as a route.

Procedure: How to Create a New Project and Start the Process Flow

To create a new project and start the process flow using iIT Designer:

- 1. From the Windows Start menu select *Programs*, *iWay* 8.0 Service Manager, tools, and then *iIT* Designer.
- 2. Connect to the repository from which you want to work, for example, iWay.
- 3. Right-click the *iWay* node and select *New Project* from the drop-down list.

The Designer Project Information dialog box opens, prompting you for a project name and optional description.

4. In the Name field, type a project name, for example, *Test*.

In the Description field, type a brief description (optional) to describe the project.

5. Click Next.

The Designer Project Bindings dialog box opens.

- 6. To create the project in the iWay Registry, select *iWay Registry* and click *Finish*. The choice of project association depends on where you intend to publish (deploy) your process flow. If you are developing a process flow for use as part of a channel, you must publish it to the iWay Registry for subsequent deployment.
- 7. The Test project node appears under the repository in which it was created (in this example, it appears under iWay).
- 8. To save the project to the repository, right-click the project node and select Save from the context menu.
- 9. Expand the Test project node to expose the project elements (Processes, Services, Transforms, and so on).
- 10. Right-click the Processes folder and select New Process from the drop-down list.

The iWay Process Configuration dialog box opens.

11. In the Name field, type XMLToHipaa_Ebix as the process flow name.

In the Description field, type a brief description (optional).

12. Click Finish.

The new XMLToHipaa_Ebix node appears under the Processes folder, and the workspace displays a Start object.



You are ready to build the XMLToHipaa_Ebix validation process flow by configuring objects to it and specifying their relationships.

Procedure: How to Configure Objects for the Process Flow

To configure objects for the process flow using iIT Designer:

1. Drag and drop the Service object from the toolbar to the workspace.

The New Service Object dialog box opens.

- 2. In the Name field, type *XMLtoHipaaTransformAgent*, and a brief description (optional) in the Description field.
- 3. Click Next.

The Service Type dialog box opens.

- 4. Select Class Name and enter com.ibi.agents.XMLToHIPAATransformAgent.
- 5. Click Next.

The Properties dialog box opens.

Service Object						
Name Type Pro	operties User Defined Prop	perties Pre-Execution Post-Execution Debug Settings				
Name	Value	Description				
* template	XMLtoHIPAA_%_^xc 💌	$^_{M}$ ML_HIPAA.xch, where [%] represents the message type type to the message type type the test of the message type type to the test of test o				
* debug	false	The transformation components are written to files in the loca				
sTerminator	7E ~ Tilde	The control character that marks the end of a specific variabl				
segSuffix	None	The Segment Suffix marks the end of the Segment. This is us				
eDelimiter	2A * Asterisk	The control character used to separate elements in a segmer				
compDelimiter	3A : Colon	The control character used to separate sub-elements/compo				
eCharacter	5C \ Backslash	The escape character is necessary if any of the EDI docume Write timestamp to log-file Inserts Group Loop in the XML Document Structure				
timestamp	false					
InsertGroupLoop	false					
Reload Services OK Cancel Help						

- 6. Set the InsertGroupLoop property to false.
- 7. In the template field, enter the following template mask, based on the HIPAA version you are processing:
 - □ For HIPAA version 4010, enter:

%_^_XML_HIPAA.xch

□ For HIPAA version 5010, enter:

XMLtoHIPAA_%_^.xch

- 8. For the debug parameter, select *false* from the drop-down list.
- 9. Click Finish.

The new Service object (XMLtoHipaaTransformAgent) appears in the workspace.

10. Select the *Start* object, right-click the *XMLtoHipaaTransformAgent* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

11. From the Event drop-down list, select OnCompletion and click OK.

This option indicates that there are no conditions that affect the path, and that the path between the two objects will always be followed.

A line appears between the objects to indicate that a relationship has been established.



12. Drag and drop the File object from the toolbar to the workspace.

The New File Object dialog box opens.

- 13. In the Name field, type *Write To Error Dir*, and a brief description (optional) in the Description field.
- 14. Click Next.

The File Type dialog box opens.

- 15. From the Type drop-down list, select *File Write*.
- 16. Click Next.

The Properties dialog box opens.

File Object							
Name Type	Properties Pre-Executi	on Post-Execution Debug Settings					
Name	Value	Description	Туре				
Source of		Source of data to write. If omitted, document will be used,	string				
* Target Dir	sreg(Hipaa_XML.Error)	The target output directory	string				
* File Pattern	errorsreg(basenam	The output file name, which can contain a ** which gets	string				
Avoid Pre	true	Should any preemitter be avoided?	boolean				
Return	input	'status': status document will be the out document. 'input'	string				
Base64 D	false	If set, the value is assumed to be in base64 notation. Onl	boolean				
* Respect T	false	If set, the emit respects the transactionality of the channel	boolean				
Call at EO	false	In streaming a last call is made AFTER the last document	boolean				
OK Cancel Help							

- 17. For the Target Directory parameter, enter a location where error data will be written, for example, sreg(*Hipaa_XML.Error*).
- 18. For the File Pattern parameter, enter error_sreg(basename)_*.xml.
- 19. For the Return parameter, select input from the drop-down list.

20. Click Finish.

The new File object (Write To Error Dir) appears in the workspace.

21. Select the XMLtoHipaaTransformAgent object, right-click the Write To Error Dir object, and select Relation from the context menu.

The Line Configuration dialog box opens.

22. From the Event drop-down list, select OnFailure and click OK.

A line appears between the objects to indicate that a relationship has been established.



23. Drag and drop the End object from the toolbar to the workspace.

The End Name and Description dialog box opens.

- 24. In the Name field, type End_Fail, and a brief description (optional) in the Description field.
- 25. Click Next.

The End Name Schema dialog box opens.

- 26. Since no schemas are used in this processing path (that is, the process flow will not be exposed as a web service), from the Schema drop-down list, select *None*.
- 27. Click Next.

The Properties dialog box opens.

28. Click Finish to accept the default values and close the dialog box.

The new End_Fail object appears in the workspace.

29. Select the *Write To Error Dir* object, right-click the *End_Fail* object, and select *Relation* from the drop-down list.

The Line Configuration dialog box opens.

30. From the Event drop-down list, select OnCompletion and click OK.



A line appears between the objects to indicate that a relationship has been established.

31. Drag and drop the Service object from the toolbar to the workspace.

The New Service Object dialog box opens.

- 32. In the Name field, type *XDHipaaValidationReportAgent*, and a brief description (optional) in the Description field.
- 33. Click Next.

The Service Type dialog box opens.

- 34. Select Class Name and enter com.ibi.agents.XDHIPAAValidationReportAgent.
- 35. Click Next.

The Properties dialog box opens.

- 36. Configure the available parameters according to your requirements.
- 37. Click Finish.

The new Service object (XDHipaaValidationReportAgent) appears in the workspace.

38. Select the XMLtoHipaaTransformAgent object, right-click the XDHipaaValidationReportAgent object, and select Relation from the context menu.

The Line Configuration dialog box opens.

39. From the Event drop-down list, select OnCompletion and click OK.



A line appears between the objects to indicate that a relationship has been established.

40. Drag and drop the Decision Test object from the toolbar to the workspace.

The New Test Object dialog box opens.

- 41. In the Name field, type *Decision Test*, and a brief description (optional) in the Description field.
- 42. Click Next.

The Test Operands dialog box opens.

File Object 📃 🗖 🔀							
Name	Туре	Properties Pre-	Executi	on Post-Execution Debug Settings			
Nam	e	Value		Description	Туре		
Sou	rce of	XPATH(/docume	nts/	Source of data to write. If omitted, document wi	string		
* Tar	get Dir	sreg(Hipaa_XML.Out		The target output directory	string		
* File	File Pattern sreg(basename)_*.x12		.x12	The output file name, which can contain a "" w	string		
Avo	oid Pre	true		Should any preemitter be avoided?	boolean		
Ret	um	input		'status': status document will be the out docum	string		
Bas	:e64 D	false		If set, the value is assumed to be in base64 not	boolean		
* Res	spect T	false		If set, the emit respects the transactionality of t_{\cdots}	boolean		
Call	at EO	false		In streaming a last call is made AFTER the last	boolean		
	OK Cancel Help						

43. In the Operand One field, enter the following:

XPATH(/documents/ValidationReport/Report/Errors/error)

- 44. From the Operation drop-down list, select Is Not Null.
- 45. Click Next.

The Value Occurrences dialog box opens.



- 46. Ensure that *Unique* is selected from the available options.
- 47. Click Finish.

The new Decision Test object appears in the workspace.

48. Select the *XDHipaaValidationReportAgent* object, right-click the *Decision Test* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

49. From the Event drop-down list, select OnCompletion and click OK.



A line appears between the objects to indicate that a relationship has been established.

50. Drag and drop the File object from the toolbar to the workspace.

The New File Object dialog box opens.

- 51. In the Name field, type *Write Good File*, and a brief description (optional) in the Description field.
- 52. Click Next.

The File Type dialog box opens.

- 53. From the Type drop-down list, select File Write.
- 54. Click Next.

The Properties dialog box opens.

File Object					
Name Type Properties Pre-Execution Post-Execution Debug Settings					
Name	Value	Description	Туре		
Source of	XPATH(/documents/	Source of data to write. If omitted, document wi	string		
* Target Dir	sreg(Hipaa_XML.Out	The target output directory	string		
* File Pattern	sreg(basename)_*.x12	The output file name, which can contain a "" w	string		
Avoid Pre	true	Should any preemitter be avoided?	boolean		
Return	input	'status': status document will be the out docum	string		
Base64 D	false	If set, the value is assumed to be in base64 not	boolean		
* Respect T	false	If set, the emit respects the transactionality of t	boolean		
Call at EO) false In streaming a last call is made AFTER the last		boolean		
OK Cancel Help					

55. For the Source of Data parameter, enter the following:

XPATH(/documents/output)

56. For the Target Directory parameter, enter the following location where valid data will be written:

sreg(Hipaa_XML.Output)

57. For the File Pattern parameter, enter the following:

sreg(basename)_*.x12

- 58. For the Return parameter, select *input* from the drop-down list.
- 59. Click Finish.

The new File object (Write Good File) appears in the workspace.

60. Select the *Decision Test* object, right-click the *Write Good File* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

61. From the Event drop-down list, select OnCustom.

🚊 🧰 Processes					
🕀 📸 Samp	oles.S	ciFiBooks.1			
🕀 📸 Samp	oles.P	ictures.Load.1			
🕀 👸 Samp	🗊 🐯 Samples.Pictures.RetrieveAlt				
⊡ 🔁 XMLT	😟 🤠 XMLToHipaa_Ebix				
표 🚞 Services		New Object	►		
표 🚞 Transfor	🗄 🛅 Transfor 💦 Save				
🕀 🛅 Adapters					
🗄 🚞 Emitters		Export			
🗄 🚞 Schemas		Test	►		
- 🥔 an e 👘 🖉					

- 62. In the Case of section, select false.
- 63. Click OK.

A line appears between the objects to indicate that a relationship has been established.



64. Drag and drop the End object from the toolbar to the workspace.

The End Name and Description dialog box opens.

- 65. In the Name field, type *End_Success*, and a brief description (optional) in the Description field.
- 66. Click Next.

The End Name Schema dialog box opens.

- 67. Since no schemas are used in this processing path (that is, the process flow will not be exposed as a web service), from the Schema drop-down list, select *None*.
- 68. Click Next.

The Properties dialog box opens.

69. Click Finish to accept the default values and close the dialog box.

The new End_Success object appears in the workspace.

70. Select the *Write Good File* object, right-click the *End_Success* object, and select *Relation* from the drop-down list.

The Line Configuration dialog box opens.

71. From the Event drop-down list, select OnCompletion and click OK.

A line appears between the objects to indicate that a relationship has been established.



72. Select the *Decision Test* object, right-click the *End_Success* object, and select *Relation* from the context menu.

The Line Configuration dialog box opens.

73. From the Event drop-down list, select OnCustom.

Case of:		
Case	Туре	Description
📃 📲 🕻 OnError	Stock	Error
📃 🖻 🕻 On Success	Stock	Success
📃 🖻 🕻 OnDefault	Stock	Used for a Decision of type Switch
🗹 🎯 🕻 true	Custom	
📃 🏽 📽 🖉 false	Custom	
Dblclick here to Add		

- 74. In the Case of section, select true.
- 75. Click OK.

A line appears between the objects to indicate that a relationship has been established.

Name	Value	Description	Туре
Source of	XPATH(/documents/output)	Source of data to write. If omitted, document will be used, else specify	string
* Target Dir	sreg(XML.Output)	The target output directory	string
* File Pattern	sreg(basename)_*.x12	The output file name, which can contain a $^{\mbox{\tiny NS}}$ which gets expanded to	string
Avoid Pre	true	Should any preemitter be avoided?	boolean
Return	input	'status': status document will be the out document. 'input': in document	string
Base64 D	false	If set, the value is assumed to be in base64 notation. Only applicable is	boolean
* Respect T	false	If set, the emit respects the transactionality of the channel. If not set, t	boolean
Call at EO	false	In streaming a last call is made AFTER the last document. Does this S_{\cdots}	boolean

The process flow is now complete.

76. To save the process flow, right-click the *XMLToHipaa_Ebix* node in the left pane and select Save from the context menu.

Case of:		
Case	Туре	Description
📃 📲 🕻 OnError	Stock	Error
📃 🏁 🕻 On Success	Stock	Success
📃 🏁 🕻 OnDefault	Stock	Used for a Decision of type Switch
📃 🖻 🛱 true	Custom	
🗹 🎯 📽 false	Custom	
Dblclick here to Add		

Now you need to validate the process flow and publish it to the Registry of the iWay Service Manager Administration Console for use in the route of a channel for outbound processing.

Validating a process flow ensures that its structure is correct. Publishing a process flow makes it available in the Registry for use in a channel configuration. For instructions on validating and publishing the process flow, see the *iWay Integration Tools Designer User's Guide*.

77. Close iIT Designer.

Your next step is to add a new route to the Registry using the iWay Service Manager Administration Console and associate the process flow with it.

Procedure: How to Define a Route and Associate the Process Flow With It

To define a route and associate the process flow with It:

- 1. From the Registry menu options in the iWay Service Manager Administration Console, click *Routes*.
- 2. On the Route Definitions pane, click *Add* to add a route.
- 3. On the New Route Definition pane, enter a name for the route and an optional description, as shown in the following table.

Parameter	Value
Name *	XmltoHipaaEbix
Description	This route will invoke the XML to HIPAA validation process. The outcome of the validation process will place valid HIPAA data in your valid outbound folder. Invalid HIPAA will be routed to an errors folder. A validation report will also be sent to the appropriate folder.

- 4. Click Finish.
- 5. On the Construct Route pane, click Add.

You are prompted for the type of component to associate with the route.

- 6. Select Process and click Next.
- 7. The next pane prompts you to select a process. Select the process flow you created earlier with iIT Designer, *XMLToHipaa_Ebix*, and click *Finish*.

The route, with its associated process flow, has been successfully defined.

Defining an Outlet

For the iWay Integration Solution for HIPAA, you will add an emitter to the Registry, then associate it with a new outlet.

Procedure: How to Add an Emitter for an Error Validation Report

To add an emitter that will emit an error validation report and error file due to the XML to HIPAA validation process:

- 1. From the Registry menu options, select *Emitters*.
- 2. On the Emitters pane, click *Add* to add an emitter.

The next pane prompts you for the emitter type.

3. Select File from the drop-down list and click Next.

The File Emitter configuration parameters pane opens.

Configuration parameters for new emitter of type File			
Destination *	path to file, * replaced with timestamp		
	<pre>sreg(Hipaa_XML.Error)\errorsreg(basename)_*.xml</pre>		
Create Directory	Create directory if it doesn't exist		
	true		
	Pick one		

4. In the Destination field, enter the following:

sreg(Hipaa_XML.Error)\error__sreg(basename)_*.xml

- 5. From the Create Directory drop-down list, select *true*.
- 6. Click Next.
- 7. On the Emitters pane, enter the name of the new emitter and an optional description, as shown in the following table:

Parameter	Value		
Name *	XmltoHipaaEbix_XML_error		
Description	XmltoHipaa Ebix_XML error		

8. Click Finish to add the emitter.

The following example shows the structure of an error validation report that is returned if the input XML document is invalid.

Document Outline	$\mathbf{t} \times$	Node	Value
documents		?=? xml	version="1.0" encoding="ISO-8859-1"
Second Se	S ValidationReport		
🖻 🔇 🖇 Report		ValidationReport	
Errors		E- & Report	
() error		Errors	
		error	Rule Violation: [X12_810_004010810.BIG01_Date] '20021145' Date format (20021145) is invalid
		🖻 🔇 🔊 input	
		i ≪ S output	
			ISA*00* *00* *12*NOTP *12*NOTP *080501*1700*U*00401*00000001*0*P*>\n~G5*IN*

Procedure: How to Add an Emitter for a Valid Validation Report

To add an emitter that will emit a valid validation report due to the XML to HIPAA validation process:

- 1. From the Registry menu options, select *Emitters*.
- 2. On the Emitters pane, click *Add* to add an emitter.

The next pane prompts you for the emitter type.

3. Select *File* from the drop-down list and click *Next*.

The File Emitter configuration parameters pane opens.

Configuration parameters for new emitter of type File				
Destination *	path to file, * replaced with timestamp			
	$sreg({\sf Hipaa}_X{\sf ML.ValidationReport}) \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$			
Create Directory	Create directory if it doesn't exist			
	true			
	Pick one			

4. In the Destination field, enter the following:

sreg(Hipaa_XML.ValidationReport)\validation_sreg(basename)_*.xml

- 5. From the Create Directory drop-down list, select *true*.
- 6. Click Next.
- 7. On the Emitters pane, enter the name of the new emitter and an optional description, as shown in the following table:

Parameter	Value		
Name *	XmltoHipaaEbix_XML_validation		
Description	XmltoHipaaEbix_XML validation report		

8. Click *Finish* to add the emitter.

The following example shows the structure of a valid validation report that is returned if the input XML document is valid.

Document Outline 🛛 🕮 🛪	Node	Value			
documents	- ?? xml	version="1.0" enco	ding="ISO-8859-1"		
⇒ 《 》 ValidationReport	E- & documents				
🖻 🔇 📎 Report	- S ValidationReport				
Success	E- & Report				
□ «» input	Success				
± & x12_810_004010	E- (> input				
So oupur	★ 《》 X12_810_004				
	E- & output				
	L. 🗈	ISA*00* *00*	*12*NOTP	*12*NOTP	*080501*1700*U*00401*000000001*0*P*>~GS*IN*N.

Procedure: How to Define the Outlets

Now that you have added two emitters to the Registry, you are ready to define the required outlets. Each emitter will be associated with a corresponding outlet.

- 1. From the Registry menu options, select Outlets.
- 2. On the Outlet Definitions pane, click *Add* to add the first outlet.
- 3. On the New Outlet Definition pane, enter the name of the first new outlet and an optional description, as shown in the following table.

Parameter	Value
Name *	XmltoHipaaEbix_XML_error
Description	Outlet which will contain error validation report and error file due to the XML to HIPAA validation process.

- 4. Click Finish to add the outlet.
- 5. On the Construct Outlet pane, click *Add* to associate the emitter with the outlet. The next pane prompts you for the component type.
- 6. Select *Emitter* and click Next.

The next pane prompts you to select an emitter.

- 7. Select *XmltoHipaaEbix_XML_error*, which is the first emitter you added earlier, and click *Finish*.
- 8. On the Outlet Definitions pane, click Add to add the second outlet.
- 9. On the New Outlet Definition pane, enter the name of the second outlet and an optional description, as shown in the following table.

Parameter	Value
Name *	XmltoHipaaEbix_XML_validation
Description	Outlet which will contain valid validation report produced by the validation process.

- 10. Click *Finish* to add the outlet.
- 11. On the Construct Outlet pane, click Add to associate the emitter with the outlet.

The next pane prompts you for the component type.

12. Select Emitter and click Next.

The next pane prompts you to select an emitter.

- 13. Select XmltoHipaaEbix_XML_validation, which is the second emitter you added earlier.
- 14. Click Finish.

Defining a Channel

Now that you have defined the required components for the outbound channel, you are ready to add the channel to the Registry and associate the conduits with it. At this time you will also add the route to the channel.

Procedure: How to Define a Channel

To define a channel:

- 1. From the Registry menu options, select *Channels*.
- 2. On the Channel Definitions pane, click *Add* to add a channel.
- 3. On the New Channel Definition pane, enter the name of the new channel (for example, *XmIToHipaa_Ebix*) and an optional description. Then click *Finish* to add the channel.
- 4. On the Construct Channel pane, click *Add* to associate the inlet, route, and outlets with the channel.

You are prompted to associate components with the channel.

- 5. Select *Inlet* and click *Next*. The next pane prompts you to select an inlet.
- 6. Select XmlToHipaa_Ebix, which you defined earlier, and click Finish.

The inlet is associated with the channel. Now you need to associate a route with the channel and set it to the default.

7. On the Construct Channel pane, click Add.

The next pane prompts you for the component type.

8. Select Route and click Next.

On the next pane, you are prompted to select a route.

9. Select XmltoHipaaEbix, which you defined earlier, and click Finish.

The Construct Channel pane reopens.

10. Click the *minus sign* (-) under Conditions to set this route as the default.

Туре	Conditions	Move	D
Inlet			
Route	لیک کی راس		
Add	Dele Click to	set defau	IF.

- 11. On the Construct Channel pane, click Add to add the next component.
- 12. When prompted for the component type, select Outlet and click Next.
- 13. Select the two outlets you defined earlier, *XmltoHipaaEbix_XML_error* and *XmltoHipaaEbix_XML_validation*.
- 14. Click Finish.
- 15. To set a condition for the outlets, on the Construct Channel pane, click the *plus sign* (+) under Conditions for the specific outlet.

Туре	Conditions
Inlet	
Route	4
Outlet	la 🎝
	\Box

The Set Condition pane opens.

16. In the Condition input field, enter the appropriate conditional expression, and click *Update*.

The following table lists the expression that must be entered for each outlet.

Outlet	Expression
XmltoHipaaEbix_XML_ validation	<pre>_isxml() and sreg(iwaf.validationSuccess) = true</pre>
XmltoHipaaEbix_XML_ error	<pre>_isxml() and sreg(iwaf.validationSuccess) != true</pre>

For details on supported conditions, see the topic on using functions in the *iWay* Service *Manager* User's Guide.

Procedure: How to Build the Channel

- 1. From the Registry menu options, select Channels.
- 2. On the Channel Definitions pane, select the channel for outbound processing defined previously, *XmlToHipaa_Ebix*, and click *Build*.

The results of the build are displayed on the right pane.

Channels

Channels are the pipes through which messages flow in iWay Service Manager. A Channel is defined as a named container of Routes (Transformers + Processes), controlled by Routing Rules and bound to Ports (Listeners/Emitters).

el
Message
Start
Validating Channel
Channel is valid
Validating Inlet
Inlet is valid
Validating Routes
Routes are valid
Validating Outlets
Outlets are valid
Build Successful
End
Channel archive C:\PROGRA~1\iway56sm\etc\repository\manager\car\XmlToHipaa_Ebix\ XmlToHipaa_Ebix.2\XmlToHipaa_Ebix.car has been created/updated

<< Back
3. Review the results of your build and then click Back.

If an error or errors are displayed in the Message column, take the appropriate action as instructed.

Procedure: How to Deploy the Channel

Deployment is the mechanism by which a channel moves from being stored in the Registry to becoming active in iWay Service Manager. For more information on deployment, see the *iWay* Service Manager User's Guide.

- 1. Select the *Deployments* option.
- 2. On the Channel Management pane, click *Deploy*.

	XmIToHipaa_Ebix	May 12 2009 06:34 PM	http://IS11068-05068	2	Processing channel - XML inbound data to Hipaa. This channel will also produce a Validation report. Channel uses SREG (Special Registers) to define destination paths. Channel also uses route-pflow. Pflow will place X12 data in its appropria paths. Channel also uses route-pflow. Pflow will place X12 data in its appropriate destination folders based on X12 validation rules. EBIXES should be attached to this channel before deployment.
<<	Back Deploy	Get Versions			

3. On the Available Channels pane, select the channel you defined previously, *XmIToHipaa_Ebix*, and click *Deploy*.

The Channel Management pane reopens.

4. Select XmIToHipaa_Ebix and click Start.

The red X under Status changes to a green check mark to indicate that the channel has been started. If an error or errors are displayed, take the appropriate action as instructed.

XmlToHipaa_Ebix file	May 12 2009 06:39 PM 2	\checkmark	✓	0-0-0-0
Deploy Undeploy R	edeploy Repair Start St	op		

Procedure: How to Verify the Channel

To make sure that the channel is working as expected, perform the following steps.

For more information on obtaining HIPAA sample files for testing purposes, see *Extracting HIPAA User Samples* on page 42.

- 1. Place an XML file as test data in the input directory (C:\file_in\hipaa\inputdir). This is the path in which XML messages are received, which you specified for the listener associated with the inlet for the channel.
- 2. Check for the HIPAA output document and validation report in the output directory (C: \File_out\hipaa\outputdir). This is the destination directory you specified for the listener.
- 3. Confirm that the output has been converted to HIPAA format.

Reusing Your Channel Configuration

Using the Archive Manager feature of iWay Service Manager, you can archive your channel configuration with its associated components and import them into another Registry. They will then be available from that Registry for modification or reuse.

For details on this feature, see the *iWay* Service Manager User's Guide.



Ebix-Supported Transaction Sets

This topic describes the HIPAA transaction sets supported by the iWay Integration Solution for HIPAA in the Ebix files supplied with the product.

In this appendix:

Supported HIPAA Transactions

Supported HIPAA Transactions

The following table lists the HIPAA ANSI X12N transaction sets that are supported by the iWay Integration Solution for HIPAA.

Version	Document	Version	Document
4010A1	All	5010X218	820
5010	997, 999	5010X220A1	834
_	(same as X12 5010)		
5010X186	824	5010X221A1	835
5010X187	269Que, 269Res	5010X222A1	837P
5010X210	275	5010X223A2	8371
5010X211	275	5010X224A2	837D
5010X212	276, 277	5010X225A2	837R
5010X213	277	5010X231A1	999
5010X214	277u	5010X279A1	270, 271
5010X215	2780ri, 278Rsp	5010X299	837 for HIX
5010X216	278Adv, 278Inf, 278Cmp, 278Rej, 278Cpl	5010X306	820 for HIX

Version	Document	Version	Document
5010X217	278Aut, 278Can, 278Req, 278Res	5010X307	834 for HIX



Using iWay Integration Tools to Configure an Ebix for HIPAA

This section describes how to use iWay Integration Tools (iIT) to configure an e-Business Information Exchange (Ebix) file for HIPAA.

In this appendix:

- Using iIT to Configure an Ebix File for HIPAA Overview
- Using iIT to Configure an Ebix File for HIPAA Prerequisites
- Downloading and Extracting an Ebix
- □ Working With iWay Integration Tools (iIT)

Using iIT to Configure an Ebix File for HIPAA Overview

You can use iWay Integration Tools (iIT) to import, edit, export, and work with e-Business Information Exchange (Ebix) files for HIPAA. The topics in this appendix describe how to:

□ Import a HIPAA 5010 837 Ebix into iIT.

- Add a qualifier at the loop 2300 level under the CLM segment to the HIPAA Ebix.
- **L** Export the edited Ebix to a physical location.

The edited Ebix can be returned and then tested with the appropriate HIPAA 837 message.

Using iIT to Configure an Ebix File for HIPAA Prerequisites

This section provides a list of prerequisites for using iWay Integration Tools (iIT) to configure an Ebix for HIPAA:

- □ Have a working knowledge of iIT and HIPAA.
- □ Ensure the iWay HIPAA adapter is installed.
- Ensure iIT Version 8.0 is installed.

Downloading and Extracting an Ebix

This section describes how to download and extract an Ebix.

Procedure: How to Download and Extract an Ebix

To download and extract an Ebix:

- 1. Download the HIPAA_ebixs.zip file from http://techsupport.informationbuilders.com.
- 2. Unzip the downloaded *HIPAA_ebixs.zip* file and save *HIPAA_5010X299.ebx* into any physical location on your local drive.

For example, this Ebix contains the HIPAA 837 document.

UTDAA FOTOVODOAT shu	2/0/2015 A-44 DM	COV CI-	100 1/0
HIPAA_5010X220A1.ebx	2/8/2015 4:44 PM	EBX File	108 KB
HIPAA_5010X221A1.ebx	2/8/2015 4:44 PM	EBX File	82 KB
HIPAA_5010X222A1.ebx	2/8/2015 4:44 PM	EBX File	165 KB
HIPAA_5010X223A2.ebx	2/8/2015 4:44 PM	EBX File	227 KB
HIPAA_5010X224A2.ebx	2/8/2015 4:44 PM	EBX File	197 KB
HIPAA_5010X225A2.ebx	2/8/2015 4:44 PM	EBX File	149 KB
HIPAA_5010X231A1.ebx	2/8/2015 4:44 PM	EBX File	72 KB
HIPAA_5010X279A1.ebx	2/8/2015 4:44 PM	EBX File	228 KB
HIPAA_5010X299.ebx	2/8/2015 4:44 PM	EBX File	151 KB
HIPAA_5010X306.ebx	2/8/2015 4:44 PM	EBX File	78 KB
HIPAA_5010X307.ebx	2/8/2015 4:44 PM	EBX File	126 KB
HIPAA_ebixs.zip	3/12/2015 12:08 P	M WinRAR ZIP archive	4,383 KB

Note: Ensure all folders used for the extracted HIPAA_ebixs.zip file do not have any blank spaces in the folder name.

Working With iWay Integration Tools (iIT)

This section describes how to import, edit, and export an Ebix using iWay Integration Tools (iIT).

Procedure: How to Import an Ebix

- 1. Start iWay Integration Tools (iIT).
- 2. Right-click the Integration Explorer pane, click *New*, and then select *Integration Project* from the context menu, as showin in the following image.

🦼 Integration - iWay Integra	ation Tools
File Edit Navigate Search I	Project Run Window Help
] 🖸 • 😸 🐘 🛆] 😤 !	🔉 💱 🐹 🏇 • 💽 • 🗛 • 🛷 • 🐲
🖌 Inte 🕴 😈 iWa 🗯	ubra
(e =) 1	
New 🕨	A Integration Project
Copy	Project
Paste	() Application
Duplicate	E Channel
Delete	n Process Flow
🚵 Import	Transform
🛃 Export	Register Set
🔊 Refresh	G Schema Set
	10 In Expression
	Example
	Ctrl+N

3. Enter a new Integration Project name, for example, *HIPAA_Ebix_edit_sample_proj*, in the Project name field, and then click *Finish*, as showin in the following image.

A New Integration Project
Integration Project
Create a new Integration project.
Project name HIPAA_Ebix_edit_sample_proj
Project location
Use default
Directory C:\OFFSHORE\IT_Builds\7.0.4_trunk_13-mar-2015\IT-7.0.4\HIP/ Browse,
Additional options
Create integration folders
Target Server Version 7.0.4-SNAPSHOT
Install additional Target Server Version
Finish Cancel

4. Right-click the Integration Explorer pane and select *Import* from the context menu, as shown in the following image.

nte 🕅	😇 iWa 📓 Libra
	bix_edit_sample_proj
🗁 Adap	ters
🧀 Applic	cations
	nels
- 🗁 Ebixe	New
- Come Flower	Go Into
🗁 Regis	Open in New Window
Scher	
- Trans	Сору
····· 🗁 XML	Paste
	Duplicate
	💢 Delete
	Move
	Rename
	N- Import
	- A Expect
	Export
	🔊 Refresh
	Validate
	Run As
	Debug As 🕨
🗄 Outline 🖾	Team 🕨
An outline is not a	Compare With
	Restore from Local History
	Source
	Integration Tools
	Properties

5. In the Import wizard, expand *iWay Integration*, select *Ebix*, and then click *Next*, as shown in the following image.



6. Click the *ellipsis button* (...), as shown in the following image.

🛃 Import		
General Prop O The name field	e rties Page is required.	
Project Folder	/HIPAA_Ebix_edit_sample_proj/Ebixes	Browse
Import		
Name	1	
Description		X
		-
Target Server Vers	ion 7.0.4-SNAPSHOT	-
	Install additional Target Server Version	
	I Create in current folder	
?	< Back Next > Einish	Cancel

The Open dialog is displayed.

7. Select the downloaded *HIPAA_5010X299.ebx* file from the physical drive location and then click *Open*, as shown in the following image.

🦼 Open			×
00 📕 00	S (C:) ▼ OFFSHORE ▼ Ebix_Builds ▼ HIPAA	👻 🛃 Search HIPAA	2
Organize 🔻 New f	older		8H - 🖪 😧
🔶 Favorites	Name ^	Date modified T	ype 🔺
🧮 Desktop	HIPAA_5010X218.ebx	2/8/2015 4:44 PM E	BX File
Downloads	HIPAA_5010X220A1.ebx	2/8/2015 4:44 PM E	BX File
Recent Places	HIPAA_5010X221A1.ebx	2/8/2015 4:44 PM E	BX File
🔚 Libraries	HIPAA_5010X222A1.ebx	2/8/2015 4:44 PM E	BX File
Documents	HIPAA_5010X223A2.ebx	2/8/2015 4:44 PM E	BX File
J Music	HIPAA_5010X224A2.ebx	2/8/2015 4:44 PM E	BX File
Pictures	HIPAA_5010X225A2.ebx	2/8/2015 4:44 PM E	BX File
Subversion	HIPAA_5010X231A1.ebx	2/8/2015 4:44 PM E	BX File
Videos	HIPAA_5010X279A1.ebx	2/8/2015 4:44 PM E	BX File
1 Computer	HIPAA_5010X299.ebx	2/8/2015 4:44 PM E	BX File
🚢 OS (C:)	HIPAA_5010X306.ebx	2/8/2015 4:44 PM E	BX File
C on THIRU	HIPAA_5010X307.ebx	2/8/2015 4:44 PM E	BX File 👻
🖵 D on THIRU	<u> </u>		
	File name: HIPAA_5010X299.ebx	Ebix File (*.ebx)	•
		Open	Cance

8. Click *Next*, as shown in the following image.

nport 🛃						
General Properties Page						
Please enter a name	and description for this imported ebix.					
Project Folder	/HIPAA_Ebix_edit_sample_proj/Ebixes Browse					
Import	C:\OFFSHORE\Ebix_Builds\HIPAA\HIPAA_5010X299.ebx					
Name	HIPAA_5010X299					
Description	A					
						
Target Server Versio	n 7.0.4-SNAPSHOT					
	Install additional Target Server Version					
	Create in current tolder					
	< Back Next > Einish Cancel					

9. Expand *HIPAA_5010X299* and select 5010X299 in the Ebix pane. Select 837 in the Ebix Entries pane, and then click *Finish*, as shown in the following image.

nport 🛃		
Ebix Entries		
Select ebix version to view av	ailable ebix entries.	
Transform ebix are specially d represents transform configur Select ebix version to view av	esigned archive files that contain ebix entries. Ebix en ation and dependencies used by iWay transform engir ailable ebix entries.	try ne.
Import as System Ebix		
Ebix	Ebix Entries	
HIPAA_5010X299 S010X299	······	
Description:		
Entry: 837 Run Time Mode: N/A Description:		A I
?	< Back Next > Einish Car	ncel



Your iIT interface should now resemble the following image:

Procedure: How to Edit an Ebix

1. Click the *HIPAA_*837_005010X299 tab and navigate to the 02 [Facility Code Qualifier] node. Expand *EDI*, 837, 2000A, 2000B, 2300, *CLM* [Health Claim], and then C023 [Health Care Service Location Information], as shown in the following image.



2. Right-click the 02 [Facility Code Qualifier] element and then click Properties from context menu, as shown in the following image.



3. Scroll down to view the Domain value, and add *B* into the Domain value field in the properties window.

Properties 🕴 🧐 Error Log 📮 Console 😫 Problems						
[02]						
	MinLength	1				
General	MaxLength	2				
	Pad					
	PadChar					
	Align					
	Notes					
	Domain	A,B				
	L					
		m				

4. Save your edited Ebix by clicking the Save icon, which is located near the File menu. If you are using a Windows platform, you can also use the shortcut key CTRL+S to save your work.

An asterisk (*) character appears next to the file name until you have saved the edited changes, as shown in the following image.



5. Click on Yes to confirm your changes.

🦼 Build	Working Set	×
0	'Rebuild entry on metadata change' is enabled. 'Preserve existing mappings' is disabled. 'Rebuild all mappings from a dictionary' is enabled. 'Rebuild schema files' is enabled. 'Rebuild rule files' is enabled.	
E Do	n't show this dialog again.	
	Yes No	



Your iIT interface should now resemble the following image:

Note: The asterisk (*) character will disappear once the edited Ebix has been saved successfully.

6. Click the *HIPAA_*837_005010X299 tab and navigate to the 02 [Facility Code Qualifier] node. Expand *EDI*, 837, 2000A, 2000B, 2300, *CLM* [Health Claim], and then C023 [Health Care Service Location Information], as shown in the following image.



7. Repeat steps 2 through 4 for each Ebix that you need to edit.

Procedure: How to Export an Ebix

To export an Ebix:

1. Right-click the *HIPAA_5010X299* Ebix from the Integration Explorer window and then select the *Export* option from the context menu, as shown in the following image.



2. Expand the *iWay Integration* folder, select *Ebix*, and then click *Next*, as shown in the following image.



3. Expand the iIT project (for example, *HIPAA_Ebix_edit_sample_proj*) from the Export wizard and expand *Ebixes*, *HIPAA*, *HIPAA_5010X299*, select the 5010X299 folder from the left pane of Export wizard, and then select the 837 check box on the right pane, as shown in the following image.

A Export				_ 🗆 🗙
Export Ø Please enter a destination d	irectory.			
HIPAA_Ebix_edit_sam	ple_proj 10X299 299	☑ ₩ 837		
To directory:			Select All	Browse
?	< Back	vext >	Finish	Cancel

4. Click *Browse* and choose a folder location to store the Ebix, and then click *Next*, as shown in the following image.

A Export			_ 🗆 🗵		
Export					
Export Ebix resources to the local file system.					
HIPAA_Ebix_edit_sample_proj Adapters Applications Channels Channels Ebixes HIPAA_Ebix_9 HIPAA Down HIPAA_5010X299 S010X299 Flows Registers Schemas Transforms XML	837				
To directory: C:\Users\Administrator\Deskto	P	Select All	Browse		
? <u><b< u="">ack</b<></u>	<u>N</u> ext >	Einish	Cancel		

5. Provide a valid name for the Ebix in the Name field, select *Pipeline* from the Runtime Mode drop-down list, add a description (optional), and then click *Finish*, as shown in the following image.

🔏 Export				_ 🗆 🗙
Export Ebix				
Create a new Ebix				
Create a new transform eb Transform ebix are specially and dependencies used by a new ebix and ebix entry f	ix by first spe y designed are iWay transfor for specified t	cifying a name chive files that rmation engine ype.	and description of a contain transform c . This wizard will allo	a new ebix. onfiguration w you to create
Name:				
Custom_Name_5010_837				
Ebix Type:				
HIPAA				
Runtime Mode:				
Pipeline				•
Standard Pipeline				
r iden ie				
Add description here				×
2	< Back	Next >	Finish	Cancel

Your exported Ebix is now available in the specified location.



Using HIPAA Special Register (SREG) Types

This section describes the Special Register (SREG) types that are created during HIPAA to XML transactions and 997/999 creation.

In this appendix:

HIPAA Special Register (SREG) Types

HIPAA Special Register (SREG) Types

New Special Registers (SREGs) are available for HIPAA preparsers and HIPAA preemitters.

```
<variable type="USR" name="edi.transactionID" otype="0">823</variable>
<variable type="USR" name="edi.type" otype="0">X12</variable>
<variable type="USR" name="edi.version" otype="0">004010</variable>
```

These may be used to route your data by placing them in your process flow.

A new SREG (edi.ackstatus) is available for the acknowledgement agent. This SREG will contain the AK501 status from the 997 that corresponds to each XML output file. This value can be used to route error data (for example, a failed 997) from standard processing.

During HIPAA to XML transactions and 997 creation, the following types of SREGs are created:

- SYS (System) These SREGs exist until you restart iWay Service Manager.
- USR/DOC These SREGs exist throughout the life of the document.
- **General CFG** These SREGs are configuration related.

SEGMENT COUNT

```
1. <variable name="SEGCOUNT" type="USR">20</variable>
2. <variable name="basename" type="DOC">stephan_850_bad</variable>
3. <variable name="console-master-port" type="SYS">9999</variable>
CORRELATION ID
4. <variable name="correlid" type="USR">000001000</variable>
5. <variable name="doclocation" type="SYS">config</variable>
END OF STREAM FLAG
```

```
6. <variable name="eos" type="USR">1</variable>
7. <variable name="extension" type="DOC">x12</variable>
8. <variable name="filename" type="DOC">stephan_850_bad.x12</variable>
```

FROM PARTY

```
9. <variable name="fromparty" type="USR">NOTP </variable>
```

GROUP CONTROL NUMBER - GE

10. <variable name="ge_groupctlnumber" type="USR">1000</variable>

NUMBER OF TRANSACTIONS - GE

```
11. <variable name="ge_numtransactions" type="USR">1</variable>
12. <variable name="ibse-port" type="CFG">9000</variable>
```

INTERCHANGE CONTROL NUMBER - IEA

13. <variable name="iea_interchangectlnum" type="USR">000001000</variable>

VALIDATION REPORT/ACK

```
14. <variable name="iwaf.ack999" type="USR" otype="0">ISA*00*
*00*
            *01*INTRCHNG RCVR I*01*INTRCHNG SNDR
I*110914*1410*^*00501*100000001*0*T*:!

GS*FA*APPLCTN RCVR*APPLCTN SNDR*20110914*1410*1*X*005010X231A1!

ST*999*0001*005010X231A1!

AK1*HS*1*005010X279A1!

AK2*270*005010X279A1!

IK3*ST*0**8!

CTX*SITUATIONAL TRIGGER*ST*0**2!

IK5*R*5!

AK9*R*1*1*0!

SE*8*0001!

GE*1*1!

IEA*1*100000001!

</variable>
```

```
15. <variable name="iwaf.validationReport" type="USR">ISA*00*
*00*
              *12*NOTP
                                 *12*NOTP
*OAOAOA*OAOA*U*00401*000001000*0*P*&qt;
GS*FA*NOTP*NOTP*OAOAOAOA*OAOA*1000*X*004010
ST*997*0001
AK1*PO*1000
AK2*850*00000010
AK3*DTM*6**8
AK4*2**8*200100
AK5*R*5
AK9*E*1*1*1
SE*8*0001
GE*1*1000
IEA*1*000001000
</variable>
16. <variable name="iway.eos" type="DOC">true</variable>
17. <variable name="iwayconfig" type="SYS">base</variable>
18. <variable name="iwayhome" type="SYS">C:/Program Files/iWay7/</variable>
19. <variable name="iwayversion" type="SYS">6.1.100.SM</variable>
20. <variable name="iwayworkdir" type="SYS">C:/Program Files/iWay7/config/
base</variable>
21. <variable name="locale" type="SYS">en_us</variable>
22. <variable name="name" type="SYS">EDI_XML</variable>
```

NUMBER OF FUNCTIONAL GROUPS

```
23. <variable name="numfunctionalgroups" type="USR">1</variable>
24. <variable name="parent" type="DOC">c:\testing\edix12\input</variable>
25. <variable name="protocol" type="SYS">FILE</variable>
26. <variable name="source" type="DOC">C:\testing\edix12\input
\stephan_850_bad.x12</variable>
```

SPLIT COUNT

```
27. <variable name="splitcount" type="USR">1</variable>
28. <variable name="tid" type="DOC">EDI_XML-FILE-W.EDI_XML.
1_20080605152319600Z</variable>
```

TRANSACTION ID

29. <variable name="edi.transactionID" type="USR">850</variable>

VERSION

30. <variable name="edi.version" type="USR">004010</variable>



Using HIPAA Separators and Terminators

All HIPAA documents are embedded with tokens that are separated by special characters called separators and terminators. These separators and terminators are used to identify:

- element separators
- □ sub-element separators
- segment terminators

This appendix provides a list of the separators and terminators that are allowed during the configuration of preparsers and preemitters in iWay Service Manager.

In this appendix:

HIPAA Separators and Terminators

HIPAA Separators and Terminators

Hex	Char	Hex	Char	Hex	Char
01	SOH	16	SYN	2F	/
02	STX	17	ETB	ЗА	:
03	ETX	18	CAN	3B	,
04	EOT	19	EM	ЗC	<
05	ENQ	1A	SUB	ЗD	=
06	ACK	1B	ESC	ЗE	>
07	BEL	1C	FS	ЗF	?
08	BS	1D	GS	40	@
09	ТАВ	1E	RS	5B	[

Hex	Char	Hex	Char	Hex	Char
OA	LF	1F	US	5C	λ
OB	VT	21	!	5D]
OC	FF	23	#	5E	٨
OD	CR	24	\$	5F	_
OE	SO	25	%	60	,
OF	SI	26	&	7B	{
10	DLE	27	,	7C	
11	DC1	28	(7D	}
12	DC2	29)	7E	~
13	DC3	2A	*	7F	DEL
14	DC4	2B	+		
15	NAK	2D	-		



Sample HIPAA Files

This appendix contains a sample HIPAA input document in Electronic Data Interchange (EDI) format, output XML document, and a 997 Functional Acknowledgment that results from inbound processing.

For more information on obtaining HIPAA sample files for testing purposes, see *Extracting HIPAA User Samples* on page 42.

In this appendix:

- Sample HIPAA Message
- Sample 997 Functional Acknowledgment
- Sample Output XML Document
- □ Sample Output Validation Report

Sample HIPAA Message

The following is a sample HIPAA document in Electronic Data Interchange (EDI) format. Shown in the next two topics are the acknowledgement and XML file that are generated from this HIPAA document on a channel configured for inbound processing.

The interchange (ISA) is 106 characters fixed length, but it has been reformatted here to fit on the page.

```
ISA*00*.....*01*SECRET....*ZZ*SUBMITTERS.ID..
*ZZ*RECEIVERS.ID...*030101*1253*^*00501*000000905*0*T*:~
GS*HR*SENDER CODE*RECEIVER CODE*19991231*0802*1*X*005010X212~
ST*276*0001*005010X212~
BHT*0010*13*ABC276XXX*20050915*1425~
HL*1**20*1~
NM1*PR*2*ABC INSURANCE*****PI*12345~
HL*2*1*21*1~
NM1*41*2*XYZ SERVICE****46*X67E~
HL*3*2*19*1~
NM1*1P*2*HOME HOSPITAL****XX*1666666668~
HL*4*3*22*0~
DMG*D8*19301210*M~
NM1*IL*1*SMITH*FRED****MI*123456789A~
TRN*1*ABCXYZ1~
REF*BLT*111~
REF*EJ*SM123456~
AMT*T3*8513.88~
DTP*472*RD8*20050831-20050906~
HL*5*3*22*0~
DMG*D8*19301115*F~
NM1*IL*1*JONES*MARY****MI*234567890A~
TRN*1*ABCXYZ2~
REF*BLT*111~
REF*EJ*J0234567~
AMT*T3*7599~
DTP*472*RD8*20050731-20050809~
HL*6*2*19*1~
NM1*1P*2*HOME HOSPITAL PHYSICIANS****XX*1666666668~
HL*7*6*22*1~
NM1*IL*1*MANN*JOHN****MI*345678901~
HL*8*7*23~
DMG*D8*19951101*M~
NM1*QC*1*MANN*JOSEPH~
TRN*1*ABCXYZ3~
REF*EJ*MA345678~
SVC*HC:99203*150****1~
DTP*472*D8*20050501~
SE*36*0001~
GE*1*1~
IEA*1*00000905~
```

Sample 997 Functional Acknowledgment

The acknowledgement service creates a functional acknowledgement (997) for inbound messages.

An acknowledgement indicates that an inbound document was received and validated for structure against the appropriate standard. An acknowledgement does not indicate that a document was processed.

The following is the acknowledgement that results from inbound processing of the preceding HIPAA document.

The interchange (ISA) is 106 characters fixed length, but it has been reformatted here to fit on the page.

```
ISA*00*.....*01*SECRET....*ZZ*RECEIVERS.ID...
*ZZ*SUBMITTERS.ID..*110223*1140*^*00501*00000905*0*T*:~
GS*FA*RECEIVER CODE*SENDER CODE*20110223*1140*1*X*005010~
ST*997*0001~
AK1*HR*1~
AK2*276*0001~
AK5*A~
AK9*A*1*1*1~
SE*6*0001~
GE*1*1~
IEA*1*000000905~
```

Sample Output XML Document

The following is the XML file that results from inbound processing of the preceding HIPAA document.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<documents>
    <ValidationReport>
        <Report>
            <Success/>
        </Report>
    </ValidationReport>
    <input>
<![CDATA[ISA*00*.....*01*SECRET....*ZZ*SUBMITTERS.ID..*ZZ*</pre>
RECEIVERS.ID...*030101*1253*^*00501*000000905*0*T*:~
        GS*HR*SENDER CODE*RECEIVER CODE*19991231*0802*1*X*005010X212~
        ST*276*0001*005010X212~
        BHT*0010*13*ABC276XXX*20050915*1425~
        HL*1**20*1~
        NM1*PR*2*ABC INSURANCE*****PI*12345~
        HL*2*1*21*1~
        NM1*41*2*XYZ SERVICE*****46*X67E~
        HL*3*2*19*1~
        NM1*1P*2*HOME HOSPITAL****XX*1666666668~
        HL*4*3*22*0~
        DMG*D8*19301210*M~
        NM1*IL*1*SMITH*FRED****MI*123456789A~
        TRN*1*ABCXYZ1~
        REF*BLT*111~
        REF*EJ*SM123456~
        AMT*T3*8513.88~
```

```
DTP*472*RD8*20050831-20050906~
        HL*5*3*22*0~
        DMG*D8*19301115*F~
        NM1*IL*1*JONES*MARY***MI*234567890A~
        TRN*1*ABCXYZ2~
        REF*BLT*111~
        REF*EJ*J0234567~
        AMT*T3*7599~
        DTP*472*RD8*20050731-20050809~
       HL*6*2*19*1~
       NM1*1P*2*HOME HOSPITAL PHYSICIANS****XX*1666666668~
       HL*7*6*22*1~
       NM1*IL*1*MANN*JOHN****MI*345678901~
       HL*8*7*23~
       DMG*D8*19951101*M~
       NM1*OC*1*MANN*JOSEPH~
        TRN*1*ABCXYZ3~
        REF*EJ*MA345678~
        SVC*HC:99203*150****1~
        DTP*472*D8*20050501~
        SE*36*0001~
       GE*1*1~
        IEA*1*00000905~]]>
    </input>
    <output>
        <HIPAA_276_005010X212>
            <TSA>
<_01_Authorization_Information_Qualifier>00
</_01_Authorization_Information_Qualifier>
<_02_Authorization_Information>.....
</_02_Authorization_Information>
<_03_Security_Information_Qualifier>01
</_03_Security_Information_Qualifier>
<_04_Security_Information>SECRET....</_04_Security_Information>
<_05_Interchange_ID_Qualifier>ZZ</_05_Interchange_ID_Qualifier>
<_06_Interchange_Sender_ID>SUBMITTERS.ID..</_06_Interchange_Sender_ID>
<_07_Interchange_ID_Qualifier>ZZ</_07_Interchange_ID_Qualifier>
<_08_Interchange_Receiver_ID>RECEIVERS.ID...
</_08_Interchange_Receiver_ID>
<_09_Interchange_Date>030101</_09_Interchange_Date>
<_10_Interchange_Time>1253</_10_Interchange_Time>
<_11_Repetition_Separator>^</_11_Repetition_Separator>
<_12_Interchange_Control_Version_Number>00501
</_12_Interchange_Control_Version_Number>
    <_13_Interchange_Control_Number>000000905
    </_13_Interchange_Control_Number>
    <_14_Acknowledgement_Requested>0</_14_Acknowledgement_Requested>
     <_15_Usage_Indicator>T</_15_Usage_Indicator>
     <_16_Component_Element_Separator>:</_16_Component_Element_Separator>
            </ISA>
            <GS>
<_01_Functional_Identifier_Code>HR</_01_Functional_Identifier_Code>
<_02_Application_Senders_Code>SENDER CODE</_02_Application_Senders_Code>
<_03_Application_Receivers_Code>RECEIVER
CODE</_03_Application_Receivers_Code>
        <_04_Date>19991231</_04_Date>
        <_05_Time>0802</_05_Time>
        <_06 Group Control Number>1</_06 Group Control Number>
       <_07_Responsible_Agency_Code>X</_07_Responsible_Agency_Information Builders
<_08_Version_Release_Industry_Identifier_Code>005010X212</_08_Version_
Release__Industry_Identifier_Code>
            </GS>
            <_276>
```

```
180
```
Sample Output Validation Report

The following is a sample output validation report in XML format containing the XML-formatted HIPAA document, resulting HIPAA formatted data, as well as the validation status.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<documents>
    <ValidationReport>
        <Report>
            <Success/>
        </Report>
    </ValidationReport>
    <input>
        <HIPAA270>
            <ISA>
<_01_Authorization_Information_Qualifier_>00</_01_Authorization_Information_Qualifier_>
<_02_Authorization_Information_>1234567890</_02_Authorization_Information_>
<_03_Security_Information_Qualifier_>00</_03_Security_Information_Qualifier_>
<_04_Security_Information_>1234567890</_04_Security_Information_>
<_05_Interchange_ID_Qualifier_>ZZ</_05_Interchange_ID_Qualifier_>
                <_06_Interchange_Sender_ID_>SUBMITTERS ID12</
_06_Interchange_Sender_ID_>
```

```
<_07_Interchange_ID_Qualifier_>ZZ</_07_Interchange_ID_Qualifier_>
                 < 08 Interchange Receiver ID >RECEIVERS ID123</
 _08_Interchange_Receiver_ID_>
                 <_09_Interchange_Date_>010122</_09_Interchange_Date_>
                 <_10_Interchange_Time_>1253</_10_Interchange_Time_>
                 <_11_Repetition_Separator_>U</_11_Repetition_Separator_>
 <_12_Interchange_Control_Version_Number_>00401</
 _12_Interchange_Control_Version_Number_>
 <_13_Interchange_Control_Number_>000000905</_13_Interchange_Control_Number_>
 <_14_Acknowledgement_Requested_>1</_14_Acknowledgement_Requested_>
                 <_15_Usage_Indicator_>T</_15_Usage_Indicator_>
 <_16_Component_Element_Separator_>:</_16_Component_Element_Separator_>
             </ISA>
             <GS>
 <_01 Functional Identifier Code >HS</_01 Functional Identifier Code >
 <_02_Application_Senders_Code_>SenderID</_02_Application_Senders_Code_>
 <_03_Application_Receivers_Code_>ReceiverID</_03_Application_Receivers_Code_>
                 <_04_Date_>20010122</_04_Date_>
                 < 05 Time >1310</ 05 Time >
                 <_06_Group_Control_Number_>1</_06_Group_Control_Number_>
 <_07_Responsible_Agency_Code_>X</_07_Responsible_Agency_Code_>
 <_08_Version__Release__Industry_Identifier_Code_>004010X092A1</_08_Version__Release_
 Industry_Identifier_Code_>
             </GS>
             < 270>
                 \langle ST \rangle
 <_01_Transaction_Set_Identifier_Code_>270</_01_Transaction_Set_Identifier_Code_>
 <_02_Transaction_Set_Control_Number_>1234</_02_Transaction_Set_Control_Number_>
                 </ST>
                 <BHT>
 <_01_Hierarchical_Structure_Code_>0022</_01_Hierarchical_Structure_Code_>
 <_02_Transaction_Set_Purpose_Code_>13</_02_Transaction_Set_Purpose_Code_>
 <_03_Reference_Identification_>10001234</_03_Reference_Identification_>
                     <_04_Date_>19990501</_04_Date_>
                     <_05_Time_>1319</_05_Time_>
                 </BHT>
                 <_2000A>
                     <HL>
 <_01_Hierarchical_ID_Number_>1</_01_Hierarchical_ID_Number_>
 <_03_Hierarchical_Level_Code_>20</_03_Hierarchical_Level_Code_>
 <_04_Hierarchical_Child_Code_>1</_04_Hierarchical_Child_Code_>
                     </HL>
                     < 2100A>
                         <NM1>
 <_01_Entity_Identifier_Code_>PR</_01_Entity_Identifier_Code_>
 <_02_Entity_Type_Qualifier_>2</_02_Entity_Type_Qualifier_>
 <_03_Name_Last_or_Organization_Name_>ABCCOMPANY</_03_Name_Last_or_Organization_Name_>
 <_08_Identification_Code_Qualifier_>PI</_08_Identification_Code_Qualifier_>
 <_09_Identification_Code_>842610001</_09_Identification_Code_>
                          </NM1>
                     </_2100A>
                     <_2000B>
                          <HL>
 <_01_Hierarchical_ID_Number_>2</_01_Hierarchical_ID_Number_>
 <_02_Hierarchical_Parent_ID_Number_>1</_02_Hierarchical_Parent_ID_Number_>
 <_03_Hierarchical_Level_Code_>21</_03_Hierarchical_Level_Code_>
 <_04_Hierarchical_Child_Code_>1</_04_Hierarchical_Child_Code_>
                         </HL>
182
                                                                         Information Builders
                          < 2100B>
                              <NM1>
 <_01_Entity_Identifier_Code_>1P</_01_Entity_Identifier_Code_>
 <_02_Entity_Type_Qualifier_>1</_02_Entity_Type_Qualifier_>
 <_03 Name_Last_or_Organization_Name_>JONES</_03 Name_Last_or_Organization_Name_>
```

< 04 Name First >MADCUE</ 04 Name First



Tutorial: Adding a Detail Line Counter to a Purchase Order Transform

This section provides a tutorial that describes how to add a detail line counter, such as a variable, to a purchase order transform. You will add a variable to the transform will count the total number of detail lines and then insert that total into the document trailer.

In this appendix:

- **Configuring the Required Variables**
- Using the Graphical Mapping Builder

Configuring the Required Variables

This section describes how to configure a variable and then add this variable to a root node.

Procedure: How to Configure a Variable

To configure a variable:

- 1. In Integration explorer, right-click the transform name and select *Properties*.
- 2. Select the variables and then click New.

3. Enter the variable Name, Value, Variable Type, and Data Type, as shown in the following image.

🖌 Properties for Copy_of_EDL_X12_to_XML.gxp 💿 💿 💌							
type filter text	Variables				⇔ • ⇔ • •		
Resource Run/Debug Settings Transform Properties @REPLACE Function	Manage the list of variables to be used in transformation mappings. Variables are used through processing functions @VARIABLE and @GetConstant. List of defined variables						
Custom Functions	Name	Value	Variable Type	Data Type	Description		
Input JDBC Data Source Mappings Output Variables XML Namespaces	detlinecnt		\$ Dynamic	number			
	New Edit Import Delete Restore Defaults Apply						
(?)					OK Cancel		

Procedure: How to Add a Variable to a Root Node

To add a variable to a root node (for example, Document):

1. Right-click the document root node, click *Add*, select *Variable*, and then click on any newly created variable to add into the Document root tag, for example *detlinecnt*.



The variable appears in the Output: XML pane, as shown in the following image.



- Move Up Output: XML 🗄 - 🚸 Document 🗄 🗇 Document - 🚸 CompanyCode С С - 🔶 CustomerNumber **C** 1 - 🚸 Direction - 🚸 DocumentType C 850 - 🔶 Footprint C PO - 🚸 Version C 1.0 🕂 🚸 Van С - 🔶 Reseller С С - 🚸 PurchaseOrderNumber - 🔶 LocationNumber С - 🔶 InternalOrderNumber С - 🔶 InternalSDQSequenceNumber С 🕂 - 📣 EDI 🕀 🌑 ED1 🗄 - 🛷 Header 🔁 🗇 Header 🕂 🚸 Name 🕀 🚸 Name 🗄 - 🚸 DetailLine 🕀 🕔 DetailLine 🗄 - 🚸 Terms 🕀 🔇 Terms ChargeOrAllowance -- \$* detlinecnt C detlinecnt
- 2. Using the up arrow on the button bar, move the newly added variable up.



The counter should be initialized to zero for each document prior to the detail line loop (you must set the counter to 0).

3. Expand the detail line group and detail group, then right-click on the group name, and add a work element that will contain the Line Count Value in the output XML.



Using the Graphical Mapping Builder

This section describes how to use the Graphical Mapping Builder to manage the mapping of the output node.

Procedure: How to Use the Graphical Mapping Builder

1. Double-click the *Work* element to open Graphical Mapping Builder.



2. Drag the *\$detlinecnt* variable from the Variables pane and drop it in Graphical Mapping Builder workspace, as shown in the following image.

🔬 Mapping Builder		
Graphical Mapping Builder		
Manage the mapping of the out choose constant or expression.	tput node. Mapping can t For Insert/Replace menu,	be done by dragging the function or the input into th drag and drop a function while holding the right mo
Function	*	
Input	×	
Variables	*	C <empty></empty>
\$ detlinecnt		
Mapping Builder		
Graphical Mapping Builder		
Manage the mapping of the outp choose constant or expression. Fo	out node. Mapping can be or Insert/Replace menu, dr	done by dragging the function or the input into tl ag and drop a function while holding the right mc
Function	*	
Input	*	
Variables	*	& @VARIABLE
\$ detlinecnt		dynamic_variable_name action

3. Double-click the *GET* action support box, select *ADD* from drop-down list, and click *OK*, as shown in the following image.

🔏 Mapping Builder						
Graphical Mapping Builder						
Manage the mapping of the output node. Mapping can be done by dragging the function or the input into the mapping area or by right clicking ar choose constant or expression. For Insert/Replace menu, drag and drop a function while holding the right mouse button.						
Function ¥						
Input ¥						
Variables *	& @VARIABLE					
S.detlineont:	dynamic_variable_name action					
test variable	C detlinecnt C GET C Orstant Expression C GET GET SET SET SET SET SET SET SET S					

The updated variable appears.

🔏 Mapping Builder						
Graphical Mapping Builder						
Manage the mapping of the output node. Mapping can be done by dragging the function or the input into the mapp choose constant or expression. For Insert/Replace menu, drag and drop a function while holding the right mouse but						
Function	*					
Input	*					
Variables	*	★ @VARIABLE				
\$ detlinecnt		dynamic_variable_name action value				
		C detlinecnt C ADD C 1				



The following screen appears in the transform.

The TotalOrder group already contains the element, *detaillinecount*, to contain the counter, as shown in the following image.



4. Double-click the *detaillinecount* element to open the Graphical Mapping Builder.

5. Drag the *\$detlinecnt* variable from the Variables pane and drop it in the Graphical Mapping Builder workspace.



- 6. Click OK and then save your transform.
- 7. Test run your transform.

The following example shows 3 as the total number of detail lines appearing in the node.

```
</PriceCost>
<work>3</work>
</Detail>
</DetailLine>
<TotalOrder>
</TotalOrder>
</TotalOrder>
</Document>
```

Feedback

Customer success is our top priority. Connect with us today!

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also preview new innovations to get an early look at new content products and services. Your participation helps us create great experiences for every customer.

To send us feedback or make a connection, contact Sarah Buccellato, Technical Editor, Technical Content Management at Sarah_Buccellato@ibi.com.

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at *Frances_Gambino@ibi.com*.

Inf%rmation Builders

iWay

iWay Integration Solution for HIPAA User's Guide

Version 7.0.x and Higher

DN3502267.0418

Information Builders, Inc. Two Penn Plaza New York, NY 10121-2898

