# Preface

- Documentation Conventions .............................................................. 8
- Related Publications ......................................................................... 9
- Customer Support ........................................................................... 9
- Help Us to Serve You Better ............................................................. 10
- User Feedback ................................................................................. 12
- Information Builders Consulting and Training ...................................... 13

## 1. Introducing the iWay Integration Solution for HIPAA

- The HIPAA Mandate ....................................................................... 15
  - Achieving Administrative Simplification ........................................ 16
- Promoting HIPAA Compliance and Integration .................................... 17
  - Legacy Integration ....................................................................... 17
- Features of the iWay Integration Solution for HIPAA .......................... 17
- HIPAA Information Roadmap ............................................................ 18

## 2. Deployment Information for Your iWay Integration Solution

- iWay Products .................................................................................. 21
  - iWay Service Manager. ................................................................. 21
  - iWay Transformer ......................................................................... 22
    - Understanding Pipelined Transformations. ................................. 22
      - Output Node Name. ................................................................. 22
  - iIT Designer ............................................................................... 23
  - Using a Channel to Construct a HIPAA Message Flow .................... 23
    - Components of a Channel. ........................................................ 24
- Components of the iWay Integration Solution for HIPAA .................. 26
  - Ebix ............................................................................................ 27
  - Preparsers .................................................................................... 27
    - HipaaSplitterPreParser ............................................................ 28
    - HIPAABatchSplitter ................................................................. 28
  - Acknowledgement Service ............................................................ 28
  - Deidentification Service ............................................................... 29
  - Preemitter ................................................................................... 29
3. Configuring the EDI Activity Driver ........................................ 31

   HIPAA EDI Activity Driver Overview ........................................... 31
   Configuring the EDI Data Provider Using iWay Service Manager ........... 31
   Configuring the EDI Activity Driver Using iWay Service Manager .......... 34

4. Working With HIPAA Inbound and Outbound Applications Using iWay Integration Tools (iIT) .................................................. 41

   HIPAA Inbound and Outbound Applications Overview ....................... 41
   HIPAA Inbound and Outbound Applications Prerequisites .................... 42
   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 .... 51
   Deploying iWay Integration Applications to iWay Service Manager .......... 54
   Setting Registers in the iWay Service Manager Administration Console .... 57
   Stopping Inbound (HIPAA to XML) and Outbound (XML to HIPAA) Processing .... 60
   Testing the Sample HIPAA Applications .................................... 62

5. Inbound Processing: HIPAA to XML ........................................... 67

   HIPAA Inbound Processing Overview .......................................... 67
   Sample Configuration for Inbound Processing: HIPAA to XML ............... 69
      Accessing the iWay Service Manager Administration Console ............ 69
      Adding an Ebix to the Registry ............................................. 70
      Adding Special Register Sets .............................................. 73
      Defining an Inlet ........................................................... 74
      Defining a Route ........................................................... 83
      Defining the Outlets ....................................................... 102
      Defining a Channel ....................................................... 110
      Reusing Your Channel Configuration .................................... 114

6. Outbound Processing: XML to HIPAA ........................................ 115

   HIPAA Outbound Processing Overview .................................... 115
   Sample Configuration for Outbound Processing: XML to HIPAA .......... 116
      Accessing the iWay Service Manager Administration Console ............ 117
      Adding an Ebix to the Registry ............................................. 117
Preface

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:

<table>
<thead>
<tr>
<th>Chapter/Appendix</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introducing the iWay Integration Solution for HIPAA</td>
<td>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.</td>
</tr>
<tr>
<td>2 Deployment Information for Your iWay Integration Solution</td>
<td>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.</td>
</tr>
<tr>
<td>3 Configuring the EDI Activity Driver</td>
<td>Describes how to configure the EDI Activity Driver using iWay Service Manager.</td>
</tr>
<tr>
<td>4 Working With HIPAA Inbound and Outbound Applications Using iWay Integration Tools (iIT)</td>
<td>Describes how to work with HIPAA inbound and outbound applications using iWay Integration Tools (iIT).</td>
</tr>
<tr>
<td>Chapter/Appendix</td>
<td>Contents</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>5 Inbound Processing: HIPAA to XML</td>
<td>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.</td>
</tr>
<tr>
<td>6 Outbound Processing: XML to HIPAA</td>
<td>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.</td>
</tr>
<tr>
<td>A Ebix-Supported Transaction Sets</td>
<td>Describes the HIPAA transaction sets supported by the iWay Integration Solution for HIPAA in the Ebix files supplied with the product.</td>
</tr>
<tr>
<td>B Using iWay Integration Tools to Configure an Ebix for HIPAA</td>
<td>Describes how to use iWay Integration Tools (iIT) to configure an e-Business Information Exchange (Ebix) file for HIPAA.</td>
</tr>
<tr>
<td>C Using HIPAA Special Register (SREG) Types</td>
<td>Describes the Special Register (SREG) types that are created during HIPAA to XML transactions and 997/999 creation.</td>
</tr>
<tr>
<td>D Using HIPAA Separators and Terminators</td>
<td>Includes a list of separators and terminators that are allowed.</td>
</tr>
<tr>
<td>E Sample HIPAA Files</td>
<td>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.</td>
</tr>
<tr>
<td>F Tutorial: Adding a Detail Line Counter to a Purchase Order Transform</td>
<td>Provides a tutorial that describes how to add a detail line counter, such as a variable, to a purchase order transform.</td>
</tr>
</tbody>
</table>

**Documentation Conventions**

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

<table>
<thead>
<tr>
<th>This Typeface or this typeface</th>
<th>Denotes syntax that you must enter exactly as shown.</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>this typeface</em></td>
<td>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.</td>
</tr>
<tr>
<td><em>underscore</em></td>
<td>Indicates a default setting.</td>
</tr>
<tr>
<td>Key + Key</td>
<td>Indicates keys that you must press simultaneously.</td>
</tr>
<tr>
<td><code>{ }</code></td>
<td>Indicates two or three choices. Type one of them, not the braces.</td>
</tr>
<tr>
<td>`</td>
<td>`</td>
</tr>
<tr>
<td>...</td>
<td>Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).</td>
</tr>
<tr>
<td>.</td>
<td>Indicates that there are (or could be) intervening or additional commands.</td>
</tr>
</tbody>
</table>

### Related Publications

Visit our Technical Documentation Library at [http://documentation.informationbuilders.com](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](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](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](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.

<table>
<thead>
<tr>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
</tr>
<tr>
<td>OS Version</td>
</tr>
<tr>
<td>JVM Vendor</td>
</tr>
<tr>
<td>JVM Version</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Version</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enterprise Information System (EIS) - if any</strong></td>
<td></td>
</tr>
<tr>
<td><strong>EIS Release Level</strong></td>
<td></td>
</tr>
<tr>
<td><strong>EIS Service Pack</strong></td>
<td></td>
</tr>
<tr>
<td><strong>EIS Platform</strong></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Request/Question</th>
<th>Error/Problem Details or Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Did the problem arise through a service or event?</td>
<td></td>
</tr>
<tr>
<td>Provide usage scenarios or summarize the application that produces the problem.</td>
<td></td>
</tr>
<tr>
<td>When did the problem start?</td>
<td></td>
</tr>
<tr>
<td>Can you reproduce this problem consistently?</td>
<td></td>
</tr>
<tr>
<td>Describe the problem.</td>
<td></td>
</tr>
<tr>
<td>Describe the steps to reproduce the problem.</td>
<td></td>
</tr>
<tr>
<td>Specify the error message(s).</td>
<td></td>
</tr>
</tbody>
</table>
### 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](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.
Chapter 1

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 back-end 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.

<table>
<thead>
<tr>
<th>Product</th>
<th>For more information, see...</th>
</tr>
</thead>
<tbody>
<tr>
<td>iWay Service Manager</td>
<td>Chapters 3, 4, 5, and 6 of this guide</td>
</tr>
<tr>
<td></td>
<td><em>iWay Service Manager User’s Guide</em></td>
</tr>
<tr>
<td>Product</td>
<td>For more information, see...</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>iWay Transformer</td>
<td><em>iWay Transformer User’s Guide</em></td>
</tr>
<tr>
<td>iWay Integration Tools (iIT) Designer</td>
<td><em>iWay Integration Tools Designer User’s Guide</em></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Format</th>
<th>Non-XML &gt; XML Pipeline</th>
<th>XML &gt; Non-XML Pipeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIPAA</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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.

**iIT 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.
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.
Components of the iWay Integration Solution for HIPAA

- 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_Vsersion.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.
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.
Chapter 3

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*.
The Data Provider pane opens.

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.
The configuration pane for the JDBC data provider opens.

<table>
<thead>
<tr>
<th>Data Provider: JDBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listed below is the definition of the selected JDBC data provider. Add/Update the values as required.</td>
</tr>
</tbody>
</table>

**JDBC Connection Pool Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Enter the name of the JDBC data provider to add.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>EDI_Activity_DB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Driver Class</th>
<th>The JDBC driver class is the name of the class that contains the code for this JDBC Driver.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>com.mysql.jdbc.Driver</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection URL</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>jdbc:mysql://localhost:3306/IWay</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>User name with respect to the JDBC URL and driver.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>iWay</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Password</th>
<th>Password with respect to the JDBC URL and driver.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>****</td>
</tr>
</tbody>
</table>

**Connection Pool Properties**

<table>
<thead>
<tr>
<th>Initial Pool Size</th>
<th>Number of connections to place in the pool at startup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximum Number of Idle Connections</th>
<th>Maximum number of idle connections to retain in the pool. 0 means no limit except what is enforced by the maximum number of connections in the pool.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximum Number of Connections</th>
<th>Maximum number of connections in the pool. 0 means no limit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Login Timeout</th>
<th>Time in seconds to wait for a pooled connection before throwing an exception. 0 means wait forever.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1</td>
</tr>
</tbody>
</table>

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.

![Image of JDBC Connections list]

**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.

   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.

   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 Parameters](image)

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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNDI Factory Name</td>
<td>String</td>
<td>The JNDI initial context factory class that is used to access a data source. Use <code>com.ibi.jndi.XDInitialContextFactory</code> for an iWay JDBC provider or leave this field blank for the JVM default.</td>
</tr>
<tr>
<td>JNDI Name</td>
<td>String</td>
<td>The JNDI name for the data source this driver will use. To use an iWay JDBC provider, enter the JNDI name as <code>jdbc/&lt;data provider name&gt;</code>, where <code>data provider name</code> is the name of the EDI Activity Driver that was specified in step 4. Otherwise the information for the defined provider will be used.</td>
</tr>
<tr>
<td>Table</td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Parameter Name</strong></td>
<td><strong>Type</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Compression</td>
<td>Drop-down list</td>
<td>Specify whether the messages are to be compressed. Values include:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>□ none (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>□ smallest</td>
</tr>
<tr>
<td></td>
<td></td>
<td>□ fastest</td>
</tr>
<tr>
<td></td>
<td></td>
<td>□ standard</td>
</tr>
<tr>
<td></td>
<td></td>
<td>□ Huffman</td>
</tr>
<tr>
<td>Start Events</td>
<td>Boolean</td>
<td>If set to <em>true</em> (default), the input messages will be recorded in the activity log. This value must be set to <em>true</em> for use of the audit reports in the console.</td>
</tr>
<tr>
<td>Internal Events</td>
<td>Boolean</td>
<td>If set to <em>true</em> (default), system events are included in the activity log. System events include activities such as parsing and transformations (optional). False is selected by default.</td>
</tr>
<tr>
<td>Security Events</td>
<td>Boolean</td>
<td>If set to <em>true</em> (default), security events are recorded. This includes digital signature, and so on. However, console activity is not recorded.</td>
</tr>
<tr>
<td>Business Error Events</td>
<td>Boolean</td>
<td>If set to <em>true</em> (default), business errors are recorded, such as rules system violations. False is selected by default.</td>
</tr>
<tr>
<td>Emit Events</td>
<td>Boolean</td>
<td>If set to <em>true</em> (default), output messages from emitter services will be recorded. This is required for use of the audit log reports in the console.</td>
</tr>
</tbody>
</table>
End Events: If set to true (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: 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: The Message Authentication Code (MAC) algorithm. None (default) indicates a MAC should not be computed.

MAC Provider: The Message Authentication Code (MAC) provider. Not Specified indicates the default provider should be used. The remaining available value is SunJCE.

MAC Secret Key: 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.

<table>
<thead>
<tr>
<th>Database Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordkey</td>
<td>Unique record identifier.</td>
</tr>
<tr>
<td>Database Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>recordtype</td>
<td>Type of this record - the event being recorded.</td>
</tr>
<tr>
<td></td>
<td>- 101 - Message start.</td>
</tr>
<tr>
<td></td>
<td>- 131 - Entry to event (see subtype codes below).</td>
</tr>
<tr>
<td></td>
<td>- 132 - Normal exit from event.</td>
</tr>
<tr>
<td></td>
<td>- 133 - Failed exit from event.</td>
</tr>
<tr>
<td></td>
<td>- 151 - Ancillary message (usually rules violation).</td>
</tr>
<tr>
<td></td>
<td>- 181 - Emit.</td>
</tr>
<tr>
<td></td>
<td>- 191 - Message end.</td>
</tr>
<tr>
<td>signature</td>
<td>Encoding of the listener name and protocol.</td>
</tr>
<tr>
<td>protocol</td>
<td>Name of the protocol.</td>
</tr>
<tr>
<td>address</td>
<td>Address to which an emit is to be issued. The format depends on the protocol.</td>
</tr>
<tr>
<td>tstamp</td>
<td>Timestamp of record.</td>
</tr>
<tr>
<td>correlid</td>
<td>ISA13</td>
</tr>
<tr>
<td>tid</td>
<td>Transaction ID assigned to this message.</td>
</tr>
<tr>
<td>msg</td>
<td>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.</td>
</tr>
<tr>
<td>context</td>
<td>Serialized special registers that were in the context at the time the record was written.</td>
</tr>
<tr>
<td>text</td>
<td>Message text for business errors (rules system violations).</td>
</tr>
<tr>
<td>Database Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>status</td>
<td>Status code recorded.</td>
</tr>
<tr>
<td></td>
<td>0 - Success</td>
</tr>
<tr>
<td></td>
<td>1 - Success, message end (191 record)</td>
</tr>
<tr>
<td></td>
<td>10 - Rules error</td>
</tr>
<tr>
<td>subtype</td>
<td>Event code for event records.</td>
</tr>
<tr>
<td></td>
<td>1 - Preparser</td>
</tr>
<tr>
<td></td>
<td>2 - Parser</td>
</tr>
<tr>
<td></td>
<td>3 - In reviewer</td>
</tr>
<tr>
<td></td>
<td>5 - In validation</td>
</tr>
<tr>
<td></td>
<td>6 - In transform</td>
</tr>
<tr>
<td></td>
<td>7 - Agent or flow</td>
</tr>
<tr>
<td></td>
<td>8 - Out transform</td>
</tr>
<tr>
<td></td>
<td>9 - Out validation</td>
</tr>
<tr>
<td></td>
<td>11 - Preemitter</td>
</tr>
<tr>
<td></td>
<td>1000 - input record written to table before transformation</td>
</tr>
<tr>
<td>partner_to</td>
<td>ISA06</td>
</tr>
<tr>
<td>partner_from</td>
<td>ISA08</td>
</tr>
<tr>
<td>encoding</td>
<td>Encoding of the listener that obtained the document.</td>
</tr>
<tr>
<td>mac</td>
<td>Not used in this version.</td>
</tr>
<tr>
<td>Driver version</td>
<td>1.0 in 8.0 SM</td>
</tr>
</tbody>
</table>
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
- 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](http://techsupport.informationbuilders.com)

   The downloaded HIPAA_usr_samples.zip contains the following files:

   - HIPAA_Accelerator.zip
   - Hipaa_usr_samples_iIT_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.

![File Structure Image]

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:

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:

![Image of file explorer windows with XML files]

**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.

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.

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_iIT_workspace.zip file and click Open.
You are returned to the Import Projects pane, as shown in the following image.

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.
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.

3. Click the Deploy icon next to the application name under the Actions column, as shown in the following image.
The Deployments pane opens, as shown in the following image.

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.

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.

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Actions</th>
<th>State</th>
<th>Since</th>
<th>Application</th>
<th>Template</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hipaa_usr_samples_App</td>
<td>✗ ✗</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

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

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 application from the Management drop-down list. Click Server in the top menu and then Register Settings in the left pane.

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

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

6. Add ValidateHIPAA and provide the appropriate values in the fields, as shown in the following image. Click Finish.
The following image shows the summary of defined Registers.

![Register Summary](image)

**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.

![Stopping Inbound Processing](image)
The inbound application channel will be stopped, as shown in the following image.

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

Click the State icon adjacent to the outbound application channel (XmlToHipaa_QS_Channel) under Management\Monitoring and click OK, as shown in the following image.
The outbound application channel will be stopped, as shown in the following image.

### 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:

   ![Image of file structure]

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

   \Hipaa_Accelerator\HIPAA_in\IB_Output
3. Observe the Reports in the following directory:
   **Hipaa_Accelerator\HIPAA_in\IB_Report**
   For example:

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:
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

**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:

\Hipaa_Accelerator\HIPAA_out

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

\Hipaa_Accelerator\HIPAA_out\OB_Output
For example:

3. Observe the Reports in the following directory:
   Hipaa_Accelerator\HIPAA_out\OB_Report

   For example:

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:

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:
Chapter 5

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**

- 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.

---

**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.

2. Under Components in the left pane of the Registry, select Ebix.
The Ebix pane opens, as shown in the following image.

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

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.

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ack</td>
<td>string</td>
<td>C:\file_out\hipaa\ack</td>
<td>Output directory for 997</td>
</tr>
<tr>
<td>Ack999</td>
<td>string</td>
<td>C:\file_out\hipaa\ack</td>
<td>Output directory for 999</td>
</tr>
<tr>
<td>Archive</td>
<td>string</td>
<td>C:\file_out</td>
<td>Archive of transformed X12 files</td>
</tr>
<tr>
<td>BadOutput</td>
<td>string</td>
<td>c:\file_out\hipaa\bad</td>
<td>XML where ask status is not equal to A(accept)</td>
</tr>
<tr>
<td>Error</td>
<td>string</td>
<td>C:\file_out</td>
<td></td>
</tr>
<tr>
<td>GoodOutput</td>
<td>string</td>
<td>c:\file_out\hipaa\good</td>
<td>XML where ask status equal to A (accept)</td>
</tr>
<tr>
<td>Input</td>
<td>string</td>
<td>c:\file_in\hipaa</td>
<td>X12 inbound file scan this directory for EDI</td>
</tr>
</tbody>
</table>

**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**

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

- **Listener.** A listener is a component that picks up input on a channel from a configured endpoint.
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Path</td>
<td>Directory in which input messages are received. A specific file name or DOS-style pattern can be used. Do not use file suffix.</td>
</tr>
<tr>
<td>Destination</td>
<td>Directory into which output files are stored. Specific file name is optional. Use * in file name to be replaced by timestamp, # by sequential counter</td>
</tr>
<tr>
<td>Removal Destination</td>
<td>Full path file pattern asserting where input files will be moved. Use * in file name to be replaced by timestamp, # by sequential counter</td>
</tr>
<tr>
<td>Suffix In</td>
<td>Limits input files to those with these extensions. Ex: xsl, ini. Do not use ‘<em>’ as mean no extension, ‘</em>’ means any</td>
</tr>
<tr>
<td>Scan subdirectories</td>
<td>If true, all subdirectories will be scanned for files to process</td>
</tr>
<tr>
<td>Do not unzip ZIP files</td>
<td>Pass ZIP files as a single file for processing (requires ACCEPT_FLAT turned on)</td>
</tr>
<tr>
<td>Suffix Out</td>
<td>Extension for output files (name is same as input file unless specified in destination parameters)</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Listeners</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

**Select listener type**

<table>
<thead>
<tr>
<th>Name *</th>
<th>Name of the new listener</th>
</tr>
</thead>
<tbody>
<tr>
<td>HipaaToXML_Ebix</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Description for the new listener</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hipaa To XML file listener</td>
<td></td>
</tr>
</tbody>
</table>

6. 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.

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:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template</td>
<td>Used to locate the template in the Ebix used in the transformation from HIPAA format to XML format.</td>
</tr>
<tr>
<td>Debug</td>
<td>If enabled, the transformation components are written to files in the local directory. This parameter is set to False by default.</td>
</tr>
<tr>
<td>Segment Terminator</td>
<td>The control character that marks the end of a specific variable-length segment.</td>
</tr>
<tr>
<td></td>
<td>To view a list of segment terminator characters, see <em>Using HIPAA Separators and Terminators</em> on page 175.</td>
</tr>
<tr>
<td>Element Delimiter</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>To view a list of element delimiter characters, see <em>Using HIPAA Separators and Terminators</em> on page 175.</td>
</tr>
<tr>
<td>Component Element Delimiter</td>
<td>The control character used to separate sub-elements/components in a composite element.</td>
</tr>
<tr>
<td></td>
<td>To view a list of component element delimiter characters, see <em>Using HIPAA Separators and Terminators</em> on page 175.</td>
</tr>
<tr>
<td>Escape Character</td>
<td>The escape character is necessary if any of the HIPAA document separators is part of the actual value of an attribute.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 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 Log Settings 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 false. By default, this parameter is set to true. |
Parameter | Description
--- | ---
Node 'delimiters' | If set to true, node delimiters are added to the generated XML document. For example:

```xml
<?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>
```

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.

<table>
<thead>
<tr>
<th>Preparers</th>
</tr>
</thead>
<tbody>
<tr>
<td>A logical process that handles documents before they are parsed by the system. Usually used to convert from non-XML to XML.</td>
</tr>
</tbody>
</table>

### Provide a name and description for the new Preparser object definition

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>Name of the new Preparser object definition</td>
</tr>
<tr>
<td></td>
<td>HipaaSplitter</td>
</tr>
<tr>
<td>Description</td>
<td>Description for the new Preparser object definition</td>
</tr>
<tr>
<td></td>
<td>Hipaa Splitter Preparser</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>HipaaToXML_Ebix</td>
</tr>
<tr>
<td>Description</td>
<td>The file inlet contains the file listener and edi splitter preparser for Hipaa to XML processing</td>
</tr>
</tbody>
</table>

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

8. Select **Preparser** and click **Next**.

On the next pane, you are prompted to select a 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.
The New Process Flow Configuration Wizard opens, as shown in the following image.

5. 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 iT 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.

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.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Protocol on which to make acknowledgment copies. Select one of the following options from the drop-down list:</td>
</tr>
<tr>
<td></td>
<td>- NONE</td>
</tr>
<tr>
<td></td>
<td>- FILE</td>
</tr>
<tr>
<td>Location</td>
<td>Location for acknowledgment copies.</td>
</tr>
<tr>
<td>End Tag</td>
<td>The surrounding XML tag.</td>
</tr>
<tr>
<td>Preemitter</td>
<td>Determines whether the preemitter should be run on acknowledgment output.</td>
</tr>
<tr>
<td>Error</td>
<td>Determines whether to send an error.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ISA Control Number</td>
<td>Element location of ISA control number. Select one of the following locations from the drop-down list:</td>
</tr>
<tr>
<td></td>
<td>- Input Document</td>
</tr>
<tr>
<td></td>
<td>- _SReg(edi.ICN)</td>
</tr>
<tr>
<td>GS Control Number</td>
<td>Element location of GS control number. Select one of the following locations from the drop-down list:</td>
</tr>
<tr>
<td></td>
<td>- Input Document</td>
</tr>
<tr>
<td></td>
<td>- _SReg(edi.GCN)</td>
</tr>
<tr>
<td>ST Control Number</td>
<td>Element location of ST control number. Select one of the following locations from the drop-down list:</td>
</tr>
<tr>
<td></td>
<td>- Input Document</td>
</tr>
<tr>
<td></td>
<td>- _SReg(edi.MCN)</td>
</tr>
<tr>
<td>Stream Acknowledgment</td>
<td>Determines the level of acknowledgment information to return. Select one of the following acknowledgment levels from the drop-down list:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Group.</strong> Returns acknowledgment information at the Group level.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Interchange.</strong> Returns acknowledgment information at the Interchange level.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Transaction.</strong> Returns acknowledgment information at the Transaction level.</td>
</tr>
</tbody>
</table>
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.

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.

33. 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.

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.
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 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.

46. 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.

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 Dialog Box]

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

<table>
<thead>
<tr>
<th>Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>HIPAA Good</td>
</tr>
<tr>
<td>R</td>
<td>HIPAA Reject</td>
</tr>
<tr>
<td>E</td>
<td>HIPAA Error</td>
</tr>
<tr>
<td>P</td>
<td>HIPAA Partial</td>
</tr>
</tbody>
</table>

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:
   \texttt{sreg(HIPAA.BadOutput)}

71. For the File Pattern parameter, enter the following:
   \texttt{sreg(basename)_*.xml}

72. For the Respect Transactionality parameter, select \textit{true} from the drop-down list.

73. Accept the default values for the remaining parameters.

74. Click \textit{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 \textit{OnCustom}.

77. In the Case of section, select cases \textit{R (HIPAA Reject)}, \textit{E (HIPAA Error)}, and \textit{P (HIPAA Partial)}.

78. Click \textit{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 drop-down 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.

82. 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.

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>HIPAAtoXML_Route</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>
4. Click Finish.
5. On the Construct Route pane, click Add.
   You are prompted for the type of component to associate with the route.
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.

<table>
<thead>
<tr>
<th>Configuration parameters for new emitter of type File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination *</td>
</tr>
<tr>
<td>path to file,* replaced with timestamp</td>
</tr>
<tr>
<td>sreg(hipaa.GoodOutput)/SREG(basename)_*.xml</td>
</tr>
<tr>
<td>Create Directory</td>
</tr>
<tr>
<td>Create directory if it doesn’t exist</td>
</tr>
<tr>
<td>false</td>
</tr>
<tr>
<td>Pick one</td>
</tr>
</tbody>
</table>

4. Supply configuration parameters for the new File emitter as follows, then click Next.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination *</td>
<td>sreg(Hipaa.GoodOutput)/SREG(basename)_*.xml</td>
</tr>
<tr>
<td></td>
<td>This value is the directory where the valid XML output is placed.</td>
</tr>
<tr>
<td></td>
<td>sreg(Hipaa.GoodOutput) is a special register value that uses a defined directory in which valid output files are stored after transformation.</td>
</tr>
<tr>
<td></td>
<td>Make sure that you have created this directory; otherwise, errors will occur during deployment.</td>
</tr>
<tr>
<td></td>
<td>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 iWay Service Manager User’s Guide.</td>
</tr>
<tr>
<td></td>
<td>On output, an asterisk (*) in the destination file name is replaced by a date and time stamp.</td>
</tr>
<tr>
<td></td>
<td>Since you are using the HipaaSplitterPreParser, include an asterisk in the file name to create each XML output file with a unique name.</td>
</tr>
<tr>
<td>Create Directory</td>
<td>false</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>HipaaFileOut</td>
</tr>
<tr>
<td>Description</td>
<td>Hipaa file output</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Configuration parameters for new emitter of type File</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Destination</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Create Directory</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Destination</strong> *</td>
<td>sreg(Hipaa.Ack)/SREG(basename)*.txt</td>
</tr>
<tr>
<td></td>
<td>This value is the directory where the acknowledgement output is placed. You can use an extension other than .txt, for example, .edi or .data.</td>
</tr>
<tr>
<td></td>
<td>sreg(Hipaa.Ack) is a special register value that uses a defined directory in which output files are stored after transformation.</td>
</tr>
<tr>
<td></td>
<td>Make sure that you have created this directory; otherwise, errors will occur during deployment.</td>
</tr>
<tr>
<td></td>
<td>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 iWay Service Manager User’s Guide.</td>
</tr>
<tr>
<td><strong>Create Directory</strong></td>
<td>false</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong> *</td>
<td>HipaaAckOut</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Hipaa Acknowledgement Out</td>
</tr>
</tbody>
</table>
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.

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

4. Supply configuration parameters for the new File emitter as follows, then click Next.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination *</td>
<td>sreg(Hipaa.BadOutput)/SREG(basename)_*.xml</td>
<td>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 iWay Service Manager User's Guide. 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.</td>
</tr>
<tr>
<td>Create Directory</td>
<td>false</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>HipaaBadFileOut</td>
</tr>
<tr>
<td>Description</td>
<td>Hipaa Bad Output File</td>
</tr>
</tbody>
</table>
**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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>HipaaAckOut</td>
</tr>
<tr>
<td>Description</td>
<td>Outlet containing Hipaa Acknowledgement to be sent to your trading partner</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Component Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emitter</td>
<td>Emitters are protocol handlers, that drive the output of a channel to a configured endpoint.</td>
</tr>
<tr>
<td>Preemitter</td>
<td>A logical process that handles documents immediately prior to transmission. Usually this converts from XML to non-XML.</td>
</tr>
<tr>
<td>Encryptor</td>
<td>Encrypts the document</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>HipaaBadFileOut</td>
</tr>
<tr>
<td>Description</td>
<td>Outlet containing converted Hipaa data that failed rules validation</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>HipaaFileOut</td>
</tr>
<tr>
<td>Description</td>
<td>Outlet containing converted Hipaa data into XML format</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>HipaaToXML_Ebix</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

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.

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.

<table>
<thead>
<tr>
<th>Outlet</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>HipaaAckOut</td>
<td><code>_isflat()</code></td>
</tr>
<tr>
<td>HipaaBadFileOut</td>
<td><code>_isXML()</code> AND COND(SREG(edi.ackstatus),NE,'A')</td>
</tr>
</tbody>
</table>
Sample Configuration for Inbound Processing: HIPAA to XML

### 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/Emittors).

   **Sample Configuration for Inbound Processing: HIPAA to XML**

   ```
   HipaaToXML_Ebix
   Build result for channel
<table>
<thead>
<tr>
<th>Message level</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>Start</td>
</tr>
<tr>
<td>Info</td>
<td>Validating Channel</td>
</tr>
<tr>
<td>Info</td>
<td>Channel is valid</td>
</tr>
<tr>
<td>Info</td>
<td>Validating Inlet</td>
</tr>
<tr>
<td>Info</td>
<td>Inlet is valid</td>
</tr>
<tr>
<td>Info</td>
<td>Validating Routes</td>
</tr>
<tr>
<td>Info</td>
<td>Routes are valid</td>
</tr>
<tr>
<td>Info</td>
<td>Validating Outlets</td>
</tr>
<tr>
<td>Info</td>
<td>Outlets are valid</td>
</tr>
<tr>
<td>Info</td>
<td>Build Successful</td>
</tr>
<tr>
<td>Info</td>
<td>End</td>
</tr>
<tr>
<td>Info</td>
<td>Channel archiving C:\PROGRA~1:iWay Service Manager\repository\channel\HipaaToXML_Ebix\HipaaToXML_Ebix_2:HipaaToXML_Ebix has been created/updated</td>
</tr>
</tbody>
</table>
   ```

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*.

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.

**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*. 

---

Sample Configuration for Inbound Processing: HIPAA to XML

114  Information Builders
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 XML-formatted 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 XML-formatted 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

- 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 Manager](image)

2. Under Components in the left pane of the Registry, select Ebix.
3. Click Add to add a new Ebix.
   The New Ebix pane opens.

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.

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive</td>
<td>string</td>
<td>C:\file_out\hipaa\archived</td>
<td>Archive of transformed XML files</td>
</tr>
<tr>
<td>Error</td>
<td>string</td>
<td>C:\file_out\hipaa\error</td>
<td>HIPAA errors</td>
</tr>
<tr>
<td>Input</td>
<td>string</td>
<td>C:\file_in\hipaa\input</td>
<td>Input outbound flow</td>
</tr>
<tr>
<td>Output</td>
<td>string</td>
<td>C:\file_out\hipaa\output</td>
<td>Output outbound flow</td>
</tr>
<tr>
<td>ValidationReport</td>
<td>string</td>
<td>C:\file_out\hipaa\report</td>
<td>All validation reports</td>
</tr>
</tbody>
</table>

**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.

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

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>XmlToHipaa_Ebix</td>
</tr>
<tr>
<td>Description</td>
<td>XML to Hipaa file listener</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>XmlToHipaa_Ebix</td>
</tr>
<tr>
<td>Description</td>
<td>The file inlet contains a file listener for XML to HIPAA processing.</td>
</tr>
</tbody>
</table>

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 *XmlToHipaa_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.

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.

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.
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.
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.

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.

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

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

\[ \text{XPATH(}/\text{documents/output}) \]

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

\[ \text{sreg(Hipaa/XML/Output)} \]

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

\[ \text{sreg(basename)}_*\text{x12} \]

58. For the Return parameter, select \textit{input} from the drop-down list.

59. Click \textit{Finish}.

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

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

The Line Configuration dialog box opens.
61. From the Event drop-down list, select "OnCustom".

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.

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.

76. To save the process flow, right-click the XMLToHipaa_Ebix 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.

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>XmlToHipaaEbix</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

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

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.

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>XmltoHipaaEbix/XML_error</td>
</tr>
<tr>
<td>Description</td>
<td>XmltoHipaa Ebix/XML error</td>
</tr>
</tbody>
</table>

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.

**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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination</td>
<td>path to file, * replaces timestamp</td>
</tr>
<tr>
<td></td>
<td>[sreg(Hipaa_XML.ValidationReport)\validation_sreg(basename)_*.xml]</td>
</tr>
<tr>
<td>Create Directory</td>
<td>Create directory if it doesn’t exist</td>
</tr>
<tr>
<td></td>
<td>[true]</td>
</tr>
</tbody>
</table>

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.

---

**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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td><code>XmltoHipaaEbix_XML_validation</code></td>
</tr>
<tr>
<td>Description</td>
<td>Outlet which will contain valid validation report produced by the validation process.</td>
</tr>
</tbody>
</table>

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, `XmlToHipaa_Ebx`) 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.

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.
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.

<table>
<thead>
<tr>
<th>Outlet</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlToHipaaEbix_XML_validation</td>
<td>_isxml() and sreg(iwaf.validationSuccess) = true</td>
</tr>
<tr>
<td>XmlToHipaaEbix_XML_error</td>
<td>_isxml() and sreg(iwaf.validationSuccess) != true</td>
</tr>
</tbody>
</table>

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.
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.

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

The Channel Management pane reopens.
4. Select *XmlToHipaa_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.

**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*. 
### Supported HIPAA Transactions

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

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

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

<table>
<thead>
<tr>
<th>Version</th>
<th>Document</th>
<th>Version</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>5010X217</td>
<td>278Aut, 278Can, 278Req, 278Res</td>
<td>5010X307</td>
<td>834 for HIX</td>
</tr>
</tbody>
</table>
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.
- 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:

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.

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 shown in the following image.
B. Using iWay Integration Tools to Configure an Ebix for HIPAA
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.
4. Right-click the Integration Explorer pane and select *Import* from the context menu, as shown in the following image.
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.

![Image of the Open dialog]

The Open dialog is displayed.
7. Select the downloaded \textit{HIPAA\_5010X299.ebx} file from the physical drive location and then click \textit{Open}, as shown in the following image.
8. Click Next, as shown in the following image.
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.
Your iT 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.

![Properties window](image)

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

![Properties window](image)

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.
Your iTT 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.
4. Click *Browse* and choose a folder location to store the Ebix, and then click *Next*, as shown in the following image.
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.

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 preparers 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.
- 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">00001000</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*0!&
   SE*8*0001!&
   GE*1*1!&
   IEA*1*100000001!&
</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>
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

<table>
<thead>
<tr>
<th>Hex</th>
<th>Char</th>
<th>Hex</th>
<th>Char</th>
<th>Hex</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>SOH</td>
<td>16</td>
<td>SYN</td>
<td>2F</td>
<td>/</td>
</tr>
<tr>
<td>02</td>
<td>STX</td>
<td>17</td>
<td>ETB</td>
<td>3A</td>
<td>:</td>
</tr>
<tr>
<td>03</td>
<td>ETX</td>
<td>18</td>
<td>CAN</td>
<td>3B</td>
<td>;</td>
</tr>
<tr>
<td>04</td>
<td>EOT</td>
<td>19</td>
<td>EM</td>
<td>3C</td>
<td>&lt;</td>
</tr>
<tr>
<td>05</td>
<td>ENQ</td>
<td>1A</td>
<td>SUB</td>
<td>3D</td>
<td>=</td>
</tr>
<tr>
<td>06</td>
<td>ACK</td>
<td>1B</td>
<td>ESC</td>
<td>3E</td>
<td>&gt;</td>
</tr>
<tr>
<td>07</td>
<td>BEL</td>
<td>1C</td>
<td>FS</td>
<td>3F</td>
<td>?</td>
</tr>
<tr>
<td>08</td>
<td>BS</td>
<td>1D</td>
<td>GS</td>
<td>40</td>
<td>@</td>
</tr>
<tr>
<td>09</td>
<td>TAB</td>
<td>1E</td>
<td>RS</td>
<td>5B</td>
<td>[</td>
</tr>
<tr>
<td>Hex</td>
<td>Char</td>
<td>Hex</td>
<td>Char</td>
<td>Hex</td>
<td>Char</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td>0A</td>
<td>LF</td>
<td>1F</td>
<td>US</td>
<td>5C</td>
<td>\</td>
</tr>
<tr>
<td>0B</td>
<td>VT</td>
<td>21</td>
<td>!</td>
<td>5D</td>
<td>]</td>
</tr>
<tr>
<td>0C</td>
<td>FF</td>
<td>23</td>
<td>#</td>
<td>5E</td>
<td>^</td>
</tr>
<tr>
<td>0D</td>
<td>CR</td>
<td>24</td>
<td>$</td>
<td>5F</td>
<td>_</td>
</tr>
<tr>
<td>0E</td>
<td>SO</td>
<td>25</td>
<td>%</td>
<td>60</td>
<td>'</td>
</tr>
<tr>
<td>0F</td>
<td>SI</td>
<td>26</td>
<td>&amp;</td>
<td>7B</td>
<td>{</td>
</tr>
<tr>
<td>10</td>
<td>DLE</td>
<td>27</td>
<td>'</td>
<td>7C</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>DC1</td>
<td>28</td>
<td>(</td>
<td>7D</td>
<td>)</td>
</tr>
<tr>
<td>12</td>
<td>DC2</td>
<td>29</td>
<td>)</td>
<td>7E</td>
<td>~</td>
</tr>
<tr>
<td>13</td>
<td>DC3</td>
<td>2A</td>
<td>*</td>
<td>7F</td>
<td>DEL</td>
</tr>
<tr>
<td>14</td>
<td>DC4</td>
<td>2B</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>NAK</td>
<td>2D</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.
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*000000905*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*i*000000905~

Sample Output XML Document

The following is the XML file that results from inbound processing of the preceding HIPAA document.

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<documents>
  <ValidationReport>
    <Report>
      <Success/>
    </Report>
  </ValidationReport>
  <input>
    <![CDATA[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~]]>
  </input>
```
Sample Output XML Document

```xml
<input>
<documents>
</documents>
</input>

<output>
<HIPAA_276_005010X212>
<ISA>
</_01_Functional_Identifier_Code>HR</_01_Functional_Identifier_Code>
<_02_Application_Senders_Code>SENDER CODE</_02_Application_Senders_Code>
<_03_Application_Receiver_Code>RECEIVER CODE</_03_Application_Receiver_Code>
<_04_Date>19991231</_04_Date>
<_05_Time>0802</_05_Time>
</output>
```

180 Information Builders
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
<?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_>
            </ISA>
        </HIPAA270>
    </input>
</documents>
```
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.
**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*. 

![Diagram showing how to add a variable to a root node](image-url)
The variable appears in the Output: XML pane, as shown in the following image.
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.
3. Double-click the `GET` action support box, select `ADD` from drop-down list, and click `OK`, as shown in the following image.

The updated variable appears.
The following screen appears in the transform.

![Diagram of the transform]

The TotalOrder group already contains the element, `detaillinecount`, to contain the counter, as shown in the following image.

![Diagram of TotalOrder group]

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.

```xml
<PriceCost>
  <work>3</work>
</Detail>
</DetailLine>
</TotalOrder>
  <Total_DetailLineCount>3</Total_DetailLineCount>
</TotalOrder>
</Document>
```
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.