SeeBeyond™ eBusiness Integration Suite

HIPAA Implementation Guide

Release 4.5.3

SeeBeyond Proprietary and Confidential

The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

e*Gate, e*Index, e*Insight, e*Way, e*Xchange, e*Xpressway, iBridge, IQ, SeeBeyond, and the SeeBeyond logo are trademarks and service marks of SeeBeyond Technology Corporation. All other brands or product names are trademarks of their respective companies

© 2002 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

Portions of this software are copyrighted by Intalio, 2000–2002

This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.

Version 20020828132242.

Contents

List of Tables

Chapter 1

Introduction

This chapter introduces you to the HIPAA Implementation Guide .

The Health Insurance Portability & Accountability Act of 1996 (HIPAA) is a mandate that was developed specifically for the healthcare industry. For transactions related to healthcare, HIPAA uses a customization of X12. For pharmaceutical transactions, the HIPAA standard uses NCPDP (National Council for Prescription Drug Programs) transactions.

This book includes an overview of HIPAA, and then specific information relating to the installation and contents of SeeBeyond's HIPAA implementations.

1.1 Introduction to HIPAA

HIPAA amends the Internal Revenue Service Code of 1986. Its primary purpose is to set standards for transactions and information within the healthcare industry. HIPAA requires:

- Improved efficiency in healthcare delivery by standardizing electronic data interchange
- Protection of confidentiality and security of health data through setting and enforcing standards.

More specifically, HIPAA calls for:

- Standardization of electronic patient health, administrative, and financial data
- Unique health identifiers for individuals, employers, health plans, and healthcare providers
- Security standards protecting the confidentiality and integrity of "individually identifiable health information," past, present or future

1.2 Intended Reader

The reader of this guide is presumed to be a developer or system administrator with responsibility for developing components of the e*Gate™ system or the SeeBeyond™ eBusiness Integration Suite, to be thoroughly familiar with Windows 2000 and

Windows NT operations and administration, and to be familiar with Microsoft Windows graphical user interfaces.

1.3 Supporting Documents

The following SeeBeyond documents provide additional information that might prove useful to you.

- *HIPAA ETD Library User's Guide*
- *X12 ETD Library User's Guide*
- *NCPDP-HIPAA ETD Library User's Guide*
- *X12 ETD Library User's Guide*
- *e*Gate Integrator Installation Guide*
- *e*Xchange Partner Manager Installation Guide*
- *e*Xchange Implementation Guide*
- *e*Index Global Identifier User's Guide*

Chapter 2

HIPAA Overview

This chapter provides an overview of HIPAA, including general information, a list of the specific transactions that comprise the HIPAA standard, and the structure of HIPAA envelopes, data elements, and syntax.

2.1 Introduction to HIPAA

The following sections provide an introduction to HIPAA.

2.1.1. What Is HIPAA?

HIPAA is an acronym for the Health Insurance Portability and Accountability Act of 1996. This Act is designed to protect patients. Among other things, it defines specifications affecting standards of treatment and privacy rights. It provides a number of standardized transactions that can be used for such things as a healthcare eligibility inquiry or a healthcare claim. HIPAA legislates that all of the healthcare industry will be on the same implementation timetable. All institutions performing electronic healthcare insurance transactions must implement these standardized transactions by October 2002, unless an extension to October 2003 has been granted to the institution.

HIPAA has three primary goals.

- Define standards for electronic transactions and code sets used for financial and clinical electronic data interchange (EDI).
- Establish unique identifiers for the three participants in the provision of healthcare services: providers, payers, and employers.
- Mandate security and privacy standards for the protection of individually identifiable healthcare information.

HIPAA regulations affect many organizations dealing with the medical industry, such as:

- providers
- health plans
- employers

For provider systems, HIPAA does not mandate they perform EDI and therefore many of the standards do not apply. However, if a provider elects to perform EDI, then their

EDI transactions are required to be in compliance with all of the HIPAA transaction requirements.

The impact of HIPAA on health plans is potentially far greater than the impact on provider systems. Where providers have the option to perform EDI, HIPAA requires health plans to support the nine standard EDI transactions (for a list of the nine standard transactions, see **Table 1 on page 16**).

2.1.2. HIPAA Goals

Electronic Health Transactions Standards

Historically, health providers and plans used many different electronic formats. Implementing a national standard means that everyone uses one format, thereby simplifying and improving transaction efficiency. HIPAA defines standards for nine healthcare transactions, and mandates that all providers, health plans, and employers performing EDI comply with the standards. The HIPAA transactions cover the following situations:

- eligibility for a health plan
- claims or equivalent encounter information
- payment and remittance advice
- coordination of benefits
- health claims status
- referral certification and authorization
- first report of injury
- enrollment and disenrollment in a health plan
- health plan premium payments
- pending transaction - health claims attachments

For transactions relating to such things as healthcare claims, the HIPAA standard uses a range of customized X12 transactions as listed above. For transactions relating to prescriptions, HIPAA uses NCPDP transactions. For information on HIPAA X12, see **"HIPAA X12" on page 15**, and for information on NCPDP, see **"NCPDP" on page 18**.

Health organizations must also adopt standards for the coding of information within the individual transactions. For example, coding systems that describe diseases, injuries, and other health problems, as well as their causes, symptoms, and actions taken must be uniform. HIPAA also establishes national standards for these code sets based on currently available standards (for example, ICD9, CPT4, and so on).

Unique Identifiers

As well as meeting the need for standard encoding of information within the transactions, HIPAA also establishes the requirement to uniquely identify the

participants involved in the provision of and payment for healthcare services. These participants include the provider, the payer (health plan), and the employer.

Security and Electronic Signatures

The security standards provide a level of protection for all health information that is housed or transmitted electronically and that pertains to an individual. Organizations that use electronic signatures also have to meet a standard ensuring message integrity, user authentication, and non-repudiation.

The security standard mandates safeguards for physical storage and maintenance, transmission, and access of individual health information. It applies not only to the transactions adopted under HIPAA, but to all individual health information that is maintained or transmitted.

The security standard does not require specific technologies to be used; solutions vary from business to business, depending on the needs and technologies in place.

No transactions adopted under HIPAA currently require an electronic signature.

Privacy and Confidentiality

In general, privacy is about who has the right to access personally identifiable health information. This covers all individually identifiable health information regardless of whether the information is, or has been, in electronic form.

The privacy standards:

- Limit non-consensual use and release of private health information.
- Give patients the right to access their medical records and to know who else has accessed them.
- Restrict most disclosure of health information to the minimum needed for the intended purpose.
- Establish new criminal and civil sanctions for improper use or disclosure.
- Establish new requirements for access to records by researchers and others.

2.1.3. Trading Partner Agreements

Although the regulations mandated by HIPAA are very strict and specific, it is still important to have trading partner agreements for individual trading relationships.

Following the HIPAA standard ensures that transactions comply with the regulations mandated by the government. HIPAA requirements are completely described in the HIPAA implementation guide for each transaction, and must not be modified by a trading partner.

However, there is room for negotiation in terms of the specific processing of the transactions in each trading partner's individual system. The specifics might vary between sites. The trading partner agreement is a useful repository for this type of site-specific information.

There are three levels of information that guide the final format of a specific transaction. These three levels are:

- The HIPAA standard

  HIPAA publishes a standard structure for each HIPAA transaction.

- Industry-specific Implementation Guides

  Specific industries, including healthcare, publish implementation guides customized for that industry. Normally, these are provided as recommendations only. However, in the case of HIPAA, it is extremely important to follow these guidelines since HIPAA regulations are law.

- Trading Partner Agreements

  It is normal for trading partners to have individual agreements that supplement the standard guides. The specific processing of the transactions in each trading partner's individual system might vary between sites. Because of this, additional documentation that provides information about the differences is helpful to the site's trading partners and simplifies implementation. For example, while a certain code might be valid in an implementation guide, a specific trading partner might not use that code in transactions. It would be important to include that information in a trading partner agreement.

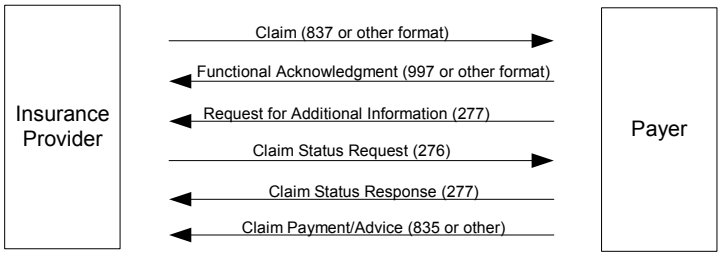
2.2 HIPAA X12

The following section provides an introduction to HIPAA X12, including information about HIPAA X12 transactions and message structures.

2.2.1. Sample Scenario

An example of a HIPAA X12 transaction exchange between a healthcare provider and a payer is shown in Figure 1.

Figure 1 Sample HIPAA Transaction Exchange



2.2.2. Batch and Real-Time Transactions

The HIPAA standard supports the sending and receiving of messages in both batch and real-time (interactive) modes.

Batch

In batch mode, transactions are grouped together and multiple transactions are sent in a single message. The batch can either go directly to the receiver or via a clearing house. The connection does not remain open while the receiver processes the messages. If there is an expected response transaction (for example, a 271 in response to a 270) the receiver creates the response transaction offline and then sends it.

Real-Time

If a transaction is processed in real time, it is sent individually. Transactions that require an immediate response are normally sent in real time. In real-time mode, the sender sends the request transaction, either directly or through a clearing house, and the connection is kept open while the receiver processes the transaction and returns a response transaction. Response times are typically no more than one minute, and often less.

In real-time mode, the receiver must send a response; either the expected response transaction, such as a 271 in response to a 270, or a standard acknowledgment such as the 997.

2.2.3. Data Overview

HIPAA X12 transactions all use the standard components of the X12 standard, covered in **Appendix A**, **"ASC X12 Overview" on page 100**.

Specifically, the transactions use the following elements:

- Segments
- Data elements
- Looping structures

In addition, consistent use of these transaction elements is required across all HIPAA implementation guides.

The X12 portion of the HIPAA ETD Library provides Event Type Definitions for all nine standard X12 transactions that have been adopted by HIPAA, as listed in Table 1.

These transactions are based on the October 1997 X12 standard; that is, Version 4, Release 1, Sub-release 0 (004010) (version 4010).

Table 1 HIPAA X12 Transactions

Number	Name
270	Eligibility Coverage or Benefit Inquiry
271	Eligibility Coverage or Benefit Information
276	Health Care Claim Status Request
277	Health Care Claim Status Notification
278	Two versions: Health Care Services Review Information and Request for Review/Response to Request

Table 1 HIPAA X12 Transactions

Number	Name
820	Payment Order Remittance Advice
834	Benefit Enrollment and Maintenance
835	Health Care Claim Payment Advice
837	Health Care Claim (three versions: Professional, Dental, and Institutional)

The NCPDP portion of the HIPAA ETD Library provides request and response transactions for all the HIPAA-approved NCPDP transaction codes, as listed in Table 2.

Table 2 NCPDP-HIPAA Transaction Codes

Code	Transaction Name
E1	Eligibility Verification
B1	Billing
B2	Reversal
B3	Rebill
P1	Prior Authorization Request and Billing
P2	Prior Authorization Reversal
P3	Prior Authorization Inquiry
P4	Prior Authorization Request Only
N1	Information Reporting
N2	Information Reporting Reversal
N3	Information Reporting Rebill
C1	Controlled Substance Reporting
C2	Controlled Substance Reporting Reversal
C3	Controlled Substance Reporting Rebill

2.2.5. Acknowledgment

The HIPAA X12 transactions either have specific designated response transactions, or use the standard 997 Functional Acknowledgment.

The 997 is used by the following transactions:

- 837 (sent by the payer to acknowledge claim receipt)
- 277 (sent by the provider to acknowledge receipt of a Health Care Payer Unsolicited Claim Status request)
- 277 (sent by the provider to acknowledge receipt of a Health Care Claim Request for Additional Information)

- 835 (sent by the provider to acknowledge receipt of a Health Care Claim Payment/ Advice notification)

2.3 NCPDP

The following section provides an introduction to NCPDP, including information about NCPDP transactions and message structures.

2.3.1. What Is NCPDP?

NCPDP (National Council for Prescription Drug Programs) is an organization, accredited by ANSI, that is tasked with standards development for the pharmaceutical industry.

The mission of NCPDP is twofold:

- To create and promote standards for data interchange in pharmaceutical services (including electronic data interchange)
- To provide educational information and resources to members

In following the above, NCPDP hopes to enhance the quality of healthcare by creating, and encouraging the use of, a high-quality data interchange standard.

2.3.2. History

Pharmacies started moving toward computerization in the late 1970s. By 1977, standardization of forms was seen as a need and NCPDP was formed to meet that need. The first NCPDP standardized form was released in 1978. By 1987, electronic claims were introduced. In 1988, version 1.0 of the NCPDP Telecommunications Standard was released. Since then, the standard has continued to be developed.

2.3.3. What Is the NCPDP Telecommunications Standard?

The NCPDP Telecommunications Standard (Telecom) is a data transmission standard specifically designed for the communication of prescription information between pharmacies and payers. It was developed to provide a consistent standard for pharmaceutical drug claims. This standard defines the structure for prescription claim transactions between providers (for example, pharmacies or doctors) and claims adjudicators. It provides for communications in both directions.

The HIPAA standard for electronic healthcare transactions and code sets adopts the following NCPDP standards for pharmacy claims:

- NCPDP Telecommunication Standard Format, Version 5.1
- NCPDP Batch Standard, Version 1 Release 1 (1.1)

Note: *At the request of NCPDP, DSMO (Designated Standards Maintenance Organization) has revised support from Batch Standard Version 1.0 to Batch*

Standard Version 1.1 for usage with Telecommunication Standard Version 5.1. For backwards compatibility, Batch 1.0 files are still provided in the NCPDP-HIPAA ETD Library.

Health plans, healthcare clearinghouses, and healthcare providers who use electronic transactions are required to use these standards after October 2002, unless they have been granted an extension to October 2003.

2.3.4. Components of an NCPDP Envelope

NCPDP messages are all ASCII text with the exception of the delimiters, which are hexadecimal.

Structure of a Request Transaction

An NCPDP Business Request Transaction has the following main parts:

- An electronic envelope, including such items as sender ID, receiver ID, message type, password, and date/time.
- A prescriber section, including such items as prescriber identifier (for example, State License), prescriber name, business name, business address, and specialty code.
- A pharmacy section, including such items as NCPDP provider identifying code, pharmacy name, pharmacist name, pharmacy address, and pharmacy phone number.
- A patient section, including such items as patient name, date of birth, gender, address, and the pharmacy or prescriber's internal ID code for the patient.

Structure of a Response Transaction

An NCPDP Response Transaction includes:

- An electronic envelope.
- A response status, which can be any one of the following:
  - An acknowledgment of receipt of the transaction
  - A "paired" response transaction (this might approve the request, deny it, or approve it with changes)
  - An error acknowledgment

2.3.5. Batching in NCPDP

NCPDP supports batching of transactions.

An NCPDP batch file is comprised of three sections:

- A transaction header (one per batch)

- Data (one or many, to a maximum of 9,999,999,997), each containing a Transaction Reference Number to uniquely identify the transaction within the file
- A transaction trailer (one per batch)

2.3.6. Acknowledgment Types

The transactions defined within NCPDP are of two types: request transactions and response transactions. There are no discrete acknowledgment transactions.

However, a "captured" response (one of the several types of response transactions) can be used when information transactions are sent and require nothing more than acknowledgment of their receipt at the processor or endpoint.

2.3.7. Transaction Codes

NCPDP uses transaction codes to indicate the type of transaction being performed.

A list of NCPDP transaction codes is provided in **Table 2 on page 17**.

2.4 Additional Information

For more information on HIPAA, visit the following Web sites:

- **http://www.hcfa.gov/HIPAA/HIPAAHM.HTM**
- **http://www.hipaa-dsmo.org**
- **http://www.wedi.org/**
- **http://www.claredi.com/**
- **http://aspe.os.dhhs.gov/admnsimp/**

For more information on NCPDP, visit the official NCPDP Web site at this address:

- **http://www.ncpdp.org/**

Note: *This information is correct at the time of going to press; however, SeeBeyond has no control over these sites. If you find the link is are no longer correct, use a search engine to search for* ***HIPAA*** *or* ***NCPDP****.*

Chapter 3

The SeeBeyond Solution

This chapter provides an overview of SeeBeyond's solution for HIPAA implementations.

3.1 Introduction

The SeeBeyond eBusiness Integration Suite supports the translations and field mapping features needed to comply with nationally mandated code sets while preserving local autonomy. It also includes the pre-built message structures for all HIPAA transactions, and the ability to map proprietary, internal messaging formats to the appropriate HIPAA transactions.

3.1.1. e*Xchange Partner Manager

e*Xchange Partner Manager allows organizations to use technology for business-to-business (B2B) and business-to-consumer (B2C) e-commerce. In addition to the standard e*Xchange functionality, e*Xchange provides pre-built Java validation rules for the nine standard X12 transactions for HIPAA that are Claredi compliant, as well as HIPAA-compliant security for transmission over public networks, if desired.

3.1.2. e*Gate Integrator

e*Gate Integrator can be used without e*Xchange Partner Manager to transform data from other formats to the standard X12 format for HIPAA. It also provides connectivity with, and between, the diverse systems and applications that participate in the HIPAA transactions. The HIPAA solution requires two e*Gate add-ons; the HIPAA ETD Library and the X12 4010 ETD Library.

3.1.3. e*Index Global Identifier

e*Index Global Identifier provides the ability to maintain internal numbering for providers, health plans, employers, and patients, and cross-indexes these internal numbers to the nationally assigned identifiers for external communication. This may become useful when introducing the HIPAA requirement of unique identifiers.

3.2 e*Xchange Partner Manager

e*Xchange Partner Manager provides functionality to receive, process, and route inbound and outbound messages in batch, fast batch, and interactive transmission modes.

For HIPAA, e*Xchange provides pre-built validation rules for the nine standard transactions, as well as HIPAA-compliant security for transmission over public networks, if desired. Specifically, e*Xchange provides the following functions:

- Validates messages based on Event Type Definitions and Collaboration Rules that conform to HIPAA regulations. These validations are Claredi-compliant (see **"Testing the SeeBeyond Solution" on page 25**) for information about Claredi), and are described in detail in **Chapter 4**.
- Automatically generates and reconciles acknowledgments, providing the acknowledgment handling required by HIPAA.
- Stores trading partner information, messages, acknowledgments, and errors in a database. HIPAA requires that seven years of patient data be stored. This is handled by the e*Xchange database; and the e*Xchange Repository Manager allows management and archiving of data.
- Allows users to view messages and supports security of data access via user ID and password verification via the e*Xchange Web interface.
- Provides an audit of who views the data. This is a HIPAA mandate that SeeBeyond supports via the Web interface Message Tracking audit feature.
- Tracks transactions per trading partner, which is also a HIPAA mandate supported via Message Tracking.

eSecurity Manager offers the following additional functionality that may be desired by HIPAA:

- Exchange content integrity.
- Origin authentication via digital signatures.
- Non-repudiation of transmission and receipt.

3.2.1. e*Xchange Files for HIPAA Transactions

e*Xchange includes Java Collaboration Rules (**.class**) for the standard HIPAA X12 transactions. These files work with the HIPAA ETD Library Java (**.xsc**) files and the X12 4010 ETD Library Monk files (**.ssc**), which are provided as an e*Gate-add on (for more information, see the *HIPAA ETD Library User's Guide*).

To indicate that the e*Xchange HIPAA files match the HIPAA standard, they have "hipaa" in the file name. An example is shown below.

- Java Collaboration Rules file for an X12 270, Eligibility Coverage Inquiry, May 2000 release: **val_X12_004010X092_00_hipaa270_EligCoveOrBeneInqui.class**.

In some cases, there are different versions of the HIPAA files. For example, there are three versions for 837, as follows:

- validate_X12_004010X098_00_hipaaQ1_837_HealCareClai (professional)
- validate_X12_004010X097_00_hipaaQ2_837_HealCareClai (dental)
- validate_X12_004010X096_00_hipaaQ3_837_HealCareClai (institutional)

HIPAA files are automatically installed in the following location:

- **\eGate\server\registry\repository\default\collaboration_rules\HIPAA**

For a complete list of HIPAA files provided with e*Xchange see **"e*Xchange Files for HIPAA Transactions" on page 114**.

3.2.2. The e*Xchange HIPAA e*Way

e*Xchange provides an e*Way within the e*Xchange Schema, **ewHipaaValidation**, that serves as a placeholder for the HIPAA Java Collaboration Rules. This e*Way is not designed to publish or subscribe to any data, but it does need to be started in order for HIPAA transactions to be processed using the Collaboration Rules.

Each Collaboration Rule in the HIPAA e*Way corresponds with a specific X12 transaction type, and so also corresponds with a specific ETD provided in the HIPAA ETD Library. Using the HIPAA e*Way Collaboration Rules in addition to the HIPAA ETDs provides more comprehensive validation of your HIPAA X12 transactions.

To use the Collaboration Rules of the HIPAA e*Way, you must specify "HIPAA" as the Validation Collaboration Type in the Message Profile, and you must specify the name of the Collaboration (not the Collaboration Rule file name) as the Validation Collaboration.

3.2.3. The HIPAA ETD Library

The HIPAA ETD Library e*Gate add-on provides JAVA ETD files for each HIPAA X12 transaction. These files work together with the e*Xchange HIPAA Collaboration Rules to validate the HIPAA rules in the X12 implementation guides. Each ETD includes Java methods to provide error message handling, indicate national identifier preferences, and set or retrieve delimiters. These ETDs cannot be modified.

The output ETD for all of the HIPAA validation Collaborations is **X12ValidationResult**, which contains information about any data, envelope, or unmarshalling errors found while processing HIPAA transactions. The format of the error output from **X12ValidationResult** is described in **"Validation Error Reporting" on page 36**.

3.2.4. Complete HIPAA Transaction ETDs

In addition to the standard e*Xchange format files, installation also includes a version of HIPAA Monk ETD files that include the GS/GE and ISA/IEA enveloping. These are suitable for use outside e*Xchange when a complete Event structure is required; for example, when using e*Gate to translate from X12 to a business application's

proprietary data format. Note that these files do not provide the comprehensive HIPAA validations that are provided in the HIPAA ETD Library files.

There are HIPAA X12 ETD files for the May 1999 and May 2000 HIPAA implementation guide releases.

The file names have "_xlate" (for May 1999 files) or "_xlat" (for May 2000 files) appended to the file name to indicate that these are the translation files and that they include the interchange control and functional group header and footer. To indicate that the files match the HIPAA standard, they have "hipaa" in the file name. Examples are shown below.

- ETD file for an X12 270, Eligibility Coverage Inquiry, May 1999 release: **X12_270EligibCoverageBenefitInquiry_004010X092_hipaa_xlate.ssc**
- ETD file for an X12 270, Eligibility Coverage Inquiry, May 2000 release: **X12_270EligibCoverageBenefitInquiry_4010X092_00_hipaa_xlat.ssc**

These files are stored in the following location:

- **\eGate\server\registry\repository\default\monk_scripts\eXchange\HIPAA**

For a complete list of files, see **"HIPAA e*Xchange Files for e*Gate" on page 115**.

Note: *These files use dynamic delimiters, and can only be used in translating from X12 to a proprietary format.*

3.3 e*Gate Integrator

The HIPAA ETD Library includes the pre-built Java message structures for all HIPAA transactions, and the ability to map proprietary, internal messaging formats to the appropriate HIPAA transactions. For more information on the HIPAA ETD Library, refer to the *HIPAA ETD Library User's Guide*.

Note: *Although you can use e*Gate to create EDI messages that conform to HIPAA standards, you also need to ensure that other HIPAA standards are also met; for example, privacy and security. e*Xchange Partner Manager provides a more complete HIPAA solution.*

All the HIPAA X12 ETDs accept either standard ANSI X12 format or XML format as input. By default, output is ANSI. However, you can optionally define that the output is XML. Although the XML format does not meet the HIPAA requirements for EDI, this format is useful when displaying the data in a Web browser.

3.3.1. e*Gate Files for HIPAA Transactions

The X12 portion of the HIPAA ETD Library provides Java Event Type Definitions (**.xsc** and **.jar** files) for all nine standard X12 transactions that have been adopted by HIPAA. These ETDs are stored in the following locations:

- **\<eGate>\server\registry\repository\default\etd\templates\Hipaa_1999**

- **\<eGate>\server\registry\repository\default\etd\templates\Hipaa_2000**

These transactions are based on the October 1997 X12 standard; that is, Version 4, Release 1, Sub-release 0 (004010) (version 4010).

For a list of files, see **"e*Gate Files for HIPAA Transactions" on page 117**.

The NCPDP portion of the HIPAA ETD Library provides request and response transactions for all the HIPAA-approved NCPDP transaction codes. These ETDs are stored in:

- **\<eGate>\server\registry\repository\default\etd\templates\NCPDP**

For a list of NCPDP-HIPAA files, see **"e*Gate Files for HIPAA Transactions" on page 117**.

3.4 Testing the SeeBeyond Solution

Claredi and the WEDI SNIP (Workgroup for Electronic Data Interchange, Strategic National Implementation Process) task group have developed recommended types of HIPAA testing. Testing is performed in six different areas, ranging from basic integrity checking to a more detailed level of testing. The types of HIPAA tests include:

- Type 1: Syntax Integrity Testing
- Type 2: Syntactical Requirement Testing
- Type 3: Balancing Testing
- Type 4: Situation Testing
- Type 5: External Code Set Testing
- Type 6: Product Types or Lines of Service Testing

Additional testing specific to Trading Partners is also identified, but is offered as an additional service. Using Claredi's HIPAA test tools and data, SeeBeyond conducted a series of tests on its HIPAA solution and successfully passed certification requirements for WEDI SNIP types 1 through 6. Even with successful testing, Claredi notes that "certification is only useful when it is not 'generic', but identifies and matches both the requirements of the receiver and the capabilities of the sender".

SeeBeyond products, having successfully gone through Claredi certification process, will greatly enable implementations to quickly meet certification requirements.

Chapter 4

e*Xchange HIPAA Validations

This chapter provides information about the HIPAA validation rules supported by e*Xchange. HIPAA validation rules are described for each transaction set in the X12 implementation guides.

4.1 Overview

Under HIPAA regulations, several validations must be performed at the segment and element levels to comply with the rules stated in the HIPAA implementation guides. SeeBeyond meets this requirement by incorporating automatic validations into the HIPAA ETD libraries and Collaboration Rules.

The validations discussed in this chapter apply to the May 2000 standards. For more information about the e*Xchange files that support these standards, see **"e*Xchange Files for HIPAA Transactions" on page 22** and **"Complete HIPAA Transaction ETDs" on page 23**. Additional information is also available in **Appendix B**.

4.1.1. Validated Transaction Sets

Validations need to be performed at the element, segment, and loop levels against HIPAA transactions to be sure that the data contained within each transaction meet the rules put forth in the HIPAA implementation guides. Nine standard X12 transaction types have been adopted by HIPAA (for a list of transaction types, see **Table 1 on page 16**). e*Xchange supports validations for all nine transaction sets, based on the May 2000 standards.

4.1.2. HIPAA Validations Summary

The X12 implementation guides describe the requirements for the elements, segments, and loops contained in the X12 transaction sets. e*Xchange uses the HIPAA ETD libraries and specialized Collaboration Rules to automatically perform many of the HIPAA validations for you. Seven primary types of validations are performed.

- External Code Set Validations
- Data Pattern Validations
- Date/Time Pattern Validations
- Balance Validations

- HL Segment Validations
- Conditional Test Validations
- Unique Identifier Validations

Most of these validations are incorporated into the HIPAA ETD library. Some validations, most notably the balancing logic, are controlled through Collaboration rules. This chapter provides information about how e*Xchange supports the HIPAA validations for each type. Notes about specific HIPAA validation rules are included at the end of this chapter.

4.2 External Code Set Validations

Under HIPAA, a code set is any set of codes that is used to encode data elements within a transaction, such as tables of terms, medical concepts, medical diagnosis codes, country codes, medical procedure codes, and so on. Code sets include both the codes and their descriptions. Code sets for medical data are required for data elements in the transaction standards adopted under HIPAA for diagnoses, procedures, and drugs.

Code set validations address type 5 testing, as recommended by WEDI SNIP and Claredi, by checking for valid code set values and their appropriate use as specified in the X12 HIPAA implementation guides. Note that the code sets may be configured to be case insensitive, allowing both uppercase and lowercase characters (for example, the Currency code set).

4.2.1. Code Set Modifiers

Code set modifiers only exist for code sets 130, 133, and 513 (for a description of these code sets, see Table 3 below). In order for the modifier data element to be validated, the code set preceding the modifier must be 130, 133, or 513. Otherwise the modifier data element is not validated.

4.2.2. Code Set Validations

e*Xchange validates the code sets mandated under the HIPAA regulations for format and, in most cases, for content. Table 3 lists the code sets supported by the e*Xchange validations.

Table 3 HIPAA External Code Sets

Code Set ID	Code Set Description	Notes
4	ABA Routing Number	This code set changes frequently, so is only validated for format.

Table 3 HIPAA External Code Sets

Code Set ID	Code Set Description	Notes
5	Countries, Currencies, and Funds	Both 2-letter and 3-letter country codes are validated. Includes the following code set names: ▪ 5_Country ▪ 5_Country_2 ▪ 5_Country_3 ▪ 5_Currency
16	D-U-N-S Number	This code set changes frequently, so is only validated for format.
22	States and Outlying Areas of the US	Includes the following code set names: ▪ 22_USState ▪ 22_CAProvince
41	Universal Product Code	This code set changes frequently, so is only validated for format.
43	FIPS-55 - Named Populated Places	This code set is validated for format only.
51	ZIP Code	5-digit ZIP codes are supported. For data with 9-digit ZIP codes, only the first 5 digits are checked.
60	Depository Financial Institution (DFI) Identification Number	This code set changes frequently, so is only validated for format.
77	X12.3 Data Element Dictionary/ X12.22 Segment Directory	This code set is only used in the 997 Functional Acknowledgment transaction set.
91	Canadian Financial Institution Branch and Institution Number	This code set is validated for format only.
94	International Organization for Standardization (Date and Time)	This code set is validated for date and time format.
102	Languages	
121	Health Industry Identification Number	This code set is validated for format only.
130	Health Care Financing Administration Common Procedural Coding System	The 130_ProcedureModifier code set modifier is used to modify this code set.
131	ICD-9-CM	Accepts data with or without decimal points. Includes the following code set names: ▪ 131_Procedure ▪ 131_Disease

Table 3 HIPAA External Code Sets

Code Set ID	Code Set Description	Notes
132	National Uniform Billing Committee (NUBC) Codes	Includes the following code set names: ▪ 132_Revenue ▪ 132_PlaceOfService ▪ 132_Occurrence ▪ 132_OccurrenceSpan ▪ 132_Value ▪ 132_Condition
133	Current Procedural Terminology (CPT) Codes	The 133_ProcedureModifier code set modifier is used to modify this code set.
134	National Drug Codes	
135	American Dental Association Codes	
139	Claim Adjustment Reason Codes	
158	HCFA Codes	Includes the following: 1 Carrier Identification Number 2 Fiscal Intermediary Identification Number 3 Medicare Provider and Supplier Identification Number
229	Diagnosis Related Group Number (DRG)	
230	Admission Source Codes	
231	Admission Type Codes	
235	Claim Frequency Type Codes	
236	Uniform Billing Claim Form Bill Type	
237	Place of Service from Health Care Financing Administration Claim Form	
239	Patient Status Codes	
240	National Drug Code by Format	Includes the following code set names: ▪ 240_N1 ▪ 240_N2 ▪ 240_N3 ▪ 240_N4
245	National Association of Insurance Commissioners (NAIC) Code	This code set is validated for format only.
307	National Association of Boards of Pharmacy Number	This code set is validated for format only.
359	Treatment Codes	
411	Remittance Remark Codes	

Table 3 HIPAA External Code Sets

Code Set ID	Code Set Description	Notes
457	NISO Z39.53 Language Code List	
507	Health Care Claim Status Category Code	
508	Health Care Claim Status Code	
513	Home Infusion EDI Coalition (HIEC) Product/Service Code List	The 513_ProcedureModifier code set modifier is used to modify this code set.
522	Health Industry Labeler Identification Code	March 2002
530	NCPDP Reject/Payment Codes	
537	HCFA National Provider Identifier	This code set is not yet final.
540	Health Care Financing Administration National Plan ID	This code set is not yet final.
DOD1	Military Rank and Health Care Service Region	
DOD2	Paygrade	
	Modifiers for code sets 130 and 133. Sometimes these codes sets are required for the same element, as in 837I, 837P, 835, 276, and 277 transactions.	Includes the following code set names: ▪ 130_ProcedureModifier ▪ 133_ProcedureModifier ▪ 513_ProcedureModifier
	Provider Taxonomy Code	Only the first nine characters are checked.
	Health Insurance Prospective Payment System (HIPPS) codes	Supported code set name is HIPPS_Nursing_Rate_Code.

4.3 Data Pattern Validation

Several elements in X12 transactions must be validated for data format. These elements primarily contain identification codes and telephone numbers. The e*Xchange system validates a number of data patterns as listed in Table 4 below. In Table 4, "#" represents any numeric character and "X" represents any alphanumeric character.

Table 4 Data Pattern Validations

Qualifier	Description	Required Format
SSN	Social Security Number	####-##-#### or #########
FTIN	Federal Tax Identification Number	###-##-#### or #########

Table 4 Data Pattern Validations

Qualifier	Description	Required Format
UPIN	Provider UPIN Number	XXXXXX
EIN	Employer Identification Number	##-####### or #########
PH	Telephone Number	Must be all numeric characters and at least 10 characters long
D-U-N-S	D-U-N-S Number	#########
D-U-N-S4	D-U-N-S Number + 4-digit Suffix	##############
EM	e-mail Address	only one '@' exists (not in beginning or end)
OCI	Employer Identification Number preceded by “1”; D-U-N-S number preceded by “3”; user assigned number preceded by “9”	1######### 3######### 9#########
1EIN	Employer identification number preceded by “1”	1#########

4.4 Date/Time Pattern Validations

Date/time elements in the X12 transaction sets require different formats, often depending on a date/time qualifier element preceding the date/time element. The format qualifiers are usually specified directly in the transaction and define the required format for the date/time elements they precede. e*Xchange validates a number of date/time patterns. A complete list of validated formats appears in Table 5 below. This table uses the following abbreviations:

- C - Century
- Y - Year
- M - Month (MM represents the month in numeric characters; MMM in alphabetic)
- D - Day (DD represents the day of month; DDD the day of year)
- Q - Quarter
- H - Hour
- m - Minute
- S - Seconds

Table 5 Date and Time Pattern Validations

Date Qualifier	Description
CC	First two digits of the year, expressed in the format CC
CD	Month and year, expressed in the format MMMYYYY
CM	Date, expressed in the format CCYYMM

Table 5 Date and Time Pattern Validations

Date Qualifier	Description
CQ	Date, expressed in the format CCYYQ
CY	Year, expressed in the format CCYY
D6	Date, expressed in the format YYMMDD
D8	Date, expressed in the format CCYYMMDD
DA	Date, expressed in the format CCYYMM
DB	Date, expressed in the format MMDDCCYY
DD	Day in numeric format DDD
DT	Date and time, expressed in the format CCYYMMDDHHmm
DT	Date and time range, expressed in the format CCYYMMDDHHmmSS-CCYYMMDDHHmmSS
EH	Last digit of the year and the Julian date, expressed in the format YDDD
KA	Date, expressed in the format YYMMMDD
MD	Month and day, expressed in the format MMDD
MM	Month in numeric format MM
RD	Range of dates, expressed in the format MMDDCCYY-MMDDCCYY
RD2	Range of years, expressed in the format YY-YY
RD4	Range of years, expressed in the format CCYY-CCYY
RD5	Range of years and months, expressed in the format CCYYMM-CCYYMM
RD6	Range of dates, expressed in the format YYMMDD-YYMMDD
RD8	Date, expressed in the format CCYYMMDD-CCYYMMDD
RDM	Range of dates, expressed in the format YYMMDD-MMDD
RDT	Range of date and time, expressed in the format CCYYMMDDHHmm-CCYYMMDDHHmm
RMD	Range of months and days, expressed in the format MMDD-MMDD
RMY	Range of years and months, expressed in the format YYMM-YYMM
RTM	Range of time, expressed in the format HHmm-HHmm
RTS	Date and time, expressed in the format CCYYMMDDHHmmSS
TC	Julian date, expressed in the format DDD
TM	Time, expressed in the format HHmm
TQ	Date, expressed in the format MMYY
TR	Date and time, expressed in the format DDMMYYHHmm

Table 5 Date and Time Pattern Validations

Date Qualifier	Description
TS	Time, expressed in the format HHmmSS
TT	Date, expressed in the format MMDDYY
TU	Date, expressed in the format YYDDD
YM	Year and month, expressed in the format YYMM
YMM	Range of year and months, expressed in the format CCYYMMM-MMM
YY	Last two digits of the year, expressed in the format YY

4.5 Balancing Validations

In certain transaction sets, some elements are governed by mathematical formulas. For example, the value of one element must equal the sum of other elements in the transaction, or the count of a element must be within a certain range.

The SeeBeyond HIPAA solution validates several of the "balance" requirements. The balancing validations performed by e*Xchange are listed in Table 6 below.

Table 6 Balancing Validations

Transaction Set	Balancing Formula
820	BPR02 = SUM of 2300A_RMR04 + SUM of 2300B_RMR04 RMR04 = RMR05 + ADX01
835	$SVC02 - \sum_{2110CAS} (CAS06 + CAS09 + CAS12 + CAS15 + CAS18) \equiv SVC03$
835	$CLP03 - \left(\sum_{2100CAS} (CAS06 + CAS09 + CAS12 + CAS15 + CAS18) + \sum_{2110CAS} (CAS06 + CAS09 + CAS12 + CAS15 + CAS18) \right) \equiv CLP04$
835	$\sum_{2100} CLP04 - \sum_{PLB} (PLB04 + PLB06 + PLB08 + PLB10 + PLB12 + PLB14) \equiv BPR02$
837d	Count of 2330B_REF < 3 2300_CLM02 = SUM of 2400_SV302
837i	2300_CLM02 = SUM of 2400_SV202
837p	Count of 2400_CRC < 3 Count of 2400_DTP < 15 Count of 2330B_REF < 3 2300_CLM02 = SUM of 2400_SV102

4.6 HL Segment Validations

The HL segment is used in several X12 transaction sets to identify hierarchical levels of detail information, such as relating dependents to a subscriber. Hierarchical levels may differ between transaction sets. Each implementation guide states the available levels, repeat values, and whether subordinate levels exist for the transaction sets covered by the guide. e*Xchange validates the following HL segment rules:

- HL01

  This element must be unique or must start with "1" and increment by one, depending on the transaction set.

- HL02

  A parent loop must exist with the ID indicated in this element.

- HL03

  All data that follows an HL segment is associated with the entity identified by the level code; this association continues until the next occurrence of an HL segment. The child parent relationship must also be accurate; for example, the loop containing dependent information must be a child of the subscriber loop.

- HL04

  If this element is "1", a child element must exist. If this element is "0" (zero), no child elements should exist.

4.7 Conditional Validations

In order to achieve HIPAA standards for transaction sets, certain conditional, or situational, tests must be performed against elements, segments, and loops in HIPAA transactions. Some of these conditions are based on inter-segment or intra-segment dependencies. These tests are performed when a element, segment, or loop must meet a requirement under the specified condition. Conditional validations are defined in the X12 implementation guides and encompass a wide variety of scenarios, such as dependencies on values in other elements or segments, required values under certain conditions, mandated identifiers, and so on. For example, if a claim is for an accident, then the accident date is required. The conditional validations in e*Xchange address the type 4 situation testing recommended by WEDI SNIP and Claredi.

While the e*Xchange system provides a comprehensive set of conditional validations, some of rules set forth in the implementation guides may be interpreted differently by different groups. For more information about the validation rules found to be ambiguous, see **"HIPAA Validation Rules Implementation Notes" on page 39**.

4.8 HIPAA Unique Identifier Validations

4.8.1. About National Identifiers

HIPAA requires unique identification of the entities involved in healthcare services, including patients, providers, health plans, and employers. The required identifiers created to satisfy the unique identification requirement are:

- National Plan ID
- National Provider ID
- National Employer ID
- National Individual ID

4.8.2. National Identifiers in e*Xchange

Not all of the HIPAA mandated identifiers are currently required, so e*Xchange allows support for each identifier to be configured. This is done by setting a flag that alerts e*Xchange to whether the mandate should be followed. This provides the ability to use alternative identifiers before the mandate goes into effect, and to then switch to the required identifiers and enable identifier validation once the mandate does go into effect.

The identifier flag has two settings for each identifier type.

- Prior

  This flag indicates that the HIPAA mandated unique identifier is not in place and allows alternative identifiers to be used.

- Mandate

  This flag indicates that the HIPAA mandated identifier is in place and does not allow any alternative identifiers to be used.

Identifier flags are configured in the Collaboration Rules for each transaction. For information about the Java methods to use to set these flags, see Chapter 5, "HIPAA ETD Library Java Methods", in the *HIPAA ETD Library User's Guide*. You can use the Collaboration Rules Editor to modify the flag settings.

4.8.3. National Identifiers and Transaction Sets

The identifier settings you configure affect how certain elements within each transaction set are validated. When the flag is set to "true", the affected elements are validated to ensure that the only allowed value is the Identification Code Qualifier specified for the HIPAA identifier in the implementation guide. When the flag is set to "false", alternative Identification Code Qualifiers can be used.

A list of elements affected by the identifier flags appears in Table 7 below. Only elements where "Yes" is specified are affected by the identifier flag for each transaction.

Table 7 Transaction Elements Affected by Identifier Settings

	TRANSACTION ELEMENTS				
TRANSACTION SET	**NM108 (ZZ/XV/XX)**	**N103 (XX/XV)**	**SBR09 (Not Use)**	**PRV_2 (HPI)**	**ENT_3 (65)**
270	Yes			Yes	
271	Yes			Yes	
276	Yes				
277	Yes				
278_request	Yes				
278_response	Yes				
820	Yes	Yes			Yes
834	Yes	Yes			
835	Yes	Yes			
837D	Yes		Yes		
837I	Yes		Yes		
837P	Yes		Yes		

4.9 Validation Error Reporting

Errors resulting from HIPAA validations can be viewed in three different ways. Message Tracking in the e*Xchange Web interface displays error messages in HTML format; the e*Xchange Schema log files display error messages in ASCII text format; and the X12ValidationResult ETD structures error messages in a format that can be translated into a 997 Functional Acknowledgment response. This section discusses the format of the HIPAA transaction error message. Information about the codes used in the error messages is provided in **Appendix C**.

4.9.1. The HIPAA Validation e*Way

When a data element fails the HIPAA validations performed by e*Xchange, an error message is produced in a specialized format defined by the validation ETD **X12ValidationResult** in the Collaborations of the **ewHipaaValidation** e*Way. These errors can then be processed by e*Xchange to produce 997 Functional Acknowledgment transactions. You can also output the data directly from the validation ETD. The same errors can be viewed in Message Tracking and in the log files.

Validation e*Way Error Message Format

Each validation error consists of 11 elements, including header and error information. The first four elements of the output message contain header information. Elements

five through eleven of the output message display information about the errors that occurred, including an error description and references to 997 Functional Acknowledgment error codes.

Table 8 below lists each element in the validation error message, along with a description of each element. For a complete list of validation error messages and the corresponding 997 Functional Acknowledgment error codes, see **Appendix C**.

Table 8 Validation Error Message Format

Element Position	Description
1	Level indicator
2	Segment ID Code
3	Segment position (this value is equivalent to the Foresight line number minus 2)
4	Loop ID code
5	Segment syntax error as it appears in the AK304 element in the 997 transaction
6	Element position in segment
7	Sub-element position in composite
8	Reference number of the data element
9	Error code as it appears in the AK403 element in the 997 transaction
10	The data contained in the offending element
11	e*Xchange error description

Understanding the Error Message

In the error message format described in Table 8 above, elements two through ten are used to compose the 997 Functional Acknowledgment response. The level indicator in element one allows you to filter out certain low-level errors. The descriptions in element 11 are written to the e*Xchange database. The elements in the error messages are delimited by asterisks (*) and error messages are delimited from each other by a pipe (|).

Following is an example of an e*Xchange validation error message.

```
5*CLP*20*2100*8*8**1331*7*Fa* 2100_CLP_8 at 20 [Fa]: Value cannot be
found in code source 237
```

The above example can be broken down as follows:

- This example illustrates a level **5** error in the **CLP** segment in position **20** and in the **2100** loop.

  5*CLP*20*2100*8*8**1331*7*Fa* 2100_CLP_8 at 20 [Fa]: Value cannot be found in code source 237

- In the AK304 element of the 997 Functional Acknowledgment, this is an error **8** (defined as "Segment has data errors").

  5*CLP*20*2100***8***8**1331*7*Fa* 2100_CLP_8 at 20 [Fa]: Value cannot be found in code source 237

- The error is in position **8** in the CLP segment (CLP_08), and there is no sub-element position noted.

  5*CLP*20*2100*8***8****1331*7*Fa* 2100_CLP_8 at 20 [Fa]: Value cannot be found in code source 237

- The reference number is **1331**, and the error in the AK403 element of the 997 Functional Acknowledgment is **7** (defined as "Invalid code value").

  5*CLP*20*2100*8*8****1331*7***Fa* 2100_CLP_8 at 20 [Fa]: Value cannot be found in code source 237

- **Fa** is the text that is in error, and the e*Xchange error message is **2100_CLP_8 at 20 [Fa]: Value cannot be found in code source 237**.

  5*CLP*20*2100*8*8**1331*7***Fa*2100_CLP_8 at 20 [Fa]: Value cannot be found in code source 237**

Important: *If the input data in error contains characters used for delimiters, those characters will be modified in the error message data. A pipe (|) in the data will be changed to "[or]"; a caret (^) or a tilde (~) will be changed to a dash (-).*

4.9.2. Message Tracking

The e*Xchange error messages produced by the HIPAA Validation e*Way can be viewed using the Message Tracking feature of the e*Xchange Web interface. Message Tracking allows you to view any messages that have been processed by e*Xchange, including any errors that might be associated with a message. This useful tool helps you to pinpoint the source of an error so it can be resolved, and provides a way for you to fix and resend any messages that had errors. For more information about Message

Tracking, see Chapter 10, "Web Interface: Message Tracking", in the *e*Xchange Partner Manager User's Guide*.

Message Tracking Error Message Format

The error messages displayed in Message Tracking differ slightly from the format of the HIPAA Validation e*Way error messages, though the information is similar. Message Tracking does not include the level indicator or the AK304 and AK403 error codes. Using the HIPAA Validation e*Way example error message from above, the same message in Message Tracking would appear as follows:

```
2100_CLP_8 at 20 [Fa]: Value cannot be found in code source 237
```

4.9.3. Log File Error Messages

The HIPAA transaction error messages can also be viewed in the e*Gate log files for the e*Xchange e*Way that is running the validations. These files are located in **\<eGate>\client\logs**, and can be viewed using any standard text editor. The format of the log file error messages is identical to that of the HIPAA Validation e*Way error message format.

For more information about e*Gate log files, see the *e*Gate Integrator Alert and Log File Reference Guide*.

4.10 HIPAA Validation Rules Implementation Notes

While SeeBeyond's HIPAA solution provides a comprehensive set of HIPAA rule validations, there are possible ambiguities in the HIPAA rules as stated in the implementation guides. General ambiguities found in the rules include:

- The rule contradicts another rule.
- The rule calls for a non-supported code set.
- The rule can be interpreted in different ways.
- Inter-segment dependencies are not specifically defined because the dependencies span across two loops.
- Certain requirements are specified according to state laws, which may differ.

In addition, the recommended number of claims for each transaction is not enforced since this is only a recommended number. For some of the rules affected by ambiguities, the HIPAA Collaboration Rules can be modified to assist with validating according to specific interpretations.

Loop Trigger Validations

The first segment in a repeating loop is known as the "loop trigger" and is required when the loop is used. Its usage only indicates if the loop is optional or required. If a

loop is used, the first segment, the trigger, of that loop is required even if it is marked Situational.

For example, for loop 2010AB in 837 (professional) transactions, the first segment, NM1, is a required segment if the loop is used even though it is marked Optional. If the NM1 segment is populated, then the remaining segments are validated per the rules in the implementation guide.

Rules Clarifications

Table 10 through Table 21 list the HIPAA guidelines found to be ambiguous in each transaction set, and clarifies how those rules are implemented in SeeBeyond's HIPAA solution. Table 9 provides a general list of validation rules that relate to multiple transaction sets.

Table 9 Notes on General HIPAA Validations

Message Elements	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
N402	▪ N402 is required only if city name (N401) is in the U.S. or Canada. ▪ Element is always required according to usage rules.	Health Care Claim: Dental, page 82	These rules appear to contradict each other. This is implemented such that N402 is only required if the value of N404 indicates that the city (N401) is in the United States or Canada. N402 is also validated for code set usage.
Multiple (REF01 is an example)	This element cannot be "SY" because the Social Security Number may not be used for Medicare. Or Social Security Number should not be used if the plan is Medicare or other federal plan.	Health Care Claim: Dental, page 84	There is no explicitly defined Medicare designator. Elements that include this rule are validated to ensure that only the listed codes are accepted as the reference qualifier.
NM109	Member identification number.	Health Care Claim Status Request and Response, page 503	Since no formatting requirements are specified for these elements, formatting is not validated.
NM104	This element is required if NM102=1 (person).	Health Care Claim: Professional, page 100	This is implemented as follows: ▪ NM104 cannot be empty if NM102=1. ▪ If NM102 does not equal 1, then NM104 must be empty.
NM105	This element is required if NM102=1 (person) and the middle name/initial of the person is known.	Health Care Claim: Professional, page 100	If NM102 does not equal 1, then NM105 must be empty.

Table 9 Notes on General HIPAA Validations

Message Elements	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
NM107	This element is required if known.	Health Care Claim: Professional, page 101	If NM102 does not equal 1, then NM107 must be empty.
2400_DTP	Required if the service date (2400_DTP) is different than the service date reported at the DTP segment in the 2300 loop.	Health Care Claim: Dental, page 273	It does not affect processing if the dates match in loops 2300 and 2400, so these dates are only validated for format.
N2 Segment	Required if the name in NM103 is greater than 35 characters.	Health Care Claim: Dental, page 79	There is no explicit indicator that the name is greater than 35 characters, so this rule is not enforced.
Multiple (2310A_NM1, for example)	Element, segment, or value is required on all inpatient claims or encounters.	Health Care Claim: Institutional, page 321	There is no clear indicator of whether a claim is for an inpatient visit, so this rule cannot be validated.

Table 10 Notes on Validations for Transaction Set 270

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2110C_III	Use this segment only one time for the Principal Diagnosis Code and only one time for Facility Type Code.	Health Care Eligibility Benefit Inquiry and Response, page 101	This segment can be repeated ten times, but can only be repeated once for each of these two code types.
INS	Use this segment only in the absence of all of the data for the mandated search option identified in Section 1.3.8. Use only if it is necessary to identify the dependent's relationship to the subscriber identified in loop 2100C or the dependent's birth sequence in the case of multiple births with the same birth date.	Health Care Eligibility Benefit Inquiry and Response, page 126	If all of the data for the mandated search option is entered, any information in the INS segment is ignored. The second requirement is not based on an explicit indicator within the transaction.

Table 10 Notes on Validations for Transaction Set 270

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
Multiple	Rules that pose requirements on a response message based on information contained in a request message. For example, for 2110_REF01 in the 835 transaction set, the rule for the code "6R", Provider Control Number states "This is the Line Item Control Number submitted in the 837, which is utilized by the provider for tracking purposes, if submitted on the claim this must be returned on remittance advice."	Health Care Claim Payment/Advice, page 154	Currently, the e*Xchange validations do not span across multiple messages and validations are not performed on rules that are dependent on information contained in a message other than the message being validated. e*Xchange does validate the unique ID retrieved from the request transaction with the unique ID retrieved from the response transaction.

Table 11 Notes on Validations for Transaction Set 271

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2110D_III	Use this segment only one time for the Principal Diagnosis Code and only one time for Facility Type Code.	Health Care Eligibility Benefit Inquiry and Response, page 140	This segment can be repeated ten times, but can only be repeated once for each of these two code types.
2100C	Use of this segment is required if the transaction is not rejected and address information is available from the information source's database.	Health Care Eligibility Benefit Inquiry and Response, page 200	Address information is dependent upon a source other than the current message, so whether address information is available cannot be verified.
2100D_INS17 and 2100C_INS17	Use to indicate the birth order in the event of multiple birth's in association with the birth date supplied in DMG02.	Health Care Eligibility Benefit Inquiry and Response, pages 210 and 215	This may be interpreted to mean that if DMG02 is not used, then INS17 should not be used. However, this is not explicitly stated so is not validated.

Table 12 Notes on Validations for Transaction Set 276

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
PER segment	PER03 and PER04 are required elements, while PER05 through PER08 are situational (that is, they are dependent on whether PER03 and PER04 exist).	Health Care Claim Status Request and Response, page 57	The stated rule indicates that a telephone number is required if this segment is used. However, the example shown contains only a fax number in the segment. This rule was implemented according to the stated rules and not the example, so a telephone number is required if any other type of communications number exists in this segment.
2200E	"Use this only if the subscriber is the patient." and "Use this segment when the patient is someone other than the subscriber."	Health Care Claim Status Request and Response, pages 103, 105, and 107	2200E contains only dependent information, so patient must be someone other than the subscriber.
2200E	1 Required for institutional claims. The date is the statement from and through date. 2 For professional claims this will be the claim from and through date.	Health Care Claim Status Request and Response, page 111	There is no explicit indication of institutional claim or professional claim within the transaction, so these requirements cannot be verified. If claim level date range is not used then the Line Service Date at 2210E is required.

Table 13 Notes on Validations for Transaction Set 277

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2200E_REF 2200E_DTP	"Use this only if the subscriber is the patient." and "Use this segment when the patient is someone other than the subscriber."	Health Care Claim Status Request and Response, pages 103, 105, and 107	2200E contains only dependent information, so patient must be someone other than the subscriber. These rules were clarified in the addenda, where the rule for 2200E_REF was modified to state that this segment should be used when the patient is someone other than the subscriber.

Table 13 Notes on Validations for Transaction Set 277

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2200D loop	▪ This loop is required. ▪ For the REF, DTP, and STC segments, "Use this only if the subscriber is the patient."	Health Care Claim Status Request and Response, pages 165 - 169	In the implementation guide, 2100D_NM has a repetition of >1 and 2200D_TRN is required, creating an inter-loop dependence; so whether the patient was also the subscriber could not be determined. These rules were modified in the addenda, so 2100D_NM now has a repetition of 1 and 2200D_TRN is not required. Based on the modifications, these conditions are validated.
2100 D loop	Loop repetition is >1.	Health Care Claim Status Request and Response, pages 74 - 76	As described above, this creates an inter-loop dependence and whether the patient is a subscriber cannot be determined. The rule in the addendum, making the loop repetition =1, is used instead.

Table 14 Notes on Validations for Transaction Set 278 Request

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2000D_DTP	Use if the certification request is pregnancy related.	Health Care Services Review — Request for Review and Response, page 100	There is no explicit indicator of pregnancy-related requests in the transaction, so this cannot be verified.
2010E_PRV	Required when requesting certification for a specialist or specialty entity.	Health Care Services Review — Request for Review and Response, page 135	There is no explicit indicator for specialists or a specialty entity in the transaction, so this cannot be verified.

Table 14 Notes on Validations for Transaction Set 278 Request

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2000F_DTP	Use this segment only if the request is for surgery and the HI Procedures segment in Loop 2000F is not used to identify specific surgical procedures. If the HI segment is valued, place the requested or anticipated surgical procedure date in the HI segment procedure date field (HIxx-4).	Health Care Services Review — Request for Review and Response, page 157	It is not explicit whether HI contains a procedure or service code, so this rule is not validated.
2000F_HI (01-7, 02-7, and so on to 12-7)	Revision level of a particular format, program, technique, or algorithm. Required if the code list referenced in HI02-1 has a version identifier. Otherwise not used.	Health Care Services Review — Request for Review and Response, pages 162 through 172	This version identifier is not used to define code set versioning. Reference dates will be used instead.
2000F_CR612	Required if different from the date of the request.	Health Care Services Review — Request for Review and Response, page 209	Processing is unaffected if the dates are the same, so identical dates are allowed in these elements.

Table 15 Notes on Validations for Transaction Set 278 Response

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2000C_HI02-3	Required if X12N syntax conditions apply.	Health Care Services Review — Request for Review and Response, page 255	X12N syntax always applies.
2000F_HI (01-7, 02-7, and so on to 12-7)	Revision level of a particular format, program, technique, or algorithm. Required if the code list referenced in HI02-1 has a version identifier. Otherwise not used.	Health Care Services Review — Request for Review and Response, pages 347 through 361	This version identifier is not used to define code set versioning. Reference dates will be used instead.

Table 16 Notes on Validations for Transaction Set 820

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
BPR03	For HIPAA Health Premium Payments, code “D” is not valid.	Payroll Deducted and Other Group Premium Payment for Insurance Products, page 37	All health premium payments are assumed to be HIPAA, so this is always verified.

Table 16 Notes on Validations for Transaction Set 820

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
REF01-14	For HIPAA Health Premium Payments, this element is REQUIRED.	Payroll Deducted and Other Group Premium Payment for Insurance Products, page 48	All health premium payments are assumed to be HIPAA, so this is always verified.
BPR07 BPR13	CODE SOURCE 60: (DFI) Identification Number	Payroll Deducted and Other Group Premium Payment for Insurance Products, pages 39 and 41	Code source 60 is ignored; code sources 4 and 91, which are more specific, are used instead.
1000A_N3 1000A_N4 1000B_N3 100B_N4	For EFT payments, these segments are not used.	Payroll Deducted and Other Group Premium Payment for Insurance Products, pages 59, 60, 66, and 67	There is no explicit indicator within the message of EFT payments, so this rule is not verified.

Table 17 Notes on Validations for Transaction Set 834

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
1000A_N103	The value 'ZZ', when used in this data element, shall be defined as "HIPAA Employer Identifier" once this identifier has been adopted.	Benefit Enrollment and Maintenance, page 36	See **"National Identifiers and Transaction Sets" on page 35** for more information about how this is implemented.
2000_INS07	This element is REQUIRED if a member is being enrolled in or is enrolled for a benefit covered by COBRA.	Benefit Enrollment and Maintenance, page 48	INS05 is the COBRA indicator used, so if INS05=C, then INS07 is required.
2100C	This loop is to be sent if the member has a mailing address different from the residence address sent in loop 2100A.	Benefit Enrollment and Maintenance, page 85	Processing is unaffected if the addresses are the same, so identical addresses are allowed in these segments.
2100D_NM108 2100F_NM108	The code "ZZ" will be used in this NM108 for the National Employer Identifier until a standard code is defined.	Benefit Enrollment and Maintenance, page 91	See **"National Identifiers and Transaction Sets" on page 35** for more information about how this is implemented.
2200_DSB08	The only allowed value is 585 - End Stage Renal Disease.	Benefit Enrollment and Maintenance, page 125	This element must contain the 585 code or it must be empty.

Table 17 Notes on Validations for Transaction Set 834

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2300 loop	Send this segment is REQUIRED when enrolling a new member or when adding, updating or removing coverage from an existing member.	pages 45 and 128	The code "024" in 2000_INS03 indicates a complete "cancellation, termination, or deletion of a subscriber or dependent". If INS03=024, loop 2300 cannot be used.
2300_REF01	Code 17 is NOT USED when the member identified in the related INS segment is not the subscriber.	Benefit Enrollment and Maintenance, page 135	If the related INS01 element equals "N", indicating it is not the subscriber, code 17 cannot be used in this element.
2320_REF01	ZZ will be used in this REF01 for National Individual Identifier until a standard code is defined.	Benefit Enrollment and Maintenance, page 153	See **"National Identifiers and Transaction Sets" on page 35** for more information about how this is implemented.

Table 18 Notes on Validations for Transaction Set 835

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2000_INS103 and INS104	▪ For INS03: Use this code [030] when sending a full roster to verify that the sponsor and payer databases are synchronized. ▪ For INS04: [XN] To be used in complete enrollment transmissions. This is used when INS03 is equal to 030 (Audit/ Compare).	Health Care Claim Payment/Advice, pages 45 - 47	If INS03 is "030", then the value in INS04 must be "XN".
BPR07 BPR13	CODE SOURCE 60: (DFI) Identification Number	Health Care Claim Payment/Advice, pages 48 and 50	Code source 60 is ignored; code sources 4 and 91, which are more specific, are used instead.
1000A_N103	Code XV, the Health Care Financing Administration National PlanID, is required if the National PlanID is mandated for use. Otherwise, one of the other listed codes may be used.	Health Care Claim Payment/Advice, page 63	No other codes are listed, so no other codes are allowed in this element.

Table 18 Notes on Validations for Transaction Set 835

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2000_LX	For Medicare Part A, write/read the LX segment once for each provider's fiscal period end year and month/ type of bill summary break in the file (TTYYMM in LX01). For Medicare Part B, write/read the LX segment once for unassigned claims using the value of "zero" and once for assigned claims using the value of "one".	Health Care Claim Payment/Advice, page 79	There are no explicit Medicare indicators. Any Medicare indicators are located within an inner loop (CLP) that can have multiple repetitions and therefore multiple values. These rules are not verified.
2000_TS302	Code identifying the type of facility where services were performed; the first and second positions of the Uniform Bill Type code or the Place of Service code from the Electronic Media Claims National Standard Format	Health Care Claim Payment/Advice, page 81	Code source 237 is used for this element.
2110_DTM	For Medicare service, this segment is required (for Part A, use "through date" if no service date is present).	Health Care Claim Payment/Advice, page 146	The DTM segment has multiple repetitions, and it is not explicitly stated how many repetitions are required. This rule is not verified.

Table 19 Notes on Validations for Transaction Set 837D

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
1000A_PER04, PER06, and PER08	When the communication number represents a telephone number in the United States and other countries using the North American Dialing Plan (for voice, data, fax, etc.), the communication number should always include the area code and phone number using the format AAABBBCCCC.	Health Care Claim: Dental, page 64	Telephone number elements are only checked to be sure they are at least 10 characters long and only contain numeric characters.
2300_CLM02	For encounter transmissions, zero (0) may be a valid amount.	Health Care Claim: Dental, page 151	Since it does not specify otherwise, zero (0) is allowed in all cases.
2300_CLM07	Required for Medicare claims only.	Health Care Claim: Dental, page 152	There is no explicit Medicare indicator, so this rule cannot be verified.

Table 19 Notes on Validations for Transaction Set 837D

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2310B_NM1	Required when the Rendering Provider NM1 information is different than that carried in either the Billing Provider NM1 or the Pay-to Provider NM1 in the 2010AA/AB loops.	Health Care Claim: Dental, page 195	This is implemented as follows: ▪ If 2310B_NM1 is used, it must be different than 2010AA_NM1 and 2010B_NM1. ▪ If 2310B_NM1 is used, the NM109 elements in the loops cannot match.
2310C_REF	Required when a secondary identification number is necessary to identify the entity. The primary identification number should be carried in the NM109.	Health Care Claim: Dental, page 207	There is no explicit indicator of when a secondary identification number is required, so this rule cannot be verified.
2320_AMT	The amount carried in this segment is the total amount of money paid by the payer to the patient (rather than to the provider) on this claim.	Health Care Claim: Dental, page 226	This appears to be a balance requirement but does not list the elements to sum. This cannot be verified. This applies to other Coordination of Benefits (COB) segments.

Table 20 Notes on Validations for Transaction Set 837I

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2000B_SBR09	Same as the qualifier used in CLP06 of 835 Health Care Claim Payment.	Health Care Claim: Institutional, page 105	This rule is dependent on information contained in a different transaction, so is not verified.
2000C_PAT08	Required on claims/encounters for delivery services to report newborn's birth weight.	Health Care Claim: Institutional, page 107	There may be several loops with several admission dates, so there is no explicit indicator of patient's age. This rule is not verified.
2300_CN1	The developers of this implementation guide recommend that for noncapitated situations, contract information be maintained in the receiver's files and not be transmitted with each claim whenever possible. It is recommended that submitters always include CN1 for encounters that include only capitated services.	Health Care Claim: Institutional, page 176	This is a recommendation and not a requirement, so it is not enforced.

Table 20 Notes on Validations for Transaction Set 837I

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2300_HCP01	Referenced code set for pricing methodology is not standard.	Health Care Claim: Institutional, page 309	This code set is defined at the Trading Partner level and is not verified.
2305_CR7	This segment is required to convey Home Health Plan of Treatment information for this claim when applicable.	Health Care Claim: Institutional, page 314	This segment is only used for home health care information, so this is self-validating.
2305_HSD06	Required if the physician's order or prescription for the service contains the data.	Health Care Claim: Institutional, page 318	The order or prescription cannot be referenced from the transaction, so this rule cannot be verified.
2430_CAS	Inpatient or Outpatient - Service Line Adjustments	Health Care Claim: Institutional, page 494	This appears to be a balance requirement but does not list the elements to sum. This cannot be verified. This applies to other Coordination of Benefits (COB) segments.

Table 21 Notes on Validations for Transaction Set 837P

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2000B_SBR05	Required when the destination payer (Loop 2010BB) is Medicare and Medicare is not the primary payer (SBR01 equals "S" or "T").	Health Care Claim: Professional, page 111	There is no explicit Medicare indicator, so this rule cannot be verified.
2300_CLM05-3	Code 8 may only be used where permitted by state law. See the NUBC UB92 manual for definitions of these codes. With the exception of #1 (Original) use 6, 7, and 8 for claims that have already been finalized in the payer's system. 2810 Permissible code values for this sub-element: 1 - ORIGINAL (Admit thru Discharge Claim) 6 - CORRECTED (Adjustment of Prior Claim) 7 - REPLACEMENT (Replacement of Prior Claim) 8 - VOID (Void/Cancel of Prior Claim)	Health Care Claim: Professional, page 174	Code source 235 should be used instead. This was modified in the addenda to the X12 implementation guides.

Table 21 Notes on Validations for Transaction Set 837P

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2300_REF	Clinical Laboratory Improvement Amendment (CLIA) number is required on Medicare and Medicaid claims for any laboratory forming tests covered by the CLIA Act.	Health Care Claim: Professional, page 231	There is no explicit indicator of tests covered by the CLIA act, so this rule cannot be verified.
2300_REF	The Demonstration Project Identifier is required on claims/ encounters where a demonstration project is being billed/reported. This information is specific to the destination payer reported in the 2010BB loop. If other payers have a similar number, report that information in the 2330 loop which holds that payer's information.	Health Care Claim: Professional, page 242	There is no explicit indicator for demonstration projects, so this rule cannot be verified.
2300_CRC07	▪ [CRC segment is] required on ambulance claims/encounters, i.e. when CR1 segment is used. ▪ [CR1 segment is] required on all claims involving ambulance services.	Health Care Claim: Professional, pages 257 and 259	This could be interpreted as either or both of the following: ▪ If CRC exists, then CR1 must exist. ▪ If CR1 exists, then CRC must exist. Implementation of this rule only supports CR1 as the indictor, so only the second interpretation is verified (that is, if CR1 exists, then CRC must exist).
2400_SV1 and 2300_HI	Requirements for the relationship between 2400_SV1 elements and 2300_HI elements.	Health Care Claim: Professional, pages 265-270 and 400-407	Although it is not explicitly specified, if an element in the SV1 segment pointing to an element in the HI segment is populated, then the corresponding element in the HI segment must also be populated.
2305_HSD01 through HSD08	Required if the order/prescription for the service contains the data.	Health Care Claim: Professional, pages 279, 280, and 281	The order or prescription cannot be referenced from the transaction, so this rule cannot be verified.
2400_HCP01	Referenced code set for pricing methodology is not standard.	Health Care Claim: Institutional, page 309	This code set is defined at the Trading Partner level and is not verified.
2330B_NM109	This number must be identical to SVD01 (Loop ID-2430) for COB.	Health Care Claim: Professional, page 361	There are multiple SVD segments, and it is not possible to determine which instance of SVD01 to match.

Table 21 Notes on Validations for Transaction Set 837P

Message Element	Rule	HIPAA Implementation Guide Reference	SeeBeyond Implementation Notes
2400_SV111	Required if Medicaid services are the result of a screening referral.	Health Care Claim: Professional, page 406	There is no explicit indication of a screening referral, so this cannot be verified.
2400_CR3 2400_PWK	▪ Required on Medicare claims when DMERC CMN is included in this claim. ▪ Required if it is necessary to include supporting documentation in an electronic form for Medicare DMERC claims for which the provider is required to obtain a certificate of medical necessity (CMN) from the physician.	Health Care Claim: Professional, pages 410 and 421	This is implemented such that if 2400_CR3 is populated, then 2400_PWK must also be populated.
2420B_NM103	Name Last or Organization Name - NOT USED.	Health Care Claim: Professional, pages 509 and 510	The purpose of this element is to transmit the name of the entity type indicated in NM102, therefore this rule is ignored and values are allowed in this element.
2440LQ	1 Required if the provider is required to routinely include supporting documentation (a standardized paper form) in electronic format. 2 The 2440 loop is designed to allow providers to attach any type of standardized supplemental information to the claim when required to do so by the payer. The LQ segment contains information to identify the form (LQ01) and the specific form number (LQ02).	Health Care Claim: Professional, page 567	There is no explicit implementation described here, only examples of a specific case. These rules are not implemented

Chapter 5

Processing Large Transactions

e*Xchange is able to process large outbound transactions by breaking them up into smaller sections before validation. This chapter describes how large messages are processed and the settings that need to be modified to use this feature.

5.1 Overview

e*Xchange provides the ability to process large HIPAA X12 835 outbound messages in an interactive manner through the **eX_ePM** and **eX_Batch_to_External** e*Ways within the e*Xchange Schema. Large messages are processed by breaking the messages up into smaller, more manageable pieces during processing. The large message processing components provided with e*Xchange are specifically designed to handle X12 835 transactions. You can design custom components that process other types of transactions using the X12 835 components as a basis.

5.1.1. Considerations

Certain HIPAA transactions consist of very large messages, which can cause processing errors unless the large message feature is in use. This depends on many factors in the processing environment, such as available disk size, memory, message volume, and so on. For example, you might find that there is a problem with files larger than 60 MB unless you use the large message feature. In some cases larger files might process smoothly, and in other cases smaller files might cause errors. Your processing environment will determine whether you should use large message processing, and the size at which a message should be processed as a large message. You can use large message processing for any size message, however there are a few considerations.

- You cannot view large messages using the Message Tracking feature.
- Large message processing uses additional disk space.
- Large message processing may slow down processing speed due to the additional processing performed.

This section describes the ability of the **eX_ePM** and **eX_Batch_to_External** e*Ways to process large outbound messages in an interactive manner.

5.1.2. Methodology

Source System e*Way Requirements

To enable large message processing, a message must be flagged as a large message before it reaches the **eX_ePM** e*Way. This logic should be written into the e*Way processing messages from the source system into e*Xchange. If the message is to be processed as a large message, the source system e*Way must send the file name (in the format FILE:<file_name>) instead of the message body to e*Xchange.

The file name must include the full path to the file, be base64-encoded, and be populated in the Payload section of the e*Xchange Standard Event. The file must be stored in a directory location that is accessible by the e*Xchange e*Way components. If the file is already in an X12 format, the enveloping can be in ST/SE or ISA/IEA formats.

Translation Requirements

If an X12 translation must be performed once a large message file name reaches the **eX_ePM** e*Way, the translation Collaboration must be able to accept a file name as input. The output of the translation Collaboration must be the file name of the translated message in the format of FILE:<file_name>. Each translation must create an output file that contains a single transaction, which is in X12 835 format with ST/SE or ISA/IEA enveloping.

Note: *Performing translations creates a second file of approximately the same size as the original file in the data directory. This can cause the size of the directory to grow quickly, so frequent archival or removal of the data files is recommended.*

Splitting the Message

e*Xchange provides a custom large message Collaboration for the 835 transaction, **HIPAA_2K_835_Outb_validation**, that breaks up the large message into well-formed, manageable pieces that are then validated. After any translations have been performed, the name of the new file created by the translation is sent to the large message Collaboration, and the file is divided into smaller files, each containing the specified number of CLP files and each including header information and SE/ST enveloping.

Each file is sent to the 835 validation Collaboration in the **ewHipaaValidation** e*Way (**validate_X12_004010X91_00_hipaa835_HealCareClaiPaym**). Once validation is completed successfully, a new file is created that contains a copy of the message, including the correct enveloping and delimiters as defined in the Trading Partner Profile settings. This file is named by appending a 12-digit unique number to the end of the original file name. The end result of the validation is as if the message were processed as a single entity.

Post-Processing

After validation processing is complete and the final copy of the original message is created, the **eX_ePM** e*Way sends the file name to the **eX_Batch_to_External** e*Way to

be sent to the appropriate trading partner using FTP. When the file transfer to the trading partner is complete, two copies (three if translations were performed) of the message are stored in the specified data directory. These files should be archived or deleted as needed to maintain disk space on the e*Xchange server.

5.2 Implementing Large Message Processing

Large message settings are specified at the Message Profile level and within Trading Partner Attribute pairs that you add to the e*Xchange Standard Event. These settings apply whether you are using the 835 message processing functionality provided or you are using custom Collaborations and Monk scripts to split large messages of a different transaction type.

Trading Partner Attributes

Two Trading Partner Attribute name and value pairs must be added to the structure defined in **eX_StandardEvent.xsc**; two optional attribute pairs can be added for additional control. These attributes can be added using the **addNameValuePair** method (see the *e*Xchange Implementation Guide* for more information about this method). Table 22 lists each Trading Partner Attribute used for large message processing, along with the value and description for each.

Table 22 Trading Partner Attributes for Large Message Processing

TPAttribute Name	Optional/ Required	TPAttribute Value	Description
LARGE_MSG	Required	Y	This is an indicator that the message should be processed using large message processing.
MSG_ALT_ID	Required	Must exactly match the value set in the Message Alt ID field of the Message Profile of the applicable outbound Trading Partner Profile.	This is used to identify the message profile level of the Trading Partner Profile.
LARGE_MSG_SIZE	Optional	Numeric value Default: 1000	This value determines the number of CLP segments to include in each portion of the file that is sent for validation.
LARGE_MSG_INDX	Required only for files with multiple large transactions	Numeric value Default: 1	This value identifies the transactions to be processed as a large message in a file where multiple transactions require large message processing.

LARGE_MSG_SIZE

The "LARGE_MSG_SIZE" attribute determines the number of CLP loops that are pulled out of the large message file for each partial validation that is performed, controlling both the size and the number of the message portions that are sent to validation for each large message. Use this attribute if you determine that a value other than the default of "1000" would optimize performance for your system.

LARGE_MSG_INDX

The "LARGE_MSG_INDX" attribute is not required if the message file sent to e*Xchange by the source system e*Way has a single transaction set. The default value, 1, specifies that the first large transaction in a file will be processed using large message processing. If the message file contains many large transaction sets, this attribute is required to identify which of the transaction sets should be processed as large messages.

e*Xchange processes a single transaction for each subscription to an eX_eBPM queue Event. Therefore, if a message file has 20 transactions and four of them are to be processed as large transactions, the source system e*Way must publish the e*Xchange Standard Event four times. Each time the Event is published, the same file name is populated in the Payload segment, but a different index value is included to identify the transaction to be processed.

Message Profile Settings

Certain settings in the General section of the Message Profile for the outbound message must be configured for processing large messages. You can configure these settings in the e*Xchange Web interface. Table 23 lists the required settings for large message processing.

Table 23 Message Profile Settings for Large Messages

Attribute Name	Optional/ Required	Possible Values	Description
Message Alt ID	Required	Must exactly match the value set in the MSG_ALT_ID attribute in the eX_StandardEvent structure	This value is used to identify the message profile level of the Trading Partner Profile.
Validation Collaboration Type	Required	Must be set to **JAVA**.	This value indicates that the validation Collaboration file called by the splitter function is written in Java.
Validation Collaboration	Required	Must be set to **HIPAA_2K_Outb_validation** (or the name of your custom Collaboration)	This value indicates that a Collaboration that processes large messages will be used.

The validation Collaboration used for processing large messages, **HIPAA_2K_Outb_validation**, checks whether the input string is a message or a file name. If the input string is a file name, the Collaboration calls a function,

HIPAA_835_Outb_splitter, to break up the message into smaller, well-formed messages that are passed to the standard validation. If the input string is a message, the message is sent to the standard 835 validation Collaboration. If you are configuring the Message Profile for a custom configuration of large message processing, enter the name of the validation Collaboration you created for processing messages of the specified type in this field.

A Note on Transfer Modes

For large messages, the X12 Outbound processing ignores the settings of the transfer mode in the Trading Partner Profile and processes the message in an interactive manner.

5.3 Customizing Large Message Processing Components

You can use the logic provided in the files **HIPAA_2K_835_Outb_validation.tsc** and **HIPAA_835_Outb_splitter.monk** to create a custom Collaboration and splitter to process additional types of large X12 transactions by breaking them up into smaller pieces that are validated one piece at a time. To enable this process for alternate transaction types, you must create or customize three e*Gate components:

- The e*Way for the source system
- The large message Collaboration
- The Monk splitter function

Note: *When using customized components for processing large transactions, make sure to modify the Trading Partner Attributes and Message Profile settings described earlier in this chapter accordingly.*

Source System e*Way

When processing large messages, the e*Way that processes data from the source system must be configured to be able to determine whether a message should be processed as a large message or a standard message. This is based on a file size that you determine to be the cut-off size for processing standard messages. Any messages at or above the cut-off value you specify will be processed as large messages. When a message is to be processed as a large message, the source system e*Way must save the message in a file and send the file name (in the format FILE:<file name>) to e*Xchange. Make sure the file name includes the full path of the file to be processed, and the message contained in the file is in a standard X12 messaging format with the appropriate enveloping in place.

Large Message Collaboration

You can base your large message Collaboration on the existing 835 Collaboration, **HIPAA_2K_835_Outb_validation.tsc**. The purpose of this Collaboration is primarily to direct the flow of data once it determines whether it is processing a large message or a standard message. The Collaboration specifies the name of the splitter function and of

the Java validation Collaboration or Monk validation Collaboration to be used for data validation.

The logic in the large message Collaboration is as follows. If the Collaboration determines that a standard message is being processed, the message is sent directly to the HIPAA validation Collaboration. If the large message Collaboration determines that a large message is being processed, it calls the Monk splitter function to break the message up into smaller pieces. Once the message has been broken up and smaller files are created, the splitter function sends each file to the appropriate Collaboration for validation or processing. On completion, the large message Collaboration joins the message back together, including the appropriate header information, and saves the message in a file to be picked up by the FTP e*Way. This file is named by appending a 12-digit unique number to the end of the original file name.

When you customize the large message Collaboration, make sure you specify the appropriate names for the splitter function and the validation Collaborations.

Monk Splitter Function

The Monk splitter function defines how each large message will be divided into smaller, more manageable messages. Each message split from the large message must be a well-formed X12 message, including SE/ST enveloping. In the default splitter, several functions are defined that are used to retrieve information from the large message and to parse the message into smaller pieces. You can customize these functions to define the loops or segments that exist in the transaction type you are processing.

The splitter reads the Trading Partner settings to determine how many CLP segments to include in each message and which transaction in the file to process (in cases where the file contains multiple transactions). By default, the size of the parsed messages is based on the number of CLP segments. You can modify the splitter so the size of each parsed message is based on a more appropriate segment for the transaction type you are processing.

After reading the Trading Partner settings, the splitter parses the large message and sends each piece of the message to the appropriate validation Collaboration (as defined in the large message processing Collaboration).

There are several issues to consider when determining how to break up a large message, including the following:

- Any “balance” fields that need to be verified in the parsed messages.
- Segment repetition. You can base the size of each parsed message on the number of repetitions of a particular segment. In the case of 835 large message processing, the size of each partial message is based on the CLP segments since many 835 transactions contain a very large number of CLP segments.
- The best method of parsing the message, and the appropriate structure for the parsed messages.
- Any header or trailer information that needs to be appended to the file.
- Required enveloping formats.

Chapter 6

e*Xchange Implementation

This chapter discusses the steps involved to create an e*Xchange implementation that transfers HIPAA X12 data.

6.1 Overview

An e*Xchange implementation makes use of the features designed to add and remove the EDI enveloping information for messages exchanged between trading partners.

In an e*Xchange implementation, use the e*Xchange Web Interface to set up trading partner information, and the e*Gate Enterprise Manager GUI to add user-defined e*Gate components to provide connectivity to the business application or trading partner. Once this is done, the pre-configured e*Xchange e*Gate Schema components handle enveloping and de-enveloping Events as they travel through the e*Xchange system.

The major steps for an e*Xchange implementation are as follows:

1. Verify the e*Gate and e*Xchange installation.
2. Create the trading partner profiles.
3. Configure the user-defined e*Ways that will connect the business application to e*Xchange and exchange messages with the trading partner.
4. Configure the e*Xchange e*Way.
5. Run and test the scenario.

SeeBeyond supplies the sample files needed to run the e*Xchange HIPAA scenario. If you would prefer to use the files that are already set up for you, skip steps 2 through 4, and install the sample schema files, as described under **"Installing the Sample Files" on page 62**.

6.1.1. Case Study: Sending a Health Care Claim

The case study discussed in this chapter illustrates one possible implementation of sending a health care claim to a trading partner.

In this example, a Health Care Claim (837) is sent to an external trading partner, the insurance provider. The enveloping is automatically added to the message by e*Xchange based on trading partner information retrieved from the e*Xchange

database, and then the message is sent to the external system. An acknowledgment message (997) is immediately returned by the insurance provider. Then the Health Care Payment (835) is sent from the insurance provider trading partner, and an acknowledgment message (997) is returned to complete the cycle. Figure 2 shows the message flow.

Figure 2 HIPAA Message Flow

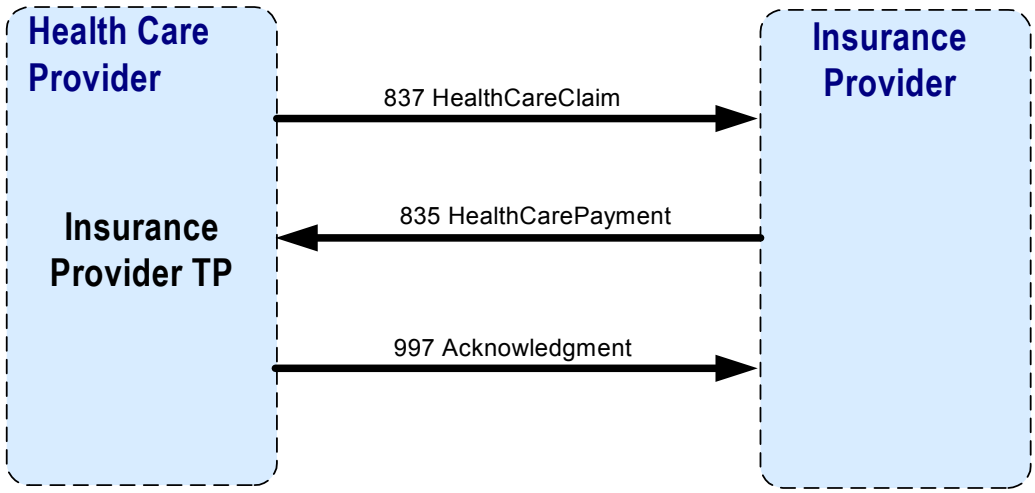


Figure 3 shows the flow of data through the sample scenario.

Figure 3 e*Xchange Scenario Data Flow

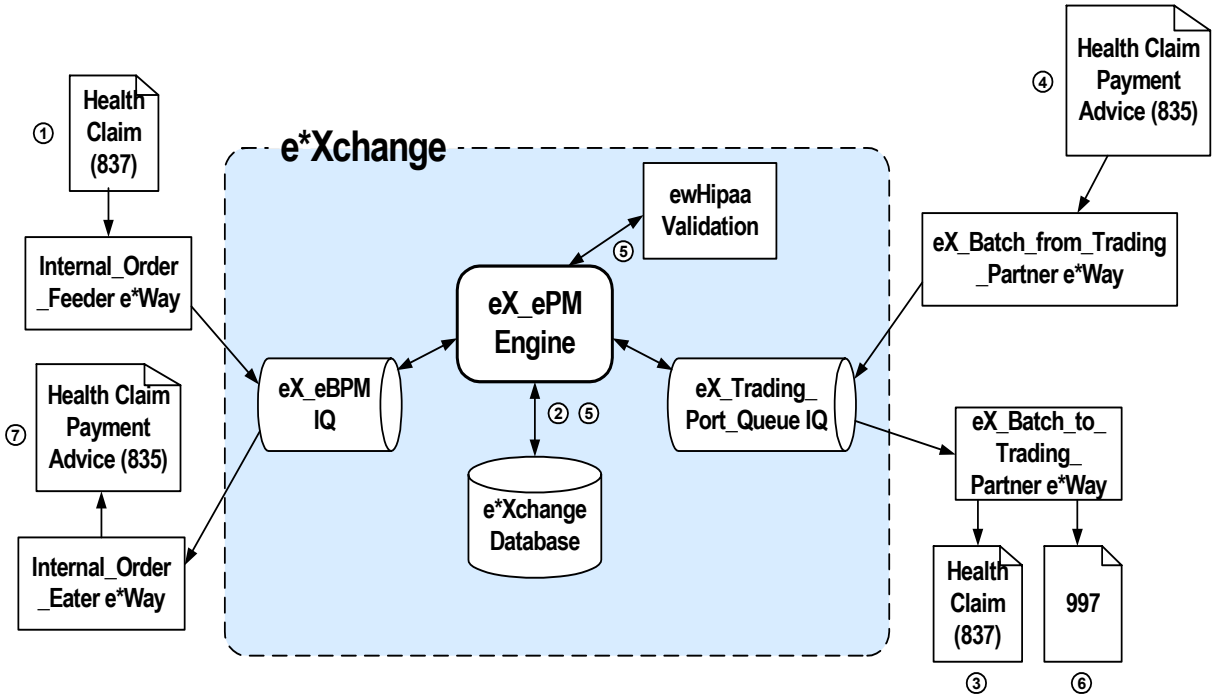


Figure 3 data flow description

1. The **Internal_Order_Feeder** e*Way picks up the health care claim message (837) and publishes it to the **eX_eBPM** IQ.
2. The e*Xchange engine picks it up from the IQ, validates it, saves it to the database, and publishes the message to the **eX_Trading_Port_Queue** IQ.
3. The **eX_Batch_to_Trading_Partner** e*Way picks up the message from the IQ and sends it out to the trading partner.
4. The **eX_Batch_from_Trading_Partner** e*Way sends the healthcare payment advice message (835) to the **eX_Trading_Port_Queue** IQ.
5. The e*Xchange engine picks the message up from the IQ, validates it using the **ewHipaaValidation** e*Way, saves it to the database, and publishes two messages:
   - an acknowledgment (997) to the **eX_Trading_Port_Queue** IQ.
   - the health claim payment advice (835) to the **eX_eBPM** IQ.
6. The **eX_Batch_to_Trading_Partner** e*Way picks up the 997 acknowledgment and sends it to the trading partner.
7. The **Internal_Order_Eater** e*Way picks up the health claim payment advice (835) from the **eX_eBPM** IQ and sends it to the internal system.

6.2 Verify the e*Gate and e*Xchange Installation

This end-to-end scenario requires e*Xchange and e*Gate to be installed on your system. For e*Xchange, you must have the Web interface and e*Xchange sample Schema installed. You also need a working e*Xchange database with the correct database connections set up. For e*Gate, install the Java HIPAA ETD Library add-on, the Database e*Way appropriate to the database platform of the e*Xchange database, and the Java Generic e*Way Extension Kit.

Refer to the *e*Xchange Partner Manager Installation Guide* for system requirements and instructions to install the e*Xchange components. The *e*Gate Integrator Installation Guide* describes e*Gate system requirements and provides instructions to install the e*Gate components.

6.3 Installing the Sample Files

The components for this implementation are provided on your installation CD, and are located in **\setup\ex\sample\HIPAA_SAMPLE_IMPLEMENTATION.zip**. You can either install the sample files and import them into the e*Gate and e*Xchange environments, or you can go through the steps of creating and configuring the required components. Either way, you need the data files located in the "data" subdirectory of the .zip file to be copied to the e*Gate environment. Follow these steps to install the components:

1. Unzip the file to a local directory.
2. Copy the folder named "data", along with all of its subfolders and files, to the e*Gate home directory.

Note: *The default registry port number is 23001.*

3. Install the e*Gate Schema using one of the following commands. The instructions refer to the schema name **HealthClaim**, however, this is user-defined.
   A. For UNIX, type the following command:

```
sh install_hipaa_sample.sh <egate_registry_host_name>
<schema_name> <user_name> <password> <egate_registry_port_num>
```

   B. For Windows, type the following command:

```
install_hipaa_sample.bat <egate_registry_host_name> <schema_name>
<user_name> <password> <egate_registry_port_num>
```

4. Use the e*Xchange Import function in e*Xchange Repository Manager to import **HIPAA.exp** into e*Xchange Partner Manager.
5. If your e*Gate home directory is named something other than "eGate", modify the file names specified for the inbound and outbound B2B protocols accordingly. See step 7 in both **"Step 3: Set up the Inbound B2B Protocol Information" on page 65** and **"Step 5: Set Up Outbound B2B Protocol Information" on page 67** for more information.
6. If your e*Gate home directory is named something other than "eGate", modify the OutputDirectory setting for the **Internal_Order_Eater** e*Way (for more information, see **"Step 1: Create and Configure the Internal_Order_Eater e*Way" on page 72**).
7. If your e*Gate home directory is named something other than "eGate", modify the PollDirectory setting for the **Internal_Order_Feeder** e*Way (for more information, see **"Step 1: Create and Configure the Internal_Order_Feeder e*Way" on page 75**
8. Configure the return message for the inbound message. See **"Step 7: Configure Return Messages for Inbound" on page 70** for detailed instructions.
9. Modify the database configuration settings in the **eX_ePM** and **eX_Poll_Receive_FTP** e*Ways by entering your e*Xchange database name, login ID, and password (for more information see **"Configure the eX_ePM e*Way" on page 78** and **"Configure the eX_Poll_Receive_FTP e*Way" on page 79**).

The steps on the following pages describe how to create each component for this implementation. You can perform these steps instead of installing the sample files as described above. See **"Running the Scenario" on page 80** for instructions to run the implementation.

6.4 Create the Trading Partner Profiles

Trading partner profiles in e*Xchange act as repositories for the information necessary to send EDI messages back and forth between entities. They contain all of the information needed to properly envelope an Event and forward it to its correct destination.

Refer to the *e*Xchange Partner Manager User's Guide* for detailed assistance with the process of creating trading partner profiles.

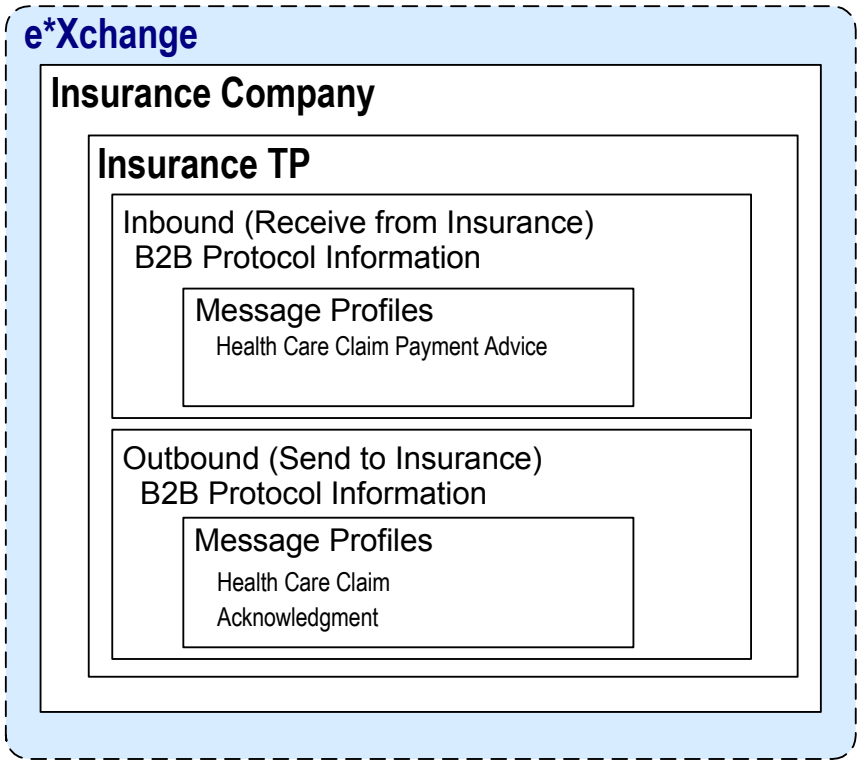
Trading Partner Information Hierarchy

e*Xchange stores trading partner information at various levels. The process of creating a trading partner profile proceeds from the most general inclusive level, that of a company with which you do business, to the most specific information regarding a message that you wish to send (the message profile).

Figure 4 shows an overview of the components that you need to create for this example, including:

- Company
- Trading Partner
- B2B Protocol Information
- Message Profiles

Figure 4 Insurance Overview



To configure the Insurance trading partner profile you must follow the steps listed below:

- **Step 1: Create the Company** on page 64
- **Step 2: Create the Trading Partner** on page 65
- **Step 3: Set up the Inbound B2B Protocol Information** on page 65
- **Step 4: Create the Inbound Message Profiles** on page 66
- **Step 5: Set Up Outbound B2B Protocol Information** on page 67
- **Step 6: Create the Outbound Message Profiles** on page 68
- **Step 7: Configure Return Messages for Inbound** on page 70

Step 1: Create the Company

To create the company

1. Log in to the e*Xchange Web interface.
2. From the **Main** page, click **Profile Management**.
3. From the **Company** page, click **New**.
4. In the **Company - adding** page, enter the **Company** name, “Insurance”.
5. Click **Next**.

   This saves your changes and returns to the **Company** page.

Note: *The security information is automatically configured for the current user.*

Step 2: Create the Trading Partner

To create the trading partner

1. From the **Company** page, ensure that the "Insurance" trading partner is selected, and then click **Continue: Trading Partner**.
2. From the **Trading Partner** page, click **New** to access the **Trading Partner - adding** page.
3. Enter the **Trading Partner Name**, "Insurance".
4. Click **Next**.

   This saves your changes and returns to the **Trading Partner** page.

Note: *The required security information defaults from the company level.*

Step 3: Set up the Inbound B2B Protocol Information

To set up the inbound B2B protocol information

1. From the **Trading Partner** page, ensure that "Insurance" is selected, and then click **Continue: B2B Protocol**.
2. From the **B2B Protocol** page, click **New** to access the **B2B Protocol - adding** page.
3. Enter the information listed in Table 24.

   In an actual implementation, your local administrator can provide you with the B2B Protocol information. For an explanation of the B2B Protocol parameters, see the *e*Xchange Partner Manager User's Guide*.

Table 24 B2B Protocol Information

Parameter	Value
eBusiness Protocol	X12
Version	4010
Direction	Inbound

4. Click **Next** to save your changes and access the **General** section.
5. Enter the information listed in Table 25.

Table 25 B2B Protocol Information, General Page

Parameter	Value
Logical Name	Insurance
Status	Active
Communication Protocol	FTP(BATCH)

6 Click **Next** to save your changes and access the **Transport Component** section.

7 In the **File Name** window, enter **<egate>\data\hipaa\TP\input*.dat**.

8 Click **Next** to access the **Message Security** section.

9 No changes are required. Click **Finish** to save the information and return to the **B2B Protocol** page.

Step 4: Create the Inbound Message Profiles

For the purposes of this scenario, you must set up the following inbound message profile:

- Health Claim Payment Advice (X12_004010X091_00_hipaa835_HealCareClaiPaym)

To set up the X12_004010X091_00_hipaa835_HealCareClaiPaym order inbound message profile

1 From the **B2B Protocol** page, click **Continue: Message Profile**.

2 From the **Message Profile** page, click the **New** button to access the **Message Profile - adding** page.

3 Enter the information listed in Table 26.

Note: *This table only lists the attributes required to make this scenario work.*

Table 26 Inbound Message Profile, General Settings

Name	Value
Name	X12_004010X091_00_hipaa835_HealCareClaiPaym
Validation Collaboration Type	HIPAA
Validation Collaboration	validate_X12_004010X091_00_hipaa835_HealCareClaiPaym
Transfer Mode	Interactive

Important: *In order to implement the full HIPAA validation functionality provided by e*Xchange, the value of the Validation Collaboration field must exactly match the name of a Collaboration contained in the **ewHipaaValidation** e*Way. Unlike Monk Validation Collaborations, the value of this field for HIPAA matches the Collaboration name instead of the Collaboration Rule file name.*

4 Click **Next** to access the **Interchange Control Envelope** section. Enter the information listed in Table 27.

Table 27 Inbound Message Profile, Interchange Control Envelope

Name	Value
ISA05 Interchange Sender Identification Qualifier	01
ISA06 Interchange Sender Identifier	6264712000
ISA07 Interchange Receiver Identification Qualifier	01
ISA08 Interchange Receiver Identifier	6264716000

Name	Value
ISA12 Interchange Version Number	00401
ISA13 Interchange Control Number	000000011

5 Click **Next** to access the **Functional Group Envelope** section. Enter the information listed in Table 28.

Note: *This table only lists the attributes required to make this scenario work.*

Table 28 Inbound Message Profile, Functional Group Envelope

Name	Value
GS01 Functional Identification Code	HP
GS02 Application Sender Code	6264712000
GS03 Application Receiver Code	6264716000
GS06 Group Control Number	1
GS07 RESP Agency Code	X
GS08 Version/Release/Industry Identification Code	004010X091

6 Click **Next** to access the **Transaction Set Envelope** section. Enter the information listed in Table 29.

Table 29 Inbound Message Profile, Transaction Set Envelope

Name	Value
ST01 Transaction Set Identification Code	835
ST02 TS Control Number	1

7 Click **Next** to access the **Return Messages** section.

8 No changes are required. Click **Finish** to save the information and return to the **Message Profile** page.

Step 5: Set Up Outbound B2B Protocol Information

To set up the outbound B2B protocol information

As a shortcut, you can copy the inbound B2B protocol information as a model for the outbound B2B protocol information.

1 On the **B2B Protocol** page, select the X12-4010-Inbound protocol that you created in **"To set up the inbound B2B protocol information" on page 65**.

2 Click **Copy**.

The **Copy Type** page appears.

3 Clear the **Include Sub-components** check box and then click **OK**.

The **B2B Protocol - copying** page appears.

4 In the **Direction** field, ensure that **Outbound** is selected.

5 Click **Next**.

The **B2B Protocol - copying**, **General** page appears.

6 No changes are needed: click **Next** to accept the values and access the **Transport Component** page.

7 In the **File Name** window, enter **<egate>\data\hipaa\TP\output\output%#.dat**.

8 Click **Next** to accept the values and access the **Message Security** page.

9 No changes are required. Click **Finish** to save the information and return to the **B2B Protocol** page.

Step 6: Create the Outbound Message Profiles

For the purposes of this scenario, you must set up the following outbound message profiles:

- Health Care Claim Message (X12_004010X098_00_hipaaQ1_837_HealCareClai)
- Acknowledgment (X12-4010-997)

To set up the X12_004010X098_00_hipaaQ1_837_HealCareClai order outbound message profile

1 From the **B2B Protocol** page, click **Continue: Message Profile**.

2 From the **Message Profile** page, click the **New** button to access the **Message Profile - adding** page.

3 Enter the information listed in Table 30.

Note: *This table only lists the attributes required to make this scenario work.*

Table 30 Outbound Message Profile, General Settings

Name	Value
Name	X12_004010X098_00_hipaaQ1_837_HealCareClai
Validation Collaboration Type	HIPAA
Validation Collaboration	validate_X12_004010X098_00_hipaaQ1_837_HealCareClai
Transfer Mode	Interactive

4 Click **Next** to access the **Interchange Control Envelope** section. Enter the information listed in Table 31.

Table 31 Outbound Message Profile, Interchange Control Envelope

Name	Value
ISA06 Interchange Sender Identifier	6264716000
ISA08 Interchange Receiver Identifier	6264712000
ISA11 IC Standards Identifier	U
ISA13 IC Control Number	38
ISA15 Test Indicator	T

5 Click **Next** to access the **Functional Group Envelope** section. Enter the information listed in Table 32.

Note: *This table only lists the attributes required to make this scenario work.*

Table 32 Outbound Message Profile, Functional Group Envelope

Name	Value
GS01 Functional Identification Code	HC
GS02 Application Sender Code	6264716000
GS03 Application Receiver Code	6264712000
GS06 Group Control Number	1211
GS08 Version/Release/Industry Identification Code	004010X098

6 Click **Next** to access the **Transaction Set Envelope** section. Enter the information listed in Table 33.

Table 33 Outbound Message Profile, Transaction Set Envelope

Name	Value
ST01 Transaction Set Identification Code	837
ST02 TS Control Number	3

7 Click **Next** to access the **Return Messages** section.

8 Select the return message (select the **Include** check box), and then enter the values shown in Table 34.

Table 34 Return Message Values: Outbound

Name	Response Time	Period	# Retries
X12_004010X098_00_hipaaQ1_837_HealCareClai	10	Minutes	1

9 Click **Finish** to save the information and return to the **Message Profile** page.

To set up the X12-4010-997 outbound inner envelope

1 From the **Message Profile** page, click the **New** button to access the **Message Profile - adding** page.

2 Enter the information listed in Table 35.

Note: *This table only lists the attributes required to make this scenario work.*

Table 35 Functional Acknowledgment, General Settings

Name	Value
Name	X12-4010-997
Transfer Mode	Interactive

3 Click **Next** to access the **Interchange Control Envelope** section. Enter the information listed in Table 36.

Table 36 Functional Acknowledgment, Interchange Control Envelope

Name	Value
ISA06 Interchange Sender Identifier	6264716000
ISA08 Interchange Receiver Identifier	6264712000
ISA11 Interchange Standards Identifier	U
ISA13 Interchange Control Number	38
ISA15 Test Indicator	T

4 Click **Next** to access the **Functional Group Envelope** section. Enter the information listed in Table 37.

Note: *This table only lists the attributes required to make this scenario work.*

Table 37 Functional Acknowledgment, Functional Group Envelope

Name	Value
GS01 Functional Identification Code	FA
GS02 Application Sender Code	6264716000
GS03 Application Receiver Code	6264712000
GS06 Group Control Number	1211
GS08 Version/Release/Industry Identification Code	004010X098

5 Click **Next** to access the **Transaction Set Envelope** section. Enter the information listed in Table 38.

Table 38 Functional Acknowledgment, Transaction Set Envelope

Name	Value
ST01 Transaction Set Identification Code	997
ST02 TS Control Number	36

6 Click **Next** to access the **Return Messages** section.

7 No changes are required. Click **Finish** to save the information and return to the **Message Profile** page.

Step 7: Configure Return Messages for Inbound

To set up the return message profile values for inbound

Once you have set up inbound and outbound message profiles, you can specify return messages.

1 From the **B2B Protocol** page, select **X12-4010-Inbound**.

2 Click **Continue: Message Profile**.

3 From the **Message Profile** page, select **X12_004010X091_00_hipaa835_HealCareClaiPaym** from the drop-down list.

4 Click the **Return Messages** link to access the **Return Messages** section.

5 Click **Edit**.

6 Select the return messages (select the check boxes), and the enter the values shown in Table 39.

Table 39 Return Message Values: Inbound

Name	Response Time	Period	# Retries
X12-4010-997	3	Minutes	1

7 Click **Apply** to save the information and return to the **Message Profile** page.

8 Click **OK**.

6.5 Clone the eXSchema

The supplied Schema named eXSchema contains the components required to run e*Xchange. Make a copy of this Schema and then configure the copy for this implementation.

To make a copy of eXSchema

1 Open eXSchema in the e*Gate Enterprise Manager GUI.

2 Export eXSchema.

3 Create a new Schema named **HealthClaim** using the exported file.

6.6 Configure the Internal_Order_Eater e*Way

This component sends the message to the internal system.

The e*Xchange Internal_Order_Eater e*Way

The e*Xchange example simulates the publication of the message to the internal system.

Follow these steps to configure the **Internal_Order_Eater** e*Way.

- **Step 1: Create and Configure the Internal_Order_Eater e*Way** on page 72
- **Step 2: Create the Internal_Order_Eater Collaboration Rule Script** on page 72
- **Step 3: Create the Internal_Order_Eater Collaboration Rule** on page 73
- **Step 4: Create the Inbound Message Profiles** on page 66

Step 1: Create and Configure the Internal_Order_Eater e*Way

To create and configure the Internal_Order_Eater e*Way

1. Create an e*Way called **Internal_Order_Eater**.
2. Open the **e*Way Properties** dialog box and, in the **Executable file** area of the **General** tab, browse for **stcewfile.exe**.
3. In the **e*Way Properties** dialog box, in the **Configuration file** area of the **General** tab, click **New**.
4. Configure the **Internal_Order_Eater** e*Way parameters using the values specified in Table 40.

Table 40 Internal_Order_Eater e*Way Parameters

Screen	Parameter	Setting
General Settings	AllowIncoming	NO
	AllowOutgoing	YES
Outbound (send) settings	OutputDirectory	<eGate>\data\hipaa\internal\output
	OutputFileName	output_order%d.dat
	(All others)	(Default)
Poller (inbound) settings	(All)	(Default)
Performance Testing	(All)	(Default)

5. When finished editing the e*Way configuration file, save your work and close the e*Way Editor.
6. Click **OK** to close the **e*Way Properties** dialog box.

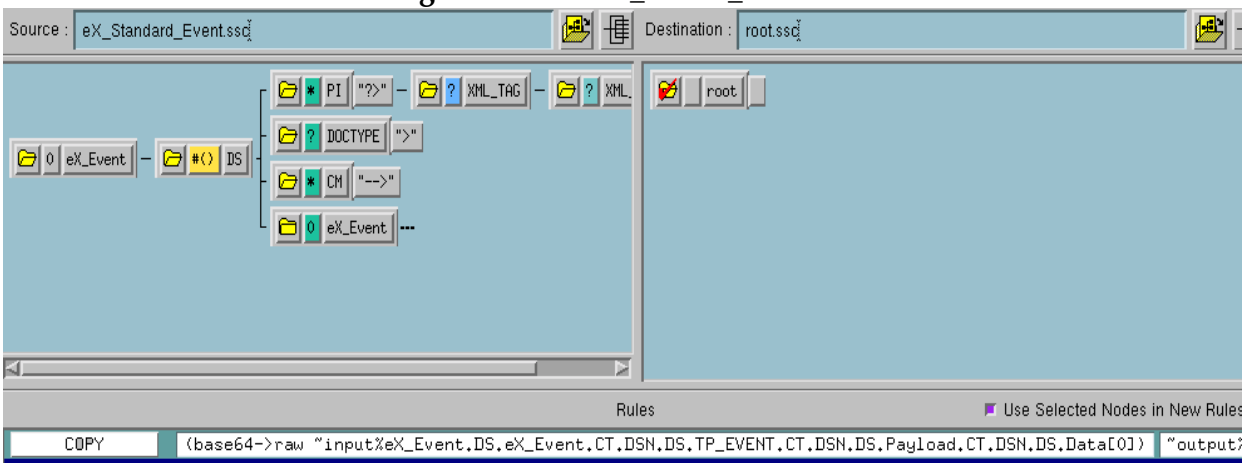
Step 2: Create the Internal_Order_Eater Collaboration Rule Script

The **Internal_Order_Eater.tsc** Collaboration Rule script helps prepare the data leaving the e*Xchange system by converting the message to raw data, and then copying it from the Payload node of the TP_EVENT section of the e*Xchange standard Event to the output ETD.

To create and configure the Internal_Order_Eater Collaboration Rule Script

1. Open the Collaboration Editor.
2. Create a new Collaboration Rules script named **Internal_Order_Eater.tsc**. The Source Event Type Definition is **eX_Standard_Event.ssc**. The Destination Event Type Definition is **root.ssc**.
3. Add the rule shown at the bottom of Figure 5.

Figure 5 Internal_Order_Eater.tsc



4 Save the Collaboration Rules script and close the Collaboration Rules Editor.

Step 3: Create the Internal_Order_Eater Collaboration Rule

Once the Collaboration Rule script has been created, you must set up the Collaboration Rules properties for the **Internal_Order_Eater** component in the Enterprise Manager GUI.

To create and configure the Internal_Order_Eater Collaboration Rule

1 Create a new Collaboration Rule named **Internal_Order_Eater**.

2 Open the **Internal_Order_Eater Collaboration Rule Properties** dialog box, and then select the **General** tab. Configure as shown in Table 41.

Table 41 Internal_Order_Eater Collaboration Rule Configuration - General Tab

Section	Value
Service	Monk
Collaboration Rule	monk_scripts\common\Internal_Order_Eater.tsc
Initialization File	monk_scripts\common\load_ext

Important: *To use the Monk function* ***base64->raw****, you must make sure the file containing this function has been loaded.*

3 Select the **Subscriptions** tab. Select **eX_to_eBPM** and move to the right pane.

4 Select the **Publications** tab. Select **eX_External_Evt** and move to the right pane.

5 Click **OK** to save the properties information and close the dialog.

Step 4: Create the Internal_Order_Eater Collaboration

The **Internal_Order_Eater** Collaboration must prepare the data leaving the e*Xchange system. The complexity of this task depends on the state of the data before the **Internal_Order_Eater** Collaboration processes it.

The **Internal_Order_Eater** Collaboration must do the following:

- Put the data into the appropriate EDI format.
- Convert the data to raw data.

To create and configure the Internal_Order_Eater Collaboration

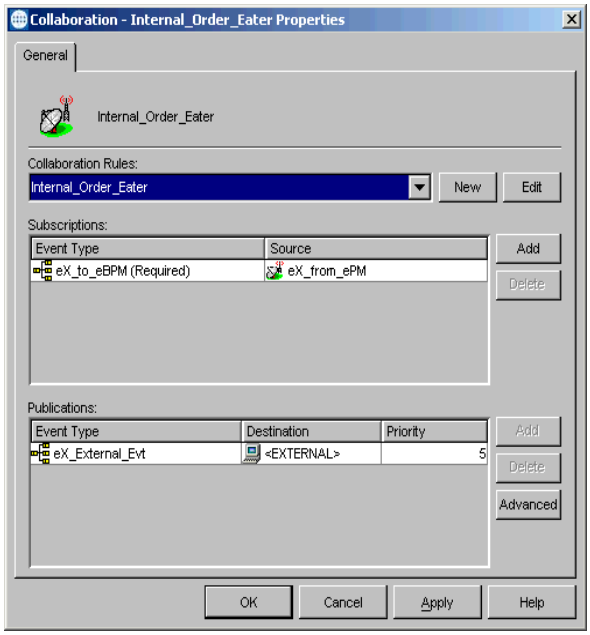
1. Select the **Internal_Order_Eater** e*Way.
2. Create a new Collaboration named **Internal_Order_Eater**.
3. Configure the Internal_Order_Eater Collaboration properties using Table 42.

Table 42 Internal_Order_Eater Collaboration configuration

Section	Value
Collaboration Rules	Internal_Order_Eater
Subscriptions	Event Type: eX_to_eBPM Source: eX_from_ePM
Publications	Event Type: eX_External_Evt Destination: <EXTERNAL>

Verify the information in the **Collaboration - Internal_Order_Eater Properties** dialog box as shown in Figure 6.

Figure 6 Collaboration - Internal_Order_Eater Properties



4. Click OK to save the Collaboration and close the dialog.

6.7 Configure the Internal_Order_Feeder e*Way

The component (e*Way or BOB) that feeds data into e*Xchange must put the data into the appropriate business protocol format. It must also populate the required fields in the Event that is processed by e*Xchange.

This component is entirely user-defined and must be added to the e*Xchange Schema. The type of component to use depends on whether a connection to a system outside e*Gate must be made, and if so, what type of system. Typically, this component is an e*Way that connects to a business application, such as SAP, that sends out electronic messages. These messages may or may not be in the format required by the trading partner to which they are being sent. If the data is not in the correct format, the e*Way must translate the data into the required format before it is sent to the e*Xchange system for enveloping and forwarding to the trading partner.

The e*Xchange Internal_Order_Feeder e*Way

The e*Xchange example simulates sending the health claim message from the internal system.

Follow these steps to configure **Internal_Order_Feeder** e*Way.

- **Step 1: Create and Configure the Internal_Order_Feeder e*Way** on page 75
- **Step 2: Create the Internal_Order_Feeder Collaboration Rule Script** on page 76
- **Step3: Create the Internal_Order_Feeder Collaboration Rule** on page 77
- **Step 4: Create the Internal_Order_Feeder Collaboration** on page 77

Step 1: Create and Configure the Internal_Order_Feeder e*Way

To create and configure the Internal_Order_Feed e*Way

1. Create a new e*Way named **Internal_Order_Feeder**.
2. Open the **e*Way Properties** dialog box and, in the **Executable file** area of the **General** tab, browse for **stcewfile.exe**.
3. In the **e*Way Properties** dialog box, in the **Configuration file** area of the **General** tab, click **New**.
4. Configure the **Internal_Order_Feeder** e*Way parameters using Table 43.

Table 43 Internal_Order_Feeder e*Way Parameters

Screen	Parameter	Setting
General Settings	(All)	(Default)
Outbound (send) settings	(All)	(Default)
Poller (inbound) settings	PollDirectory	<eGate>\data\hipaa\internal\input
	MultipleRecordsPerFile	NO
	(All others)	(Default)

Table 43 Internal_Order_Feeder e*Way Parameters

Screen	Parameter	Setting
Performance Testing	(All)	(Default)

5 When finished editing the e*Way configuration file, save your work and close the e*Way Editor.

6 Click **OK** to close the **e*Way Properties** dialog box.

Step 2: Create the Internal_Order_Feeder Collaboration Rule Script

The **Internal_Order_Feeder.tsc** Collaboration Rule script does the following:

- Converts the message to base 64 encoding, and copies it to the Payload node of the TP_EVENT section of the e*Xchange standard Event.
- Copies "O" for outbound to the direction node of the TP_EVENT section.
- Copies the trading partner logical name "Insurance" to the PartnerName node of the TP_EVENT section.

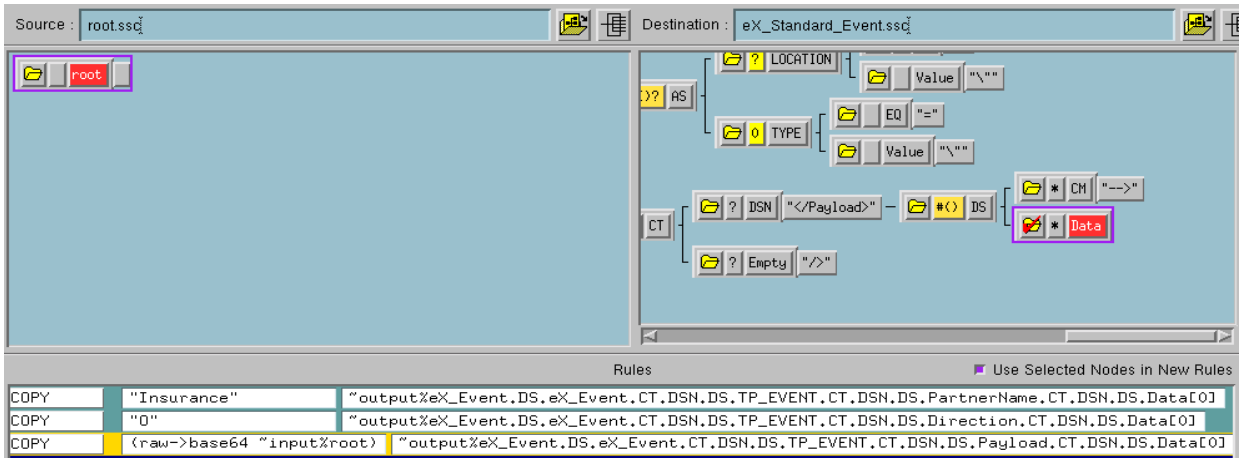
To create and configure the Internal_Order_Feeder Collaboration Rule Script

1 Open the Collaboration Editor.

2 Create a new Collaboration Rules script named **Internal_Order_Feeder.tsc**. The Source Event Type Definition is **root.ssc**. The Destination Event Type Definition is **eX_Standard_Event.ssc**.

3 Add the rules shown at the bottom of Figure 7.

Figure 7 Internal_Order_Feeder.tsc



4 Save the Collaboration Rules script and close the Collaboration Editor.

Step3: Create the Internal_Order_Feeder Collaboration Rule

Once the Collaboration Rule script has been created, you must set up the Collaboration and Collaboration Rules properties for the **Internal_Order_Feeder** component in the Enterprise Manager GUI.

To create and configure the Internal_Order_Feeder Collaboration Rule

1. Create a new Collaboration Rule named **Internal_Order_Feeder**.
2. Open the **Internal_Order_Feeder Collaboration Rule Properties** dialog box, and then select the **General** tab. Configure as shown in Table 44.

Table 44 Internal_Order_Feeder Collaboration Rule Configuration - General Tab

Section	Value
Service	Monk
Collaboration Rules	monk_scripts\common\Internal_Order_Feeder.tsc
Initialization File	monk_scripts\common\load_ext

Important: *To use the Monk function **raw->base64**, you must make sure the file containing this function has been loaded.*

3. Select the **Subscriptions** tab. Select **eX_External_Evt** and move it to the right pane.
4. Select the **Publications** tab. Select **eX_to_ePM** and move it to the right pane.
5. Click **OK** to save the Collaboration Rule and close the properties dialog.

Step 4: Create the Internal_Order_Feeder Collaboration

The **Internal_Order_Feeder** Collaboration must prepare the data coming into the e*Xchange system. The complexity of this task depends on the state of the data before the **Internal_Order_Feeder** Collaboration processes it.

The **Internal_Order_Feeder** Collaboration must do the following:

- Put the data into the appropriate EDI format.
- Convert the data to base 64 encoding.
- Populate the required nodes in the Event sent to e*Xchange for processing.

To create and configure the Internal_Order_Feeder Collaboration

1. Select the **Internal_Order_Feeder** e*Way.
2. Create a new Collaboration named **Internal_Order_Feeder**.
3. Open the **Internal_Order_Feeder Collaboration Properties** dialog box, and then configure the properties using the values specified in Table 45.

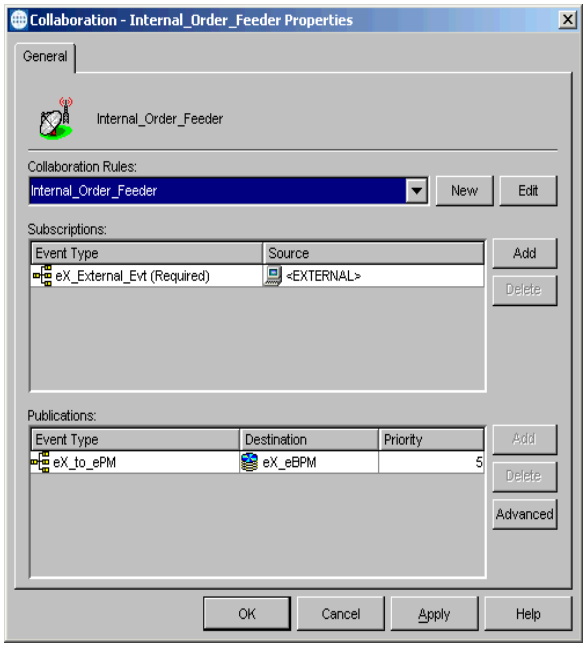
Table 45 Internal_Order_Feeder Collaboration Configuration

Section	Value
Collaboration Rules	Internal_Order_Feeder

Section	Value
Subscriptions	Event Type: eX_External_Evt Source: <EXTERNAL>
Publications	Event Type: eX_to_ePM Destination: eX_eBPM

Verify the information in the **Collaboration - Internal Order Feeder Properties** dialog box as shown in Figure 8.

Figure 8 Internal_Order_Feeder Collaboration Properties



4 Click **OK** to save the properties information and close the properties dialog box.

6.8 Configure the eX_ePM e*Way

The **eX_ePM** e*Way requires only minimal configuration. You must give it the logon information for the e*Xchange database.

To configure the eX_ePM configuration file

1 Open the **eX_ePM e*Way Properties** dialog box, and then select the **General** tab.

2 In the **Configuration File** area, click **Edit**.

3 Configure the parameters as shown in Table 46.

Table 46 eX_ePM e*Way Parameters

Screen	Parameter	Setting
General Settings	(All)	(Default)
Communication Setup	(All)	(Default)
Monk Configuration	(All)	(Default)
Database Setup	Database Name	(service name of the e*Xchange database)
	User name	(administrator user login ID)
	Password	(administrator user password)
	(All others)	(Default)

4 Save the configuration settings and close the **Edit Settings** dialog box.

6.9 Configure the eX_Poll_Receive_FTP e*Way

Although the **eX_Poll_Receive_FTP** e*Way does not appear in Figure 3, it is used to send information to the **eX_Batch_from_Trading_Partner** e*Way.

The **eX_Poll_Receive_FTP** e*Way requires only minimal configuration. You must give it the logon information for the e*Xchange database.

To configure the eX_Poll_Receive_FTP configuration file

1 Open the **eX_Poll_Receive_FTP e*Way Properties** dialog box, and then select the **General** tab.

2 In the **Configuration File** area, click **Edit**.

3 Configure the parameters as shown in Table 47.

Table 47 eX_Poll_Receive_FTP e*Way Parameters

Screen	Parameter	Setting
General Settings	(All)	(Default)
Communication Setup	(All)	(Default)
Monk Configuration	(All)	(Default)
Database Setup	Database Name	(service name of the e*Xchange database)
	User name	(administrator user login ID)
	Password	(administrator user password)
	(All others)	(Default)

4 Save the configuration settings and close the **Edit Settings** dialog box.

6.10 Running the Scenario

Running the scenario performs two functions:

1 Sends the health claim to the trading partner.

2 Processes the health claim payment sent from the trading partner.

Before you run the scenario, make sure you have performed steps 1 through 3 under **"Installing the Sample Files" on page 62**. This ensures your data files are in the correct locations.

To process the Health Claim message

1 Start the Control Broker. At the command line, enter:

```
stccb.exe -rh localhost -rs HealthClaim -ln localhost_cb -un
Administrator -up STC
```

2 Open the e*Gate Monitor. Select the HealthClaim Schema.

3 Start the **ewHipaaValidation** e*Way.

4 Start the **Internal_Order_Feeder** e*Way.

This e*Way retrieves the health care claim message and sends it to e*Xchange.

5 Start the **eX_ePM** e*Way.

6 Rename **<eGate>\data\hipaa\internal\input\hipaa-837.~in** to **hipaa-837.fin**.

The file is renamed **hipaa-837.~in** as it is picked up.

7 Start the **eX_Batch_to_Trading_Partner** e*Way.

This e*Way sends the message to the trading partner.

8 Look in the **<egate>\data\hipaa\TP\output** folder. The file **output0.dat** appears.

9 Start the **eX_Poll_Receive_FTP** e*Way.

This e*Way sends configuration information to the **eX_Batch_from_Trading_Partner** e*Way.

10 Start the **eX_Batch_from_Trading_Partner** e*Way.

This e*Way retrieves messages from the trading partner.

11 Start the **Internal_Order_Eater** e*Way.

This e*Way sends the message to the internal system.

12 Rename **<egate>\data\hipaa\TP\input\hipaa-835.dat.backup** to **hipaa-835.dat**.

This sends the health claim payment advice message from the trading partner. The file is renamed **hipaa-835.dat.backup** as it is picked up.

13 Look in the **<egate>\data\hipaa\internal\output** folder. The file **output_order1.dat** appears.

That completes the first part of the exercise. You can view the results in Message Tracking in the e*Xchange Partner Manager Web Interface.

Viewing the Results in Message Tracking

You can view the results of the message processing by using the Message Tracking feature of e*Xchange.

Message Tracking shows two entries for the incoming message. This is because an acknowledgment can be sent out immediately, and a response message is sent out later. These two responses to the trading partner are tracked separately.

To view the inbound message in Message Tracking

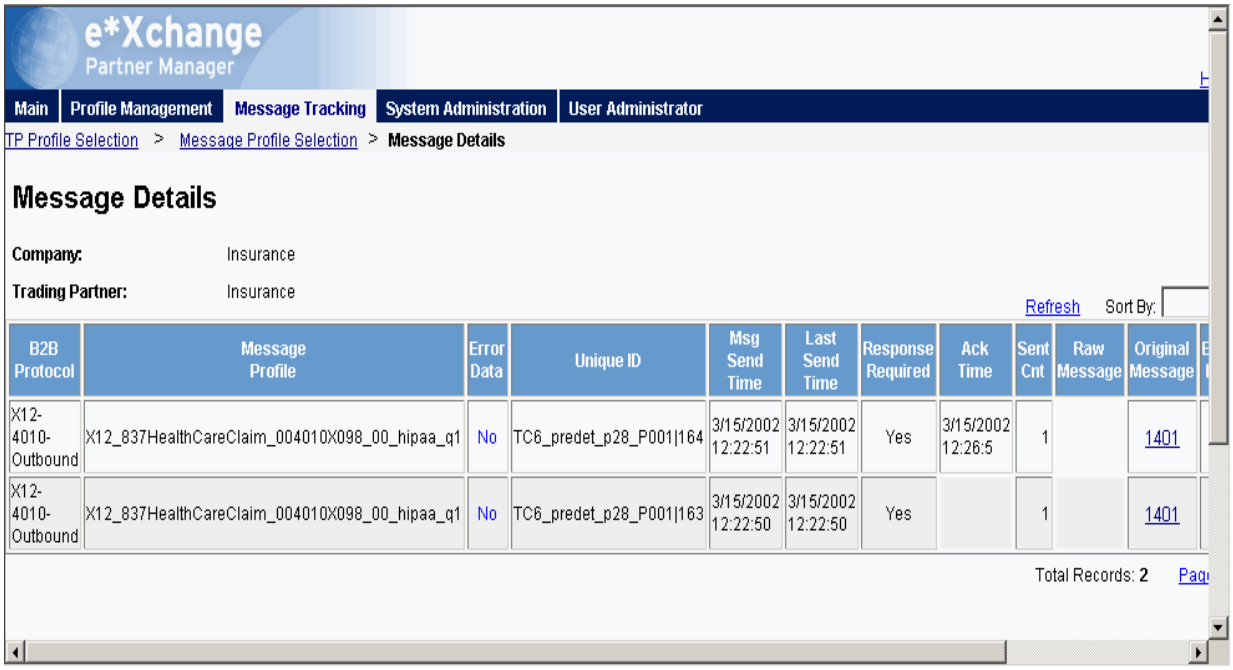
1. From the **Main** page of the e*Xchange Web interface, select **Message Tracking**.

   The **TP Profile Selection** page appears.

2. In the **Company Profile** field, select **Insurance**.
3. In the **Trading Partner Profile** field, select **Insurance**.
4. In the **eBusiness Protocol** field, select **X12**.
5. In the **Direction** field, select **Outbound**.
6. Click the **Message Profile Selection**.
7. Select the **X12_837HealthCareClaim_004010X098_00_hipaa_q1** message.
8. Click the **Message Details** link to view the resulting list.

   The results are shown in Figure 9.

Figure 9 Message Tracking: Outbound



As shown in Figure 9, e*Xchange records two entries for the message. The top entry is for the original message, for which a response message will be sent. The second entry is for the acknowledgment message.

For one entry, the **Ack Message** column has a link to the message information. Click it to view the acknowledgment message.

Chapter 7

e*Gate Implementation

This chapter discusses the steps involved to create an implementation that converts HIPAA X12 data to or from XML.

7.1 Overview

The proposed e*Gate solution makes use of the e*Gate Java Collaboration Service to transform the data from the System A format to the System B format. e*Gate is very flexible about where the actual transformation processing can occur as the data moves from System A to System B. This solution uses the Multi-Mode e*Way as the main transformation component and two Java Pass Through file e*Ways to bring data into and send data out from the e*Gate system. Figure 10 shows all the components and their relationships to one another in the complete e*Gate Schema.

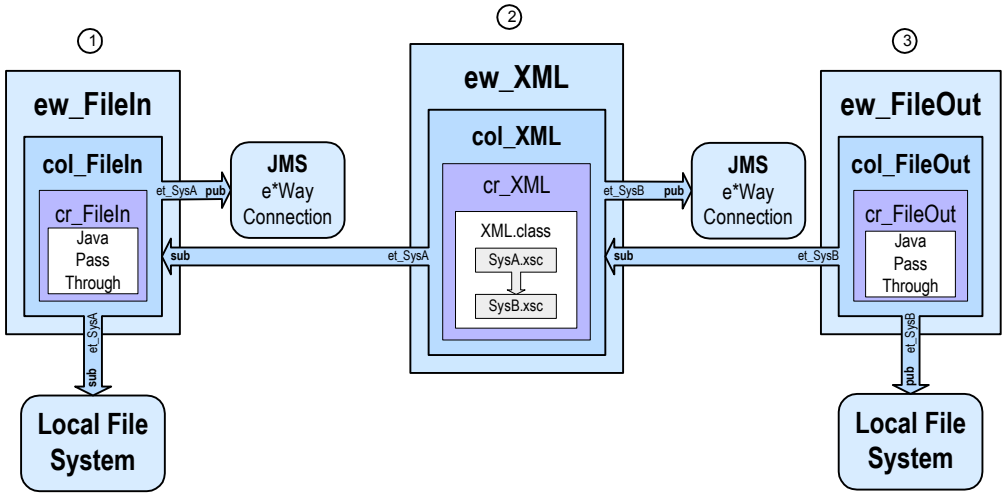
The major steps for implementing the e*Gate solution are as follows:

1. Verify the e*Gate installation.
2. Create a new Schema.
3. Create the Event Types and Java ETDs.
4. Create the Collaboration Rules.
5. Add and configure the e*Ways and the JMS e*Way Connection.
6. Add and define the Collaborations that route the data.
7. Test the Scenario.

7.1.1. The e*Gate XML Scenario

By examining Figure 10 you notice that the Schema components are created and configured from the inside out. That is, the Event Types and Collaborations are created before creating the e*Ways that use them. This method has the advantage of letting you create all the components of the same type at the same time. It also ensures that the required components are available when you need them.

Figure 10 XML Scenario Overview



Notes on the XML Scenario Overview

① **ew_FileIn** brings data from System A into e*Gate.

The **col_FileIn** Collaboration in the **ew_FileIn** e*Way subscribes to a location on the local file system. It polls this location for a text file with extension **".fin"** containing data from System A. Then it reads the message, packages the data as an **et_SysA** Event, and publishes the Event to the JMS e*Way Connection.

② **ew_XML** changes the data format.

The **col_XML** Collaboration in the **ew_XML** e*Way subscribes to **et_SysA** Events published by **col_FileIn**. It uses the Java Collaboration Rule **cr_XML** to convert to XML. This rule uses the **XML.class** which implements the transformation. Finally, **col_XML** publishes the **et_SysB** Event to the JMS e*Way Connection.

③ **ew_FileOut** writes the transformed data out to a local file system.

The **col_FileOut** Collaboration in the **ew_FileOut** e*Way subscribes to **et_SysB** Events published by a JMS e*Way Connection. The **cr_FileOut** Collaboration Rule uses the Java Pass Through service to move the data without modifying it. When an **et_SysB** Event is retrieved, the e*Way packages it as a text file and writes it to the specified location on the local file system, completing the end-to-end scenario.

7.2 Verify the e*Gate Installation

This end-to-end scenario is designed to run on a single machine. Before beginning the configuration process, you must verify that you have all the required software installed on the target machine. In addition to a standard e*Gate installation, you need to install the Multi-mode e*Way and the Java HIPAA ETD Library. Refer to the *e*Gate Integrator Installation Guide* for e*Gate system requirements and instructions to install the e*Gate components.

7.3 Create a New Schema

To create a new Schema

1. Start the e*Gate Enterprise Manager and log in as **Administrator** (or another user with administrator privileges) to the appropriate Registry Host.
2. In the **Open Schema on Registry Host** dialog box, click **New**.
3. In the **Enter New Schema Name** box, type **HIPAA**, and then click **Open**.

   The Enterprise Manager opens and displays the new **HIPAA** Schema.
4. At the bottom of the navigator (left) pane, click the **Components** tab.

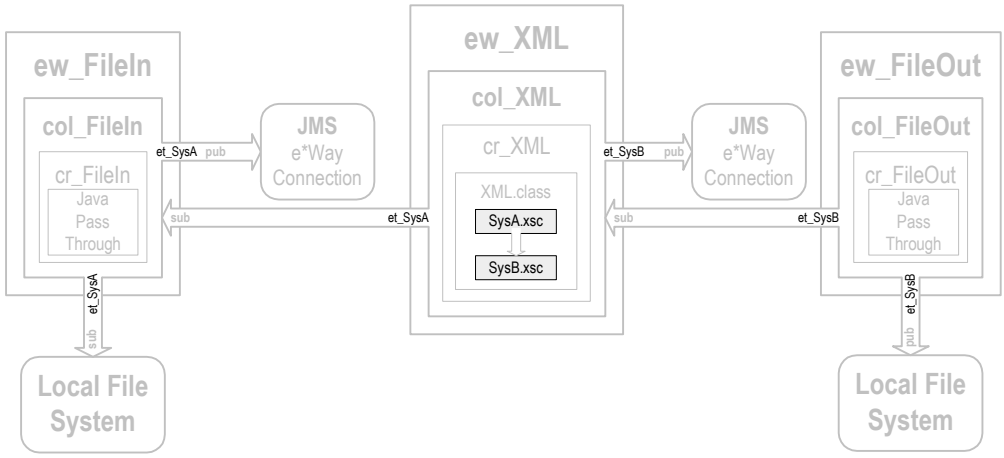
   You will perform all configuration steps in the **Components** tab.

7.4 Create the Event Types and Java ETDs

This scenario uses two Event Types. The first Event Type, **et_SysA**, models the ASCII format of the data received from System A. The second Event Type, **et_SysB**, models the XML format required by System B.

Figure 11 shows where these parts fit into the collection of interrelated components that make up the finished Schema.

Figure 11 Event Types and Java ETDs



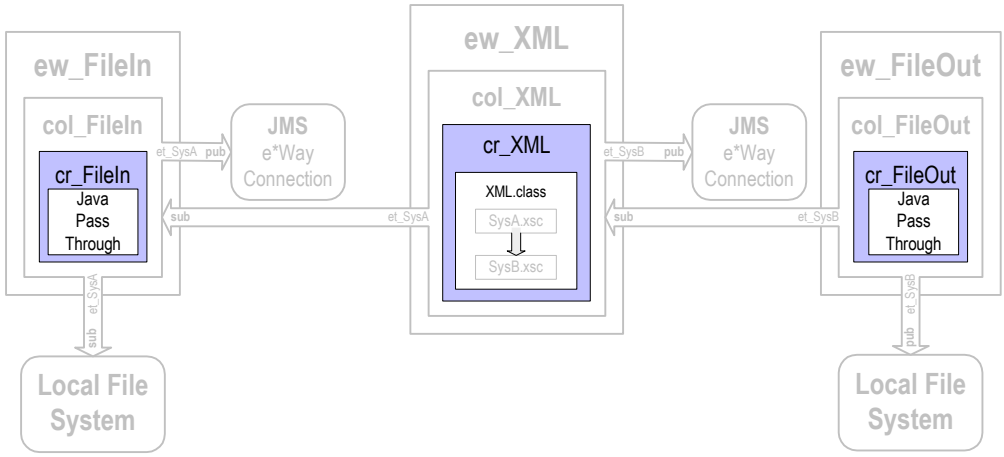
7.5 Create the Collaboration Rules

This scenario uses three Collaboration Rules: two Java Pass Through rules and one Java Collaboration. The Java Pass Through rules, **cr_FileIn** and **cr_FileOut**, are used to route

the Events through the e*Gate system and the Java Collaboration Rule **cr_XML** is used to transform the Event from Event Type **et_SysA** to Event Type **et_SysB**.

Figure 12 shows where these parts fit into the collection of interrelated components that make up the finished Schema.

Figure 12 Collaboration Rules



7.5.1. Create the Java Pass Through Collaborations

The Java Pass Through Collaborations are used to bring data into and take data away from the e*Gate system. The following procedure explains how to create the Java Pass Through Collaborations used in this scenario.

To create the cr_FileIn and cr_FileOut Collaboration Rules

1. In the **Navigator** pane of the e*Gate Enterprise Manager, click the **Collaboration Rules** folder.
2. On the **File** menu, point to **New**, and then click **Collaboration Rules**.
3. In the **New Collaboration Rules Component** dialog box, type **cr_FileIn** for the Collaboration Rule name, and then click **OK**.

   cr_FileIn is added to the list of Collaboration Rules in the e*Gate Enterprise Manager **Editor** pane.
4. On the list of Collaboration Rules, double-click **cr_FileIn**.
5. In the **Collaboration Rules** section, click **Find** and navigate to **collaboration_rules\STCLibrary**, and then double-click **STCJavaPassThrough.class**.

   The path to **STCJavaPassThrough.class** appears in the **Collaboration Rules** section of the dialog box, and the path to **STCJavaPassThrough.ctl** appears in the **Initialization File** section. The **STCJavaPassThrough.class** file configures the Collaboration Mapping Instances for you. You are not required to make any other changes to **cr_FileIn**.

6 Click **OK** to close the **Collaboration Rules - cr_FileIn Properties** dialog box.

7 Repeat steps 2 through 6 to create the **cr_FileOut** Collaboration Rule. Substitute **cr_FileOut** for the Collaboration Rule name.

7.5.2. Create the Java Collaboration Rule

The procedure for creating a Collaboration Rule that uses the Java Collaboration Service is different from creating other e*Gate Collaboration Rules. Use the following procedure to start the Java Collaboration Editor and create the Java Collaboration Rule used by this scenario.

To create cr_XML and start the Java Collaboration Editor

1 Use steps 1 through 4 from the procedure described in **"To create the cr_FileIn and cr_FileOut Collaboration Rules" on page 86** to create a new Collaboration Rule named **cr_XML**.

2 Click the **Collaboration Mapping** tab, and then click **Add Instance**.

An instance row is added to the **Collaboration Mapping** tab.

3 In the **Instance Name** column, type **In** for the instance name.

4 Click **Find**, and explore to **etd\templates\Hipaa_2000**, and then double-click **X12_004010X096_00_hipaaQ3_837_HealCareClai.xsc**.

X12_004010X096_00_hipaaQ3_837_HealCareClai.xsc is added to the **ETD** column of the instance row. You are not required to make any other changes to **In**.

5 Add another ETD instance. Use **Out** for the instance name.

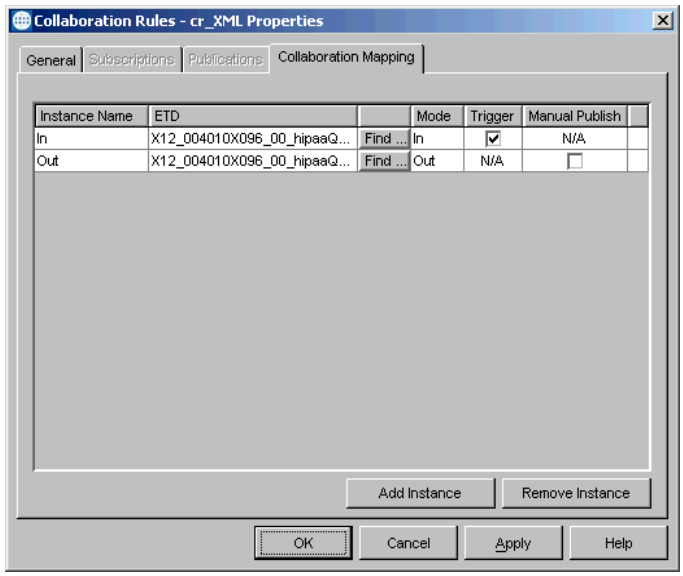
Important: *The Java ETD instance names must be unique per Schema.*

6 Find and select **X12_004010X096_00_hipaaQ3_837_HealCareClai.xsc** as the ETD for **Out**.

7 Click in the **Mode** cell in the row for **Out,** and then click **Out**.

You do not need to make any other changes to **Out**. The completed **Collaboration Mapping** tab looks like the one shown in Figure 13.

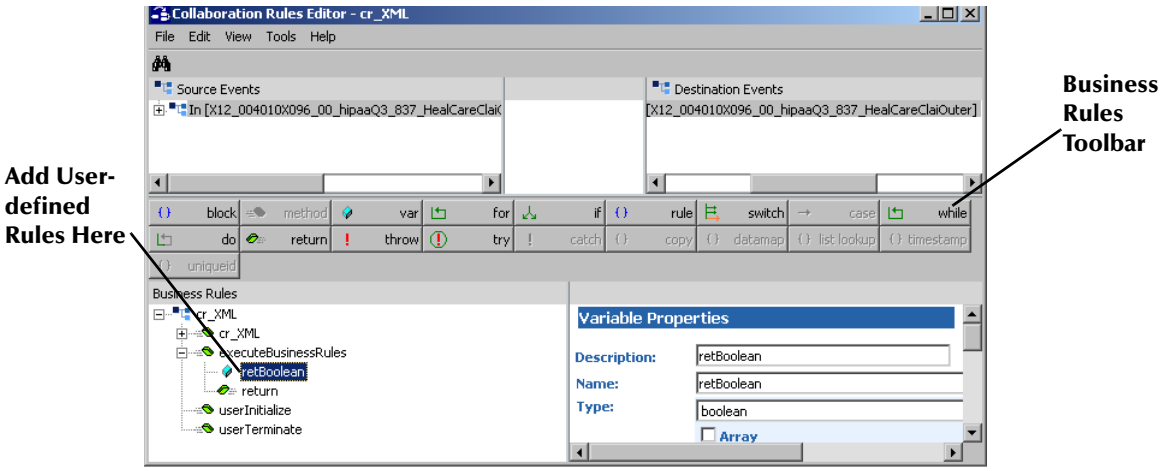
Figure 13 Completed Collaboration Mapping Tab



8 Click the **General** tab, and then in the **Collaboration Rules** area click **New**.

The Java Collaboration Editor opens a new Collaboration Rule with **In** (SysA) as the source Event and **Out** (SysB) as the destination Event as shown in Figure 14.

Figure 14 XML Before Adding User-Defined Code



To create cr_XML.class in the Java Collaboration Rules Editor

1 On the **View** menu, click **Display Code**.

The **Business Rules** pane now displays the Java code in addition to the code labels.

2 In the **Source Events** and **Destination Events** panes, expand **In** and **Out** to display the leaf nodes of the ETDs.

3 In the **Business Rules** pane, under the **executeBusinessRules** method, click the **retBoolean** variable.

All the user-defined rules you add for this scenario are added within the **executeBusinessRules** method, and are placed between the **retBoolean** variable and the **return** rule (see Figure 14).

4 With the **retBoolean** variable selected, drag the **ISA_InteContHead** segment node from the **Source Events** pane to the **ISA_InteContHead** segment node in the **Destination Events** pane.

5 With the Source **X12_004010X096_00_hipaaQ3_837_HealCareClaiInner** segment selected, click the **For** button on the **Business Rules** toolbar.

A **For** rule is added to the **executeBusinessRules** method.

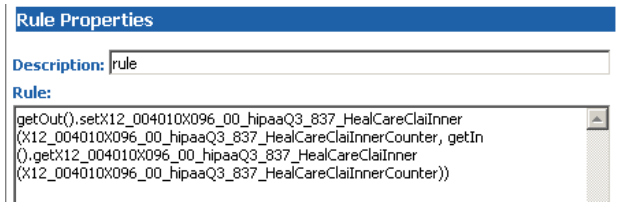
6 With the **For** rule selected in the **Business Rules** pane, do the following:

A Drag the **X12_004010X096_00_hipaaQ3_837_HealCareClaiInner** segment in the **Source Events** pane to the **X12_004010X096_00_hipaaQ3_837_HealCareClaiInner** segment in the **Destination Events** pane.

B Ensure that the repetition instance for both source and destination is **X12_004010X096_00_hipaaQ3_837_HealCareClaiInnerCounter**.

C The **Sibling or Child** dialog appears. Click **Child**.

Figure 15 Completed Rule



7 With the **For** rule selected, drag the **IEA_InteContTrai** segment node from the **Source Events** pane to the **IEA_InteContTrai** segment node in the **Destination Events** pane.

8 With the rule created in step 7 selected, click the **Rule** button on the **Business Rules** toolbar.

A **Rule** is added to the **executeBusinessRules** method.

9 Drag the **setXMLOutput** rule from the **Destination Events** pane to the **Rules** box.

The **Parameter for method: setXMLOutput()** dialog appears.

10 Enter **true**, and click **OK**.

11 On the **File** menu, click **Save**.

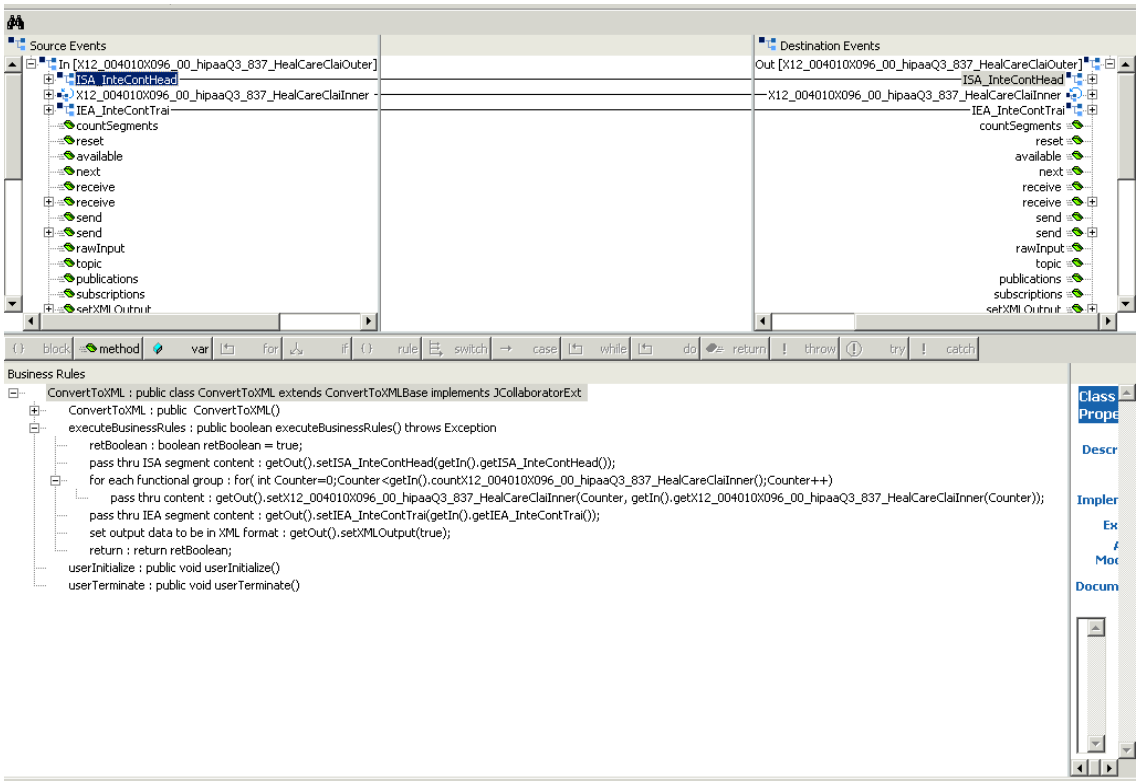
12 In the **Save** dialog box, navigate to the **collaboration_rules** folder, and then save the **cr_XML.xpr** file.

13 On the **File** menu, click **Compile**.

The Java source code is compiled. When the compiler is finished "Compile Completed" is displayed in the **Compile/Debug** pane. The **Compile/Debug** pane

also displays any errors generated by the compilation process. Clear any errors before you continue.

Figure 16 ConvertToXML After Adding User-Defined Code



14 On the **File** menu, click **Exit**.

You may be prompted to save changes. The Java Collaboration Editor closes and in the **Collaboration Rules - cr_XML Properties** dialog box, **collaboration_rules\cr_XML.class** appears in the **Collaboration Rules** box and **collaboration_rules\cr_XML.ctl** appears in the **Initialization file** box.

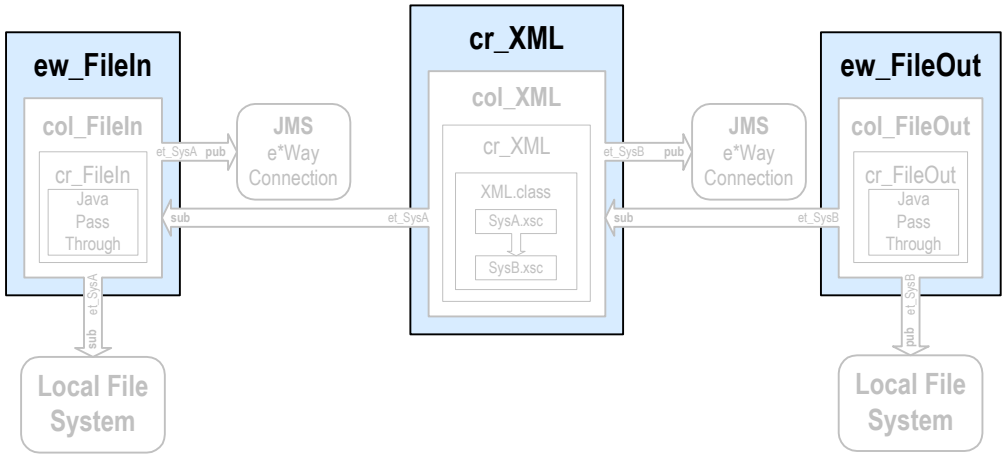
15 Click **OK** to close the **Collaboration Rules - cr_XML Properties** dialog box.

7.6 Add the e*Ways and e*Way Connection

After you have created your ETDs and Collaborations, you are ready to add and configure the e*Gate components that use these parts.

Figure 17 highlights the components added in this step.

Figure 17 e*Ways and JMS e*Way Connection



7.6.1. Add and Configure the File e*Ways

For the e*Gate XML scenario, you must create two file e*Ways, **ew_FileIn** and **ew_FileOut**, to simulate bringing data in from System A and sending it out to System B.

To add and configure the ew_FileIn file e*Way

1. In the e*Gate Enterprise Manager, in the **Navigator** pane, click the Control Broker (***hostname*_cb**).
2. On the **File** menu, point to **New**, point to **Module**, and then click **e*Way**.
3. In the **New e*Way Component** dialog box, type **ew_FileIn** for the e*Way name, and then click **OK**.

   The **ew_FileIn** e*Way is added to the Schema.
4. Right-click **ew_FileIn**, and then click **Properties**.
5. In the **e*Way - ew_FileIn Properties** dialog box, in the **Executable file** area, click **Find**.
6. In the **File Selection** dialog box, browse for and double-click the file **stcewfile.exe**.

   The **bin\stcewfile.exe** file is added as the executable file, causing the component to become a file e*Way.
7. In the **Configuration file** area, click **New**.

   The e*Way Configuration File Editor opens with a default file e*Way configuration file ready for editing.
8. In the **Goto Section** list, click **Poller (inbound) settings**.
9. In the **PollDirectory** box, type **C:\eGate\Client\Data\HIPAA** and then press **ENTER**.

C:\eGate\Client\Data\HIPAA is added as the directory to be polled to the **PollDirectory** list. No other changes are necessary to the **ew_FileIn** e*Way's configuration file.

10 On the **File** menu, click **Save**.

11 In the **Save As** dialog box, click **Save** to accept the default filename (**ew_FileIn.cfg**) and save the file.

12 On the **File** menu, click **Close** to quit the e*Way Configuration File Editor.

The **configs\stcewfile\ew_FileIn.cfg** file is added to the **Configuration file** area in the **e*Way - ew_FileIn Properties** dialog box.

13 Click the **Start Up** tab, and then select the **Start automatically** check box.

14 Click **OK** to close the **e*Way - ew_FileIn Properties** dialog box.

To add and configure the ew_FileOut file e*Way

Adding the **ew_FileOut** e*Way follows the same general procedure as that outlined for adding the **ew_FileIn** e*Way above.

1 Use steps 1 through 7 from **"To add and configure the ew_FileIn file e*Way" on page 91** to add another file e*Way named **ew_FileOut** and open its configuration file for editing.

2 In the e*Way Configuration File Editor, in **General Settings**, click **NO** for **AllowIncoming**, and **YES** for **AllowOutgoing**.

3 In the **Goto Section** list, click **Outbound (send) settings**.

4 Add **C:\eGate\Client\Data\HIPAA** as the default **OutputDirectory**.

5 Add **HIPAAoutput%d.dat** as the default **OutputFileName**.

No other changes are necessary to the **ew_FileOut** e*Way's configuration file.

6 On the **File** menu, click **Save**.

7 In the **Save As** dialog box, click **Save** to accept the default file name (**ew_FileOut.cfg**) and save the file.

8 On the **File** menu, click **Close** to quit the e*Way Configuration File Editor.

9 Click the **Start Up** tab, and then select the **Start automatically** check box.

10 Click **OK** to close the **e*Way - ew_FileOut Properties** dialog box.

7.6.2. Add the Multi-Mode e*Way

To add the multi-mode e*Way

1 In the e*Gate Enterprise Manager, in the **Navigator** pane, click the Control Broker (***hostname_cb***).

2 On the **File** menu, point to **New**, point to **Module**, and then click **e*Way**.

3 In the **New e*Way Component** dialog box, type **ew_XML** for the e*Way name, and then click **OK**.

The **ew_XML** e*Way is added to the Schema.

4 Right-click the **ew_XML** e*Way in the **Editor** pane, and then click **Properties**.

5 In the **Configuration file** area, click **New**.

The e*Way Configuration File Editor opens with a default Multi-Mode e*Way configuration file.

6 Scroll to the bottom of the **JVM Settings** parameters and click **Remote debugging port number**.

7 In the **Remote debugging port number** box, type **8000**, and then press **ENTER**.

8000 is listed as the **Remote debugging port number**. No other changes are necessary to the **ew_XML** e*Way's configuration file.

Important: *In-schema debugging must be enabled on the Participating Host for this to work. See the e*Gate Integrator Installation Guide for more information.*

8 On the **File** menu, click **Save**.

9 In the **Save As** dialog box, click **Save** to accept the default filename (**ew_XML.cfg**) and save the file.

10 On the **File** menu, click **Close** to quit the e*Way Configuration File Editor.

11 Click the **Start Up** tab, and then select the **Start automatically** check box.

12 Click **OK** to close the **e*Way - ew_XML Properties** dialog box.

7.6.3. Configure the IQ Manager

To configure the IQ Manager

1 In the e*Gate Enterprise Manager, in the **Navigator** pane, double-click the IQ manager (***hostname*_iqmgr**).

2 In the **Configuration File** area, click **New**.

3 On the **File** menu, click **Save**, and then click **Save** again to accept the default name.

4 On the **File** menu, click **Close** to quite the editor.

5 Click the **Start Up** tab, select the **Start automatically** check box, and then click **OK**.

7.6.4. Add the JMS e*Way Connection

To add the JMS e*Way Connection

1 In the **Navigator** pane of the e*Gate Enterprise Manager, click the **e*Way Connections** folder.

2 In the **Editor** pane, right-click, and then click **New e*Way Connection**.

3 In the **New e*Way Connection Component** dialog box, type **JMS** for the e*Way Connection name, and then click **OK**.

JMS is added to the list of e*Way Connections.

4 In the editor pane, double-click **JMS**.

The **e*Way Connection - JMS Properties** dialog box displays.

5 From the **e*Way Connection Type** drop-down list, select **SeeBeyond JMS**.

6 In the **Configuration File** area, click **New**.

7 From the **JMS IQ Manager** drop-down list, select your IQ Manager.

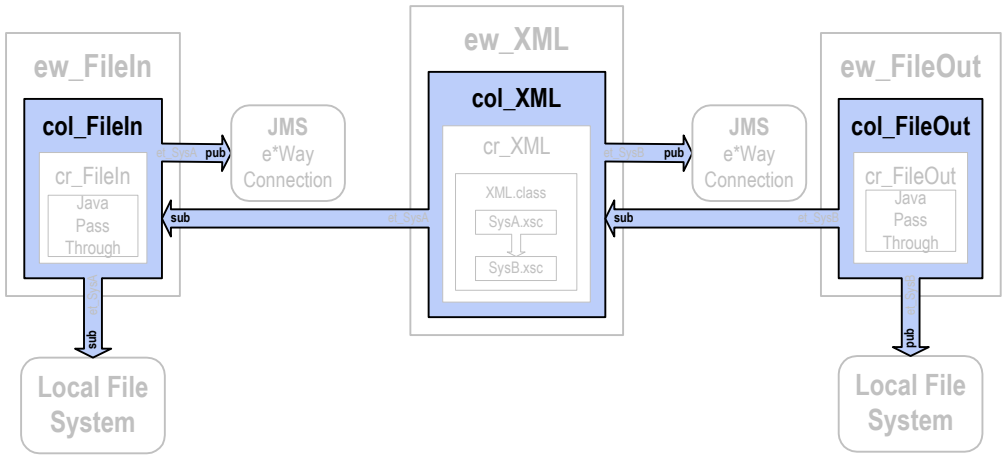
8 Click **OK** to save your configuration file.

9 Click **OK** to close the **e*Way Connection - JMS Properties** dialog box.

7.7 Add the Collaborations that Route the Data

The e*Ways in this example use Collaborations to route data through the e*Gate system. Typically, the Collaborations are configured in upstream to downstream order. Figure 18 shows the relationships of the Collaborations to the remaining components that make up the complete Schema.

Figure 18 Collaborations Showing Publish and Subscribe Relationships



7.7.1. Add and Configure col_FileIn

The **col_FileIn** Collaboration brings the data into the e*Gate system from the specified data file.

To add and configure col_FileIn

1 In the e*Gate Enterprise Manager, in the **Navigator** pane, click the **ew_FileIn** e*Way.

2 On the **File** menu, point to **New**, and then click **Collaboration**.

3 In the **New Collaboration Component** dialog box, type **col_FileIn** for the Collaboration name, and then click **OK**.

4 In the editor pane, double-click **col_FileIn**.

The **Collaboration - col_FileIn Properties** dialog box displays.

5 In the **Collaboration Rules** list, click **cr_FileIn**.

6 In the **Subscriptions** area, click **Add**.

A row is added to the **Subscriptions** box.

7 In the **Instance Name** column, select **JavaPassThroughIn**. In the **Event Type** column, click **et_SysA** on the list, and then in the **Source** column, select **<EXTERNAL>** from the list.

8 In the **Publications** area, click **Add**.

A row is added to the **Publications** box.

9 In the **Instance Name** column, select **JavaPassThroughOut**. In the **Event Type** column, click **et_SysA** on the list, and then in the **Destination** column, select **JMS** from the list.

10 Click **OK** to close the **Collaboration - col_FileIn Properties** dialog box.

7.7.2. Add and Configure col_XML

The **col_XML** Collaboration changes the data from the **et_SysA** Event Type to the **et_SysB** Event Type.

To add and configure col_XML

1 Use steps 1 through 4 from **"Add and Configure col_FileIn" on page 94** to add a Collaboration to the **ew_XML** e*Way named **col_XML** and open its properties dialog box.

2 In the **Collaboration Rules** list, click **cr_XML**.

3 In the **Subscriptions** area, click **Add**.

A row is added to the **Subscriptions** box.

4 Double-click in the **Instance Name** column and click **In** on the list.

5 Double-click in the **Event Type** column and click **et_SysA** on the list.

6 Double-click in the **Source** column and click **JMS** on the list.

7 In the **Publications** area, click **Add**.

A row is added to the **Publications** area.

8 Double-click in the **Instance Name** column, and then click **Out** on the list.

9 Double-click in the **Event Type** column, and then click **et_SysB** on the list.

10 Double-click in the **Destination** column, and then click **JMS** on the list.

11 Click **OK** to close the **Collaboration - col_XML Properties** dialog box.

7.7.3. Add and Configure col_FileOut

The **col_FileOut** Collaboration sends the transformed data out of the e*Gate system. Use the following procedure to add and configure **col_FileOut**.

To add and configure col_FileOut

1. Use steps 1 through 4 from **"Add and Configure col_FileIn" on page 94** to add a Collaboration to the **ew_FileOut** e*Way named **col_FileOut** and open its properties dialog box.
2. In the **Collaboration Rules** list, click **cr_FileOut**.
3. In the **Subscriptions** area, click **Add**.
4. A row is added to the **Subscriptions** box.
5. In the **Instance Name** column, select **JavaPassThroughIn**. In the **Event Type** column, click **et_SysB** on the list, and then in the **Source** column, select **JMS** from the list.
6. In the **Publications** area, click **Add**.

   A row is added to the **Publications** area.
7. In the **Instance Name** column, select **JavaPassThroughOut**. In the **Event Type** column, click **et_SysB** on the list, and in the **Destination** column, select **<EXTERNAL>** from the list.
8. Click **OK** to close the **Collaboration - col_FileOut Properties** dialog box.

7.8 Test the Scenario

Testing the scenario includes the following steps:

1. Review the complete Schema.
2. Test the Schema.
3. Troubleshoot any problems.

7.8.1. Review the Complete Schema

Table 48 lists all the components for the Schema. Check all the settings. Substitute the name of the machine running the Schema for *hostname* where applicable.

Table 48 HIPAA Components

Component	Logical Name	Settings
Schema	HIPAA	
Control Broker	*hostname*_cb	
IQ Manager	*hostname*_iqmgr	Service = SeeBeyond JMS Config file = *hostname*_iqmgr.cfg Start Up = Auto
Event Type	et_SysA	SysA.xsc
	et_SysB	SysB.xsc

Table 48 HIPAA Components

Component	Logical Name	Settings
Java ETD	SysA.xsc	Package Name = SysApackage
	SysB.xsc	Package Name = SysBpackage
Collaboration Rule	cr_FileIn	Service = Java JavaPassThroughIn GenericInEvent.xsc Trigger JavaPassThroughOut GenericOutEvent.xsc
	cr_XML	Service = Java In SysA.xsc In Trigger Out SysB.xsc Out
	cr_FileOut	Service = Java JavaPassThroughIn GenericInEvent.xsc Trigger JavaPassThroughOut GenericOutEvent.xsc
Java Collaboration Rule	cr_XML.class	Source = In Destination = Out
Inbound e*Way	ew_FileIn	Executable = stcewfile.exe Config file = ew_FileIn.cfg Start Up = Auto Collaboration = col_FileIn
Outbound e*Way	ew_FileOut	Executable = stcewfile.exe Config file = ew_FileOut.cfg Start Up = Auto Collaboration = col_FileOut
Multi-Mode e*Way	ew_XML	Executable = stceway.exe Config file = ew_XML.cfg Start Up = Auto Collaboration = col_XML
JMS e*Way Connection	JMS	Service = SeeBeyond JMS Config file = jms*hostname*_iqmgr.cfg
Collaboration	col_FileIn	Collab Rule = cr_FileIn Subscription = et_SysA from <EXTERNAL> Publication = et_SysA to JMS
	col_XML	Collab Rule = cr_XML Subscription = et_SysA from JMS Publication = et_SysB to JMS
	col_FileOut	Collab Rule = cr_FileOut Subscription = et_SysB from JMS Publication = et_SysB to <EXTERNAL>

7.8.2. Test the Schema

Test the scenario by sending data into the system and verifying the output.

Start the Schema

Begin the test by starting the Schema using the **stccb.exe** command. Monitor the components from the e*Gate Monitor.

To start the Schema

1. Use the following command to start the Control Broker from a command line.

   ```
   stccb.exe -rh hostname -rs HIPAA -ln hostname_cb -un username -up
   password
   ```

2. Start the e*Gate Monitor.
3. Verify that all the components in the Schema are running.

Testing in Windows 2000

1. Once all the scenario components have been started successfully, use Windows Explorer to navigate to **c:\eGate\client\data\HIPAA**.
2. Change the file extension on the input file **HIPAAinput.txt** to **.fin**.
3. Click **Yes** to confirm this choice.
4. Verify that the extension changes to **.~in** indicating that the **ew_FileIn** e*Way has retrieved the file.
5. Almost immediately, the output file, **HIPAAoutput#.dat**, should appear in the directory, indicating a successful conclusion to the test.

Figure 19 shows a section of the original HIPAA X12 data, and Figure 20 shows a section of the converted HIPAA XML data.

Figure 19 Original data

```
ISA*00*          *00*          *01*6264712000     *01*6264716000     *010126*1709*U*0
00003231 8*0*T*:~GS*HC*901234572000*908887732000*010126*1709*32318*T*004010X096~ST*837
32318~BHT*0019*00*Hipaa_012601_WO2*20010126*1615*CH~REF*87*3920394930203~NM1*41*1
*JOHNSON*BARBARA*T***46*901234591834 1~PER*IC*ARTHUR JONES*ED*(614)555-1212*ED*(614)55
1212*EM*(614)555-1212~NM1*40*2*SMITH*****46*111222333~HL*1**20*1~PRV*BI*ZZ*
1234567890098765432 1768958473~CUR*85*USA~NM1*85*2*JONES*****24*43202~N3*PO BOX 123*15
WEST 57TH STREET~N4*CINCINNATI*OH*43017*US~REF*0B*500~REF*06*3920394930203~PER*IC*MAG
```

Figure 20 Converted data


```
- <envelope format="X12">
  - <segment code="ISA" name="Interchange Control Header">
    - <element code="I01" name="Authorization Information Qualifier">
        <value>00</value>
      </element>
    - <element code="I02" name="Authorization Information">
        <value />
      </element>
    - <element code="I03" name="Security Information Qualifier">
        <value>00</value>
      </element>
    - <element code="I04" name="Security Information">
        <value />
      </element>
    - <element code="I05" name="Interchange ID Qualifier">
        <value>01</value>
      </element>
    - <element code="I06" name="Interchange Sender ID">
        <value>6264712000</value>
      </element>
    - <element code="I05" name="Interchange ID Qualifier">
        <value>01</value>
      </element>
    - <element code="I07" name="Interchange Receiver ID">
        <value>6264716000</value>
      </element>
```

Appendix A

ASC X12 Overview

This appendix provides an overview of the X12 standard, including:

- An overview of ASC X12, including the structure of an X12 envelope, data elements, and syntax.
- An explanation of how to use the generic message structures provided as an add-on to e*Gate to help you quickly create the structures you need for various X12 transactions.

For specific information on HIPAA, refer to **Chapter 2**, **"HIPAA Overview" on page 12**.

A.1 Introduction to X12

The following sections provide an introduction to X12.

A.1.1. What Is ASC X12?

ASC X12 is an EDI (electronic data interchange) standard, developed for the electronic exchange of machine-readable information between businesses.

The Accredited Standards Committee (ASC) X12 was chartered by the American National Standards Institute (ANSI) in 1979 to develop uniform standards for interindustry electronic interchange of business transactions—electronic data interchange (EDI). The result was the X12 standard.

The ASC X12 body develops, maintains, interprets, and promotes the proper use of the ASC X12 standard. Data Interchange Standards Association (DISA) publishes the ASC X12 standard and the UN/EDIFACT standard. The ASC X12 body comes together three times a year to develop and maintain EDI standards. Its main objective is to develop standards to facilitate electronic interchange relating to business transactions such as order placement and processing, shipping and receiving information, invoicing, and payment information.

The ASC X12 EDI standard is used for EDI within the United States. UN/EDIFACT is broadly used in Europe and other parts of the world.

X12 was originally intended to handle large batches of transactions. However, it has been extended to encompass real-time processing (transactions sent individually as

they are ready to send, rather than held for batching) for some healthcare transactions to accommodate the healthcare industry.

A.1.2. What Is a Message Structure?

The term *message structure* (also called a transaction set structure) refers to the way in which data elements are organized and related to each other for a particular EDI transaction.

In e*Gate, a message structure is called an Event Type Definition (ETD). Each message structure (ETD) consists of the following:

- Physical hierarchy

  The predefined way in which envelopes, segments, and data elements are organized to describe a particular X12 EDI transaction.

- Delimiters

  The specific predefined characters that are used to mark the beginning and end of envelopes, segments, and data elements.

- Properties

  The characteristics of a data element, such as the length of each element, default values, and indicators that specify attributes of a data element—for example, whether it is required, optional, or repeating.

The transaction set structure of a claim that is sent from a payer to a provider defines the header, trailer, segments, and data elements required by claim transactions. Installation of X12 templates for a specific version includes transaction set structures for each of the transactions available in that version.

e*Xchange Partner Manager uses e*Gate Event Type Definitions, which are based on the X12 message structures, to verify that the data in the messages coming in or going out is in the correct format. There is a message structure for each X12 transaction.

The list of transactions provided is different for each version of X12, and for each customized implementation. This book addresses the transactions covered by the May 1999 and May 2000 implementations of the HIPAA standard.

A.2 Components of an X12 Envelope

X12 messages are all ASCII text, with the exception of the BIN segment which is binary.

Each X12 message is made up of a combination of the following elements:

- Data elements
- Segments
- Loops

Elements are separated by delimiters.

More information on each of these is provided below.

A.2.1. Data Elements

The data element is the smallest named unit of information in the ASC X12 standard. Data elements can be broken down into two types. The distinction between the two is strictly a matter of how they are used. The two types are:

- Simple

  If a data element occurs in a segment outside the defined boundaries of a composite data structure it is called a simple data element.

- Composite

  If a data element occurs as an ordinally positioned member of a composite data structure it is called a composite data element.

Each data element has a unique reference number; it also has a name, description, data type, and minimum and maximum length.

A.2.2. Segments

A segment is a logical grouping of data elements. In X12, the same segment can be used for different purposes. This means that an element's meaning can change based on the segment. For example:

- The NM1 segment is for *any* name (patient, provider, organization, doctor)
- The DTP segment is for *any* date (date of birth, discharge date, coverage period)

For more information on the X12 enveloping segments, refer to **"Structure of an X12 Envelope" on page 103**.

A.2.3. Loops

Loops are sets of repeating ordered segments. In X12 you can locate elements by specifying:

- The transaction set (for example, 270)
- The loop (for example, "loop 1000" or "info. receiver loop")
- The occurrence of the loop
- The segment (for example, BGN)
- The element number (for example, 01)
- The occurrence of the segment (if it is a repeating segment)

A.2.4. Delimiters

In an X12 message, the various delimiters act as syntax, dividing up the different elements of a message. The delimiters used in the message are defined in the

interchange control header, the outermost layer enveloping the message. For this reason, there is flexibility in the delimiters that are used.

No suggested delimiters are recommended as part of the X12 standards, but the industry-specific implementation guides do have recommended delimiters.

The default delimiters used by the SeeBeyond HIPAA ETD Library are the same as those recommended by the industry-specific implementation guides. These delimiters are shown in Table 49.

Table 49 Default Delimiters in X12 ETD Library

Type of Delimiter	Default Value
Segment terminator	~ (tilde)
Data element separator	* (asterisk)
Subelement (component) separator	: (colon)

Within e*Xchange Partner Manager, delimiters are specified at the outer envelope level. The delimiters you define are applied to all transaction types.

If you do not specify delimiters, e*Xchange expects the default delimiters as shown in Table 49.

Note: *It is important to note that errors could result if the transmitted data itself includes any of the characters that have been defined as delimiters. Specifically, the existence of asterisks within transmitted application data is a known issue in X12, and can cause problems with translation.*

A.3 Structure of an X12 Envelope

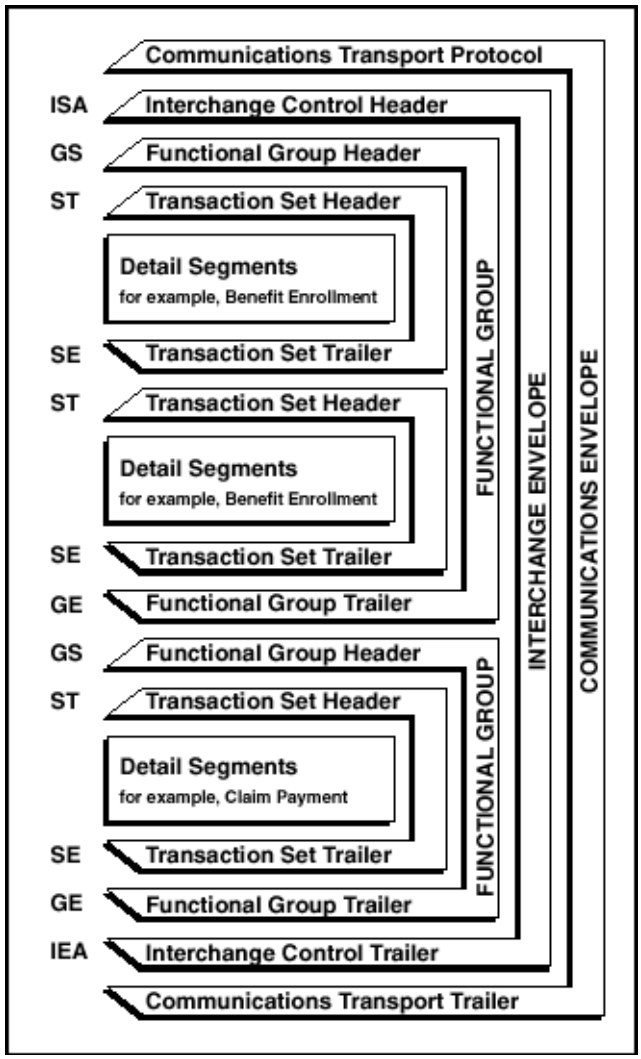
The rules applying to the structure of an X12 envelope are very strict, to ensure the integrity of the data and the efficiency of the information exchange.

The actual X12 message structure has three main levels. From the highest to the lowest they are:

- Interchange Envelope
- Functional Group
- Transaction Set

A schematic of X12 envelopes is shown in Figure 21. Each of these levels is explained in more detail in the following sections.

Figure 21 X12 Envelope Schematic



Note: *The above schematic is from Appendix B of an ASC X12 Implementation Guide.*

Figure 22 shows the standard segment table for an X12 997 (Functional Acknowledgment) as it appears in the X12 standard and in most industry-specific implementation guides.

Figure 22 X12 997 Segment Table

Table 1 - Header

POS. #	SEG. ID	NAME	REQ. DES.	MAX USE	LOOP REPEAT
010	ST	Transaction Set Header	M	1	
020	AK1	Functional Group Response Header	M	1	
		LOOP ID - AK2			999999
030	AK2	Transaction Set Response Header	O	1	
		LOOP ID - AK2/AK3			999999
040	AK3	Data Segment Note	O	1	
050	AK4	Data Element Note	O	99	
060	AK5	Transaction Set Response Trailer	M	1	
070	AK9	Functional Group Response Trailer	M	1	
080	SE	Transaction Set Trailer	M	1	

Figure 23 shows the same transaction as viewed in the Monk ETD Editor in e*Gate.

Figure 23 X12 997 Viewed in Monk ETD Editor

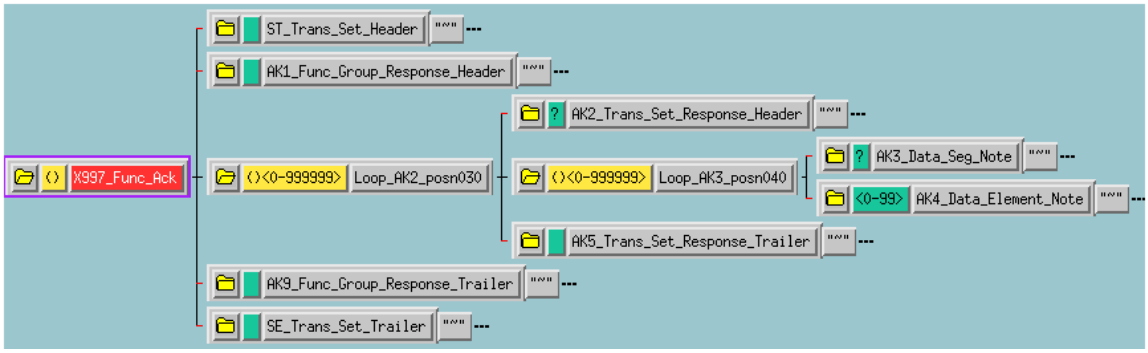


Figure 24 shows the same transaction as viewed in the Java ETD Editor.

Figure 24 X12 997 Viewed in Java ETD Editor

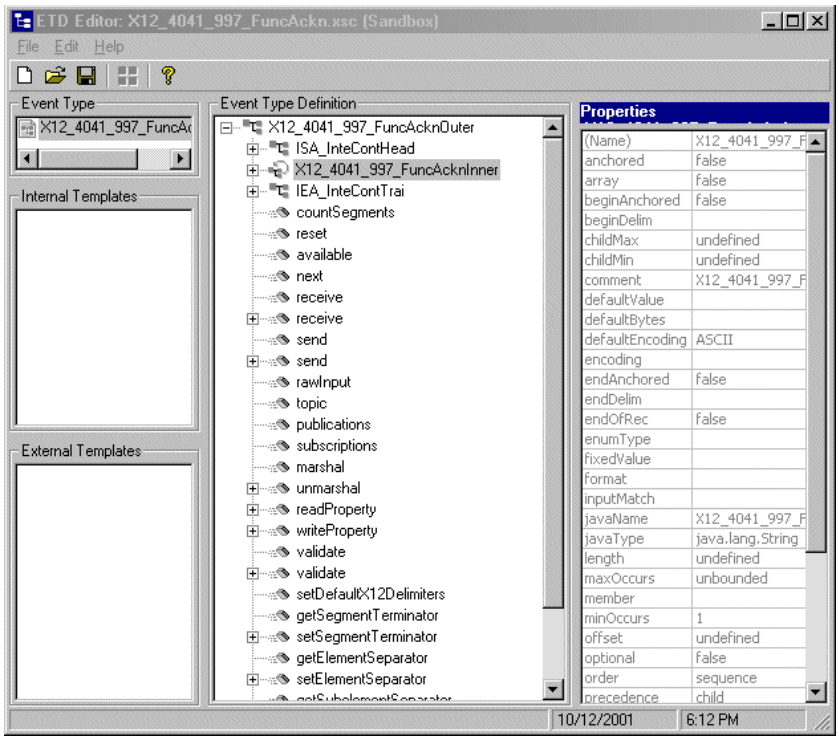


Figure 25 shows an example of a positive 997 acknowledgment, as viewed in the Message Tracking window in the e*Xchange Partner Manager.

Note: *The message shown in Figure 25 was part of a batch and therefore includes only the ST/SE (transaction set) envelope layer.*

Figure 25 Positive 997 (Functional Acknowledgment) Viewed in Message Tracking

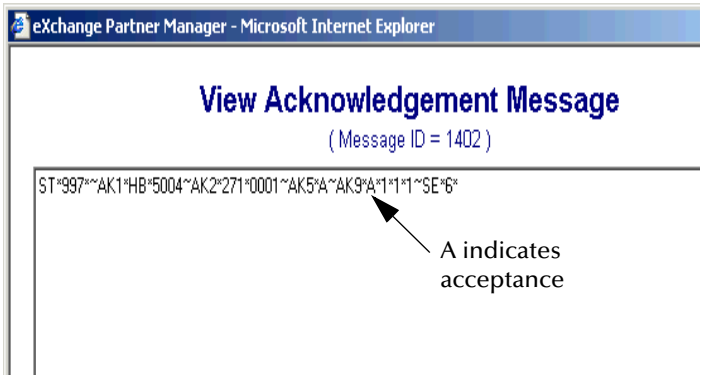
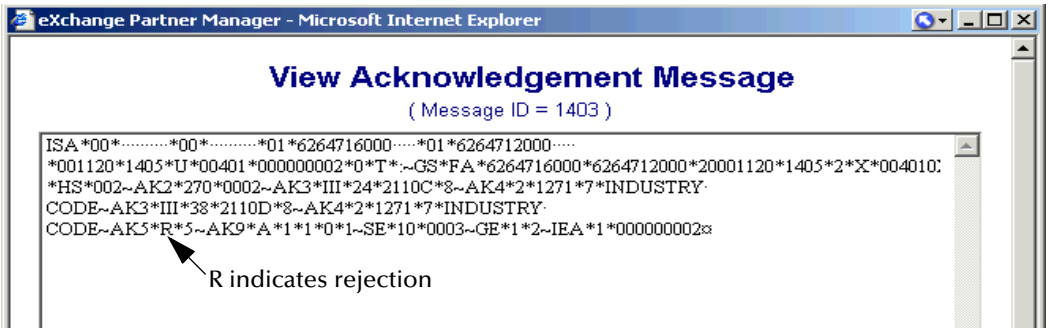


Figure 26 shows an example of a negative 997 acknowledgment, as viewed in the Message Tracking window in the e*Xchange Partner Manager.

Note: *The message shown in Figure 26 is an interactive message and therefore includes all enveloping layers.*

Figure 26 Negative 997 (Functional Acknowledgment) Viewed in Message Tracking



A.3.1. Transaction Set (ST/SE)

Each transaction set (also called a transaction) contains three things:

- A transaction set header
- A transaction set trailer
- A single message, enveloped within the header and footer

The transaction has a three-digit code, a text title, and a two-letter code; for example, **997, Functional Acknowledgment (FA)**.

The transaction is comprised of logically related pieces of information, grouped into units called segments. For example, one segment used in the transaction set might convey the address: city, state, ZIP code, and other geographical information. A transaction set can contain multiple segments. For example, the address segment could be used repeatedly to convey multiple sets of address information.

The X12 standard defines the sequence of segments in the transaction set and also the sequence of elements within each segment. The relationship between segments and elements could be compared to the relationship between records and fields in a database environment.

Figure 27 Example of a Transaction Set Header (ST)

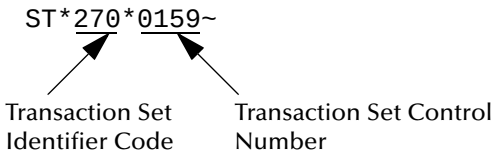
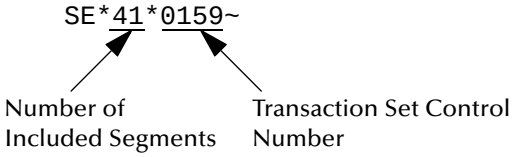


Figure 28 Example of a Transaction Set Trailer (SE)



A.3.2. Functional Group (GS/GE)

A functional group is comprised of one or more transaction sets, all of the same type, that can be batched together in one transmission. The functional group is defined by the header and trailer; the Functional Group Header (GS) appears at the beginning, and the Functional Group Trailer (GE) appears at the end. Many transaction sets can be included in the functional group, but all transactions must be of the same type.

Within the functional group, each transaction set is assigned a functional identifier code, which is the first data element of the header segment. The transaction sets that comprise a specific functional group are identified by this functional ID code.

The functional group header (GS) segment contains the following information:

- Functional ID code (the two-letter transaction code; for example, PO for an 850 Purchase Order, HS for a 270 Eligibility, Coverage, or Benefit Inquiry) to indicate the type of transaction in the functional group
- Identification of sender and receiver
- Control information (the functional group control numbers in the header and trailer segments must be identical)
- Date and time

The functional group trailer (GE) segment contains the following information:

- Number of transaction sets included
- Group control number (originated and maintained by the sender)

Figure 29 Example of a Functional Group Header (GS)

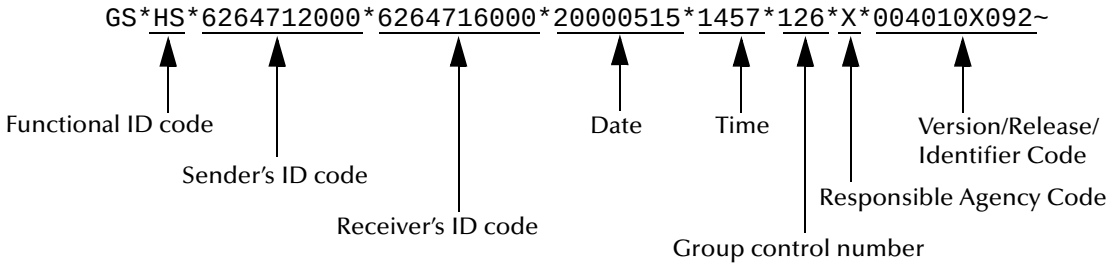
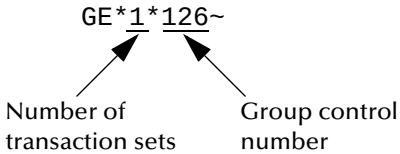


Figure 30 Example of a Functional Group Trailer (GE)



A.3.3. Interchange Envelope (ISA/IEA)

The interchange envelope is the wrapper for all the data to be sent in one batch. It can contain multiple functional groups. This means that transactions of different types can be included in the interchange envelope, with each type of transaction stored in a separate functional group.

The interchange envelope is defined by the header and trailer; the Interchange Control Header (ISA) appears at the beginning, and the Interchange Control Trailer (IEA) appears at the end.

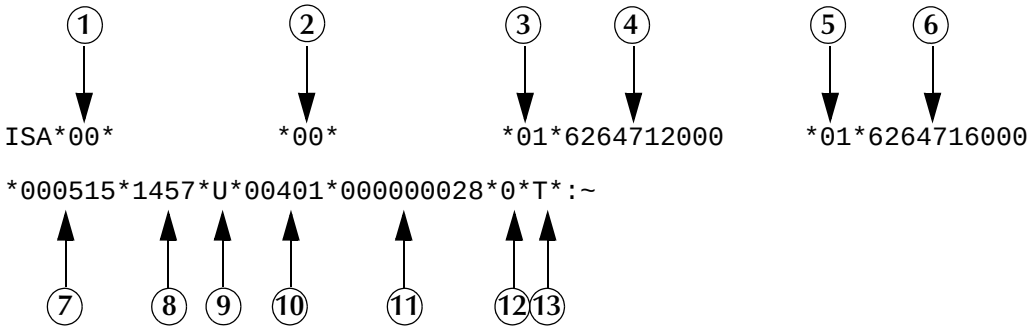
As well as enveloping one or more functional groups, the interchange header and trailer segments include the following information:

- Data element separators and data segment terminator
- Identification of sender and receiver
- Control information (used to verify that the message was correctly received)
- Authorization and security information, if applicable

The sequence of information that is transmitted is as follows:

- Interchange header
- Optional interchange-related control segments
- Actual message information, grouped by transaction type into functional groups
- Interchange trailer

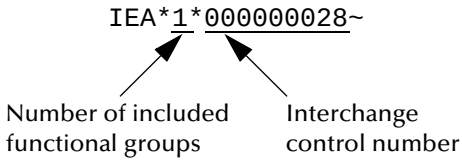
Figure 31 Example of an Interchange Header (ISA)



Interchange Header Segments from Figure 31:

1 Authorization Information Qualifier
2 Security Information Qualifier
3 Interchange ID Qualifier
4 Interchange Sender ID
5 Interchange ID Qualifier
6 Interchange Receiver ID
7 Date
8 Time
9 Repetition Separator
10 Interchange Control Version Number
11 Interchange Control Number
12 Acknowledgment Requested
13 Usage Indicator

Figure 32 Example of an Interchange Trailer (IEA)



A.3.4. Control Numbers

The X12 standard includes a control number for each enveloping layer:

- ISA13—Interchange Control Number
- GS06—Functional Group Control Number
- ST02—Transaction Set Control Number

The control numbers act as identifiers, useful in message identification and tracking. The e*Xchange Partner Manager includes a flag for each control number, so you can choose not to assign control numbers to outgoing messages and not to store control numbers on incoming messages.

ISA13 (Interchange Control Number)

The ISA13 is assigned by the message sender. It must be unique for each interchange. This is the primary means used by e*Xchange Partner Manager to identify an individual interchange.

GS06 (Functional Group Control Number)

The GS06 is assigned by the sender. It must be unique within the Functional Group assigned by the originator for a transaction set.

Note: *The Functional Group control number GS06 in the header must be identical to the same data element in the associated Functional Group trailer, GE02.*

ST02 (Transaction Set Control Number)

The ST02 is assigned by the sender, and is stored in the transaction set header. It must be unique within the Functional Group.

Note: *The control number in ST02 must be identical with the SE02 element in the transaction set trailer, and must be unique within a Functional Group (GS-GE). Once you have defined a value for ST02, e*Xchange Partner Manager uses the same value for SE02.*

A.4 Acknowledgment Types

X12 includes two types of acknowledgment, the TA1 Interchange Acknowledgment and the 997 Functional Acknowledgment.

A.4.1. TA1, Interchange Acknowledgment

The TA1 acknowledgment verifies the interchange envelopes only. The TA1 is a single segment and is unique in the sense that this single segment is transmitted without the GS/GE envelope structures. A TA1 acknowledgment can be included in an interchange with other functional groups and transactions.

A.4.2. 997, Functional Acknowledgment

The 997 includes much more information than the TA1. The 997 was designed to allow trading partners to establish a comprehensive control function as part of the business exchange process.

There is a one-to-one correspondence between a 997 and a functional group. Segments within the 997 identify whether the functional group was accepted or rejected. Data elements that are incorrect can also be identified.

Many EDI implementations have incorporated the acknowledgment process into all of their electronic communications. Typically, the 997 is used as a functional acknowledgment to a functional group that was transmitted previously.

The 997 is the acknowledgment transaction recommended by ASC X12.

The acknowledgment of the receipt of a payment order is an important issue. Most corporate originators want to receive at least a Functional Acknowledgment (997) from the beneficiary of the payment. The 997 is created using the data about the identity and address of the originator found in the ISA and/or GS segments.

Some users argue that the 997 should be used only as a point-to-point acknowledgment and that another transaction set, such as the Application Advice (824) should be used as the end-to-end acknowledgment.

A.4.3. Application Acknowledgments

Application acknowledgments are responses sent from the destination system back to the originating system, acknowledging that the transaction has been successfully or unsuccessfully completed. The application advice (824) is a generic application acknowledgment that can be used in response to any X12 transaction. However, it has to be set up as a response transaction; only TA1 and 997 transactions are sent out automatically.

Other types of responses from the destination system to the originating system, which may also be considered application acknowledgments, are responses to query transactions—for example, the Eligibility Response (271) which is a response to the Eligibility Inquiry (270). Other types of responses from the destination system to the originating system, which may also be considered application acknowledgments, are

responses to query transactions—for example, the Eligibility Response (271) which is a response to the Eligibility Inquiry (270).

A.5 Key Parts of EDI Processing Logic

The five key parts of EDI processing logic are listed in Table 50. The table describes each term, and lists its language analogy along with its associated e*Gate Collaboration scripts.

Table 50 Key Parts of EDI Processing

Term	Description	Language Analogy	e*Gate Collaboration Scripts
structures	format, segments, loops	syntax	ETD files or structures
validations	data contents "edit" rules	semantics	validation scripts
translations (also called mapping)	reformatting or conversion	translation	translation scripts
enveloping	header and trailer segments	envelopes	part of translation
acks	acknowledgments	return receipt	e*Way scripts

e*Gate uses the structures, validations, translations, enveloping, and acknowledgments listed below to support HIPAA.

A.5.1. Structures

The Event Type Definition library for HIPAA includes pre-built ETDs for all supported HIPAA versions.

A.5.2. Validations, Translations, Enveloping, Acknowledgments

e*Gate does not include any pre-built validations, transformations, or acknowledgments. These scripts can be built in either the Monk or Java versions of the Collaboration Rules Editor graphical user interface (GUI). These GUIs provide a user-friendly drag-and-drop front end for creating Monk or Java scripts. For HIPAA, e*Gate provides translations in Monk that will add the enveloping information to the HIPAA message.

Installation of the e*Xchange Partner Manager includes a set of custom Monk validations for HIPAA transactions. It also provides acknowledgments, as described in **"X12 Acknowledgments in e*Xchange Partner Manager" on page 113**.

Note: *In e*Gate, translations are called Collaborations.*

A.5.3. X12 Acknowledgments in e*Xchange Partner Manager

All X12 acknowledgments are automatically handled in e*Xchange Partner Manager. This allows you to configure the transaction set, if any, that is expected as the acknowledgment. e*Xchange Partner Manager can automatically create any type of X12 acknowledgment, including TA1, 997, 824, and transaction-specific acknowledgments.

For more information on X12 acknowledgment types, refer to **"Acknowledgment Types" on page 111**.

A.5.4. Trading Partner Agreements

There are three levels of information that guide the final format of a specific transaction. These three levels are:

- The ASC X12 standard

  ASC X12 publishes a standard structure for each X12 transaction.

- Industry-specific Implementation Guides

  Specific industries publish Implementation Guides customized for that industry. Normally, these are provided as recommendations only. However, in certain cases, it is extremely important to follow these guidelines. Specifically, since HIPAA regulations are law, it is important to follow the guidelines for these transactions closely.

- Trading Partner Agreements

  It is normal for trading partners to have individual agreements that supplement the standard guides. The specific processing of the transactions in each trading partner's individual system might vary between sites. Because of this, additional documentation that provides information about the differences is helpful to the site's trading partners and simplifies implementation. For example, while a certain code might be valid in an implementation guide, a specific trading partner might not use that code in transactions. It would be important to include that information in a trading partner agreement.

A.6 Additional Information

For more information on X12, visit the following Web sites:

- For X12 standard:

  http://www.disa.org

- For Implementation Guides: Washington Publishing Company at

  http://www.wpc-edi.com

Note: *This information is correct at the time of going to press; however, SeeBeyond has no control over these sites. If you find the links are no longer correct, use a search engine to search for* ***X12****.*

Appendix B

HIPAA Files

This appendix provides information on the HIPAA files provided with e*Gate and e*Xchange. For more information on the SeeBeyond HIPAA solution, refer to **Chapter 3**, **"The SeeBeyond Solution" on page 21**.

B.1 e*Xchange Files for HIPAA Transactions

e*Xchange provides Java Collaboration Rules for HIPAA validations, as well as translation files for use with e*Gate. Java ETDs for HIPAA transactions are provided in the e*Gate HIPAA ETD Library for both X12 and NCPDP transactions. Using the HIPPA X12 ETDs in combination with the Java Collaboration Rules provides comprehensive validations for HIPAA transactions.

Note: *The HIPAA solution also requires the X12 4010 ETD Library to work properly.*

B.1.1. HIPAA e*Xchange Validation Collaboration Rules Files

While most of the validations of HIPAA rules are performed in the JAVA HIPAA ETDs, e*Xchange provides a set of Collaboration Rules files to provide additional validations. These files are only provided for the May 2000 release of the implementation guides, and are designed for use with the May 2000 HIPAA ETD Library files. The Collaboration Rules files are located in:

- **\<eGate>\server\registry\repository\default\collaboration_rules\HIPAA**

The HIPAA Collaboration Rules are stored in the HIPAA validation e*Way, **ewHipaaValidation**. The files included with e*Xchange version 4.5.3 are listed in Table 51.

Table 51 HIPAA Collaboration Rules (May 2000) Provided with e*Xchange

File Name	Transaction
val_X12_004010X092_00_hipaa270_EligCoveOrBeneInqu	270 (Eligibility Coverage or Benefit Inquiry)
val_X12_004010X092_00_hipaa271_EligCoveOrBeneInfo	271 (Eligibility Coverage or Benefit Information)
val_X12_004010X093_00_hipaa276_HealCareClaiStatRequ	276 (Health Care Claim Status Request)

Table 51 HIPAA Collaboration Rules (May 2000) Provided with e*Xchange (Continued)

File Name	Transaction
val_X12_004010X093_00_hipaa277_HealCareClaiStatNoti	277 (Health Care Claim Status Notification)
val_X12_004010X094_00_hipaaA1_278_HealCareServReviI	278 (Health Care Services Review Information: Request for Review)
val_X12_004010X094_00_hipaaA3_278_HealCareServReviI	278 (Health Care Services Review Information: Response to Request)
validate_X12_004010X061_00_hipaa820_PaymOrdeAdvi	820 (Payment Order Remittance Advice)
validate_X12_004010X095_00_hipaa834_BeneEnroAndMain	834 (Benefit Enrollment and Maintenance)
validate_X12_004010X091_00_hipaa835_HealCareClaiPaym	835 (Health Care Claim Payment Advice)
validate_X12_004010X096_00_hipaaQ3_837_HealCareClai	837 (Health Care Claim: Professional)
validate_X12_004010X097_00_hipaaQ2_837_HealCareClai	837 (Health Care Claim: Dental)
validate_X12_004010X098_00_hipaaQ1_837_HealCareClai	837 (Health Care Claim: Institutional)

B.1.2. HIPAA e*Xchange Files for e*Gate

The following files are provided with e*Xchange, but they include the GS/GE and ISA/IEA enveloping and are suitable for use outside e*Xchange when a complete Event structure is required. For example, they are appropriate when using e*Gate to translate from X12 to a business application's proprietary data format.

These files are stored in the following location:

- **\<eGate>\server\registry\repository\default\monk_scripts\eXchange\HIPAA**

The file names have "_xlate" (for May 1999 files) or "_xlat" (for May 2000 files) appended to the file name to indicate that these are the translation files and include the interchange control and functional group header and footer. The May 2000 files have "00" in their file names to further distinguish them from the May 1999 files.

May 1999 Files

The May 1999 HIPAA transaction files for e*Gate that are included with e*Xchange version 4.5.3 are listed in Table 52.

Table 52 HIPAA Transactions (May 1999) Provided with e*Xchange for e*Gate

File Name	Transaction
X12_270EligibCoverageBenefitInquiry_004010X092_hipaa_xlate	270 (Eligibility Coverage or Benefit Inquiry)
X12_271EligibCoverageBenefitInfo_004010X092_hipaa_xlate	271 (Eligibility Coverage or Benefit Information)
X12_276HealthCareClaimStatusRequest_004010X093_hipaa_xlate	276 (Health Care Claim Status Request)
X12_277HealthCareClaimStatusNotification_004010X093_hipaa_xlate	277 (Health Care Claim Status Notification)
X12_278HealthCareServicesReviewInfo_004010X094_hipaa_a1_xlate	278 (Health Care Services Review Information: Request for Review)
X12_278HealthCareServicesReviewInfo_004010X094_hipaa_a3_xlate	278 (Health Care Services Review Information: Response to Request)
X12_820PaymentOrderRemittanceAdvice_004010X061_hipaa_xlate	820 (Payment Order Remittance Advice)
X12_834BenefitEnrollmentandMaint_004010X095_hipaa_xlate	834 (Benefit Enrollment and Maintenance)
X12_835HealthCareClaimPaymentAdvice_004010X091_hipaa_xlate	835 (Health Care Claim Payment Advice)
X12_837HealthCareClaim_004010X098_hipaa_q1_xlate	837 (Health Care Claim: Professional)
X12_837HealthCareClaim_004010X097_hipaa_q2_xlate	837 (Health Care Claim: Dental)
X12_837HealthCareClaim_004010X096_hipaae_q3_xlate	837 (Health Care Claim: Institutional)

May 2000 Files

The May 2000 HIPAA transaction files for e*Gate that included with e*Xchange version 4.5.3 are listed in Table 53.

Table 53 HIPAA Transactions (May 2000) Provided with e*Xchange for e*Gate

File Name	Transaction
X12_270EligibCoverageBenefitInquiry_004010X092_00_hipaa_xlat	270 (Eligibility Coverage or Benefit Inquiry)
X12_271EligibCoverageBenefitInfo_004010X092_00_hipaa_xlat	271 (Eligibility Coverage or Benefit Information)
X12_276HealthCareClaimStatusRequest_004010X093_00_hipaa_xlat	276 (Health Care Claim Status Request)
X12_277HealthCareClaimStatusNotification_004010X093_00_hipaa_xlat	277 (Health Care Claim Status Notification)

Table 53 HIPAA Transactions (May 2000) Provided with e*Xchange for e*Gate

File Name	Transaction
X12_278HealthCareServicesReviewInfo_004010X094_00_hipaa_a1_xlat	278 (Health Care Services Review Information: Request for Review)
X12_278HealthCareServicesReviewInfo_004010X094_00_hipaa_a3_xlat	278 (Health Care Services Review Information: Response to Request)
X12_820PaymentOrderRemittanceAdvice_004010X061_00_hipaa_xlat	820 (Payment Order Remittance Advice)
X12_834BenefitEnrollmentandMaint_004010X095_00_hipaa_xlat	834 (Benefit Enrollment and Maintenance)
X12_835HealthCareClaimPaymentAdvice_004010X091_00_hipaa_xlat	835 (Health Care Claim Payment Advice)
X12_837HealthCareClaim_004010X096_00_hipaa_q3_xlat	837 (Health Care Claim: Professional)
X12_837HealthCareClaim_004010X097_00_hipaa_q2_xlat	837 (Health Care Claim: Dental)
X12_837HealthCareClaim_004010X098_00_hipaa_q1_xlat	837 (Health Care Claim: Institutional)

B.2 e*Gate Files for HIPAA Transactions

e*Gate provides HIPAA ETD libraries for both HIPAA X12 transactions and HIPAA NCPDP transactions. The May 2000 HIPAA X12 ETDs are designed for use with the HIPAA Collaboration Rules of e*Xchange.

X12 HIPAA ETDs

The X12 portion of the HIPAA ETD Library provides Java Event Type Definitions (**.xsc** and **.jar** files) for all nine standard X12 transactions that have been adopted by HIPAA, as listed in Table 54 and Table 55. These ETDs are stored in the following locations:

- **\<eGate>\server\registry\repository\default\etd\templates\Hipaa_1999**
- **\<eGate>\server\registry\repository\default\etd\templates\Hipaa_2000**

These transactions are based on the October 1997 X12 standard; that is, Version 4, Release 1, Sub-release 0 (004010) (version 4010).

Table 54 HIPAA 1999 Java X12 ETD Files

File	Transaction Name
X12_004010X092_99_hipaa270_EligCoveOrBeneInqu	270 (Eligibility Coverage or Benefit Inquiry)

Table 54 HIPAA 1999 Java X12 ETD Files

File	Transaction Name
X12_004010X092_99_hipaa271_EligCoveOrBeneInfo	271 (Eligibility Coverage or Benefit Information)
X12_004010X093_99_hipaa276_HealCareClaiStatRequ	276 (Health Care Claim Status Request)
X12_004010X093_99_hipaa277_HealCareClaiStatNoti	277 (Health Care Claim Status Notification)
X12_004010X094_99_hipaaA1_278_HealCareServReviInfo X12_004010X094_99_hipaaA3_278_HealCareServReviInfo	278 (Two versions: Health Care Services Review Information and Request for Review/Response to Request)
X12_004010X061_99_hipaa820_PaymOrdeAdvi	820 (Payment Order Remittance Advice)
X12_004010X095_99_hipaa834_BeneEnroAndMain	834 (Benefit Enrollment and Maintenance)
X12_004010X091_99_hipaa835_HealCareClaiPaym	835 (Health Care Claim Payment Advice)
X12_004010X098_99_hipaaQ1_837_HealCareClai X12_004010X097_99_hipaaQ2_837_HealCareClai X12_004010X096_99_hipaaQ3_837_HealCareClai	Health Care Claim (three versions: Professional, Dental, and Institutional)

Table 55 HIPAA 2000 Java X12 ETD Files

File	Transaction Name
X12_004010X092_00_hipaa270_EligCoveOrBeneInqu	270 (Eligibility Coverage or Benefit Inquiry)
X12_004010X092_00_hipaa271_EligCoveOrBeneInfo	271 (Eligibility Coverage or Benefit Information)
X12_004010X093_00_hipaa276_HealCareClaiStatRequ	276 (Health Care Claim Status Request)
X12_004010X093_00_hipaa277_HealCareClaiStatNoti	277 (Health Care Claim Status Notification)
X12_004010X094_00_hipaaA1_278_HealCareServReviInfo X12_004010X094_00_hipaaA3_278_HealCareServReviInfo	278 (Two versions: Health Care Services Review Information and Request for Review/Response to Request)
X12_004010X061_00_hipaa820_PaymOrdeAdvi	820 (Payment Order Remittance Advice)
X12_004010X095_00_hipaa834_BeneEnroAndMain	834 (Benefit Enrollment and Maintenance)
X12_004010X091_00_hipaa835_HealCareClaiPaym	835 (Health Care Claim Payment Advice)
X12_004010X098_00_hipaaQ1_837_HealCareClai X12_004010X097_00_hipaaQ2_837_HealCareClai X12_004010X096_00_hipaaQ3_837_HealCareClai	Health Care Claim (three versions: Professional, Dental, and Institutional)
X12ValidationResult	Specialized validation output ETD containing the error message structure

NCPDP HIPAA ETDs

The NCPDP portion of the HIPAA ETD Library provides request and response transactions for all the HIPAA-approved NCPDP transaction codes, as listed in Table 56. These ETDs are stored in:

- **\<eGate>\server\registry\repository\default\etd\templates\NCPDP\Telecom_5_1**

Table 56 NCPDP-HIPAA ETD Files for Telecom 5.1

Code	Transaction Name
NCPDP_T51_E1_REQ_EligVeriRequ NCPDP_T51_E1_RESP_EligResp	E1 (Eligibility Verification)
NCPDP_T51_B1_REQ_BillRequ NCPDP_T51_B1_RESP_BillResp	B1 (Billing)
NCPDP_T51_B2_REQ_ReveRequ NCPDP_T51_B2_RESP_ReveResp	B2 (Reversal)
NCPDP_T51_B3_REQ_RebiRequ NCPDP_T51_B3_RESP_RebiResp	B3 (Rebill)
NCPDP_T51_P1_REQ_PrioAuthRequAndBillRequ NCPDP_T51_P1_RESP_PrioAuthReqeAndBillResp	P1 (Prior Authorization Request and Billing)
NCPDP_T51_P2_REQ_PrioAuthReveRequ NCPDP_T51_P2_RESP_PrioAuthReveResp	P2 (Prior Authorization Reversal)
NCPDP_T51_P3_REQ_PrioAuthInquRequ NCPDP_T51_P3_RESP_PrioAuthInquResp	P3 (Prior Authorization Inquiry)
NCPDP_T51_P4_REQ_PrioAuthRequOnlyRequ NCPDP_T51_P4_RESP_PrioAuthRequOnlyResp	P4 (Prior Authorization Request Only)
NCPDP_T51_N1_REQ_InfoRepoRequ NCPDP_T51_N1_RESP_InfoRepoResp	N1 (Information Reporting)
NCPDP_T51_N2_REQ_InfoRepoReveRequ NCPDP_T51_N2_RESP_InfoRepoReveResp	N2 (Information Reporting Reversal)
NCPDP_T51_N3_REQ_InfoRepoRebiRequ NCPDP_T51_N3_RESP_InfoRepoRebiResp	N3 (Information Reporting Rebill)
NCPDP_T51_C1_REQ_ContSubsRepoRequ NCPDP_T51_C1_RESP_ContSubsRepoResp	C1 (Controlled Substance Reporting)
NCPDP_T51_C2_REQ_ContSubsRepoReveRequ NCPDP_T51_C2_RESP_ContSubsRepoReveResp	C2 (Controlled Substance Reporting Reversal)
NCPDP_T51_C3_REQ_ContSubsRepoRebiRequ NCPDP_T51_C3_RESP_ContSubsRepoRebiResp	C3 (Controlled Substance Reporting Rebill)

Note: *Because of a limitation in NCPDP, the E1 (Eligibility Verification) Request and Response messages are not associated in Message Tracking (unless you perform customization to circumvent this problem). Because of this limitation, you might see timeout errors on outbound Request messages when the responses have already been received.*

NCPDP Batch 1.1 files for HIPAA are stored in:

- **\<eGate>\server\registry\repository\default\etd\templates\NCPDP\Batch_1_1**

The Batch files are named **NCPDP_Batch11**, and include a Monk ETD (**.ssc**), a Java ETD (**.xsc**), and an executable **.jar** file.

NCPDP Batch 1.0 files for HIPAA are supplied only for backwards compatibility. Batch 1.1 is now the mandated standard for HIPAA transactions. The Batch 1.0 files are stored in:

- **\<eGate>\server\registry\repository\default\etd\templates\NCPDP\Batch_1_0**

The Batch files are named **NCPDP_Batch_1_0**, and include a Monk ETD (**.ssc**), a Java ETD (**.xsc**), and an executable **.jar** file.

Appendix C

Error Codes

This appendix provides information about the error codes used in the e*Xchange validation error messages. It also includes descriptions of the associated error codes in the AK304 and AK403 segments of the 997 Functional Acknowledgment transaction.

C.1 e*Xchange Validation Error Messages

e*Xchange produces an error message for each failed HIPAA validation. Additional information about e*Xchange error messages for HIPAA, including the format of the error message produced, is described under **"Validation Error Reporting" on page 36**.

This appendix describes the error codes and descriptions associated with the validation error messages. The validation error messages list both the e*Xchange errors and the corresponding 997 Functional Acknowledgment errors that appear in the AK304 and AK403 elements. This information can be viewed in Message Tracking in the e*Xchange Web interface.

C.1.1. e*Xchange HIPAA Error Messages

Table 57 below lists all e*Xchange validation message errors, along with the corresponding AK304 and AK403 error codes. The **e*Xchange Error Text** appears in element 11 of the e*Xchange validation error message; the **AK304 Error** appears in element five; and the **AK403 Error** appears in element nine. Detailed information about AK304 and AK403 errors is provided in Table 59 and Table 60, respectively. The variable *[supplied value]* is used in the error text to indicate that the actual text is specific to the element and condition for which the error occurred.

Table 57 e*Xchange Error Messages

e*Xchange Error Text	AK304 Error	AK403 Error
Invalid date or time. Should match format *[supplied value]*		8
Value can only be *[supplied value]*		7
Value cannot be found in code source *[supplied value]*		7
Value cannot be *[supplied value]*		7
Usage should be *[supplied value]*	varies	varies

Table 57 e*Xchange Error Messages

e*Xchange Error Text	AK304 Error	AK403 Error
The field length should be *[supplied value]*		4
The field length should be at least *[supplied value]*		4
The field length should be at most *[supplied value]*		5
The value should be in format *[supplied value]*		6
The value should be equal to *[supplied value]*		6
The value should not be equal to *[supplied value]*		6
The value should match *[supplied value]*		6
The value should not match *[supplied value]*		6
The value must begin with 1 and be incremented by 1		7
The value must be unique within transaction set		7
The formula does not balance		7
HL Hierarchy ID Number should be unique. HL01 usually starts with the number one and is incremented by one for each subsequent HL segment within the transaction.		7
Root (information source) level HL segment should not be subordinate to any HL segment.		3
Can not find its parent HL segment with the Hierarchical ID Number *[supplied value]*		3
Can not find its subordinate HL segment		7
Find a subordinate HL segment with the Hierarchical ID Number *[supplied value]*		7
The value must be unique among elements *[supplied value]*		7
[Syntax rule R-Required: One or More] Conditional required data element missing		2
[Syntax rule P-Paired: All or None] Conditional required data element missing		2
[Syntax rule C-Conditional: If first, then all] Conditional required data element missing		2
[Syntax rule E-Exclusion: One or None] Exclusion Condition Violated		10
[Syntax rule L-List Conditional: If first, then at least one] Conditional required data element missing		2
Antlr.RecognitionException encountered (during unmarshalling). Exception *[supplied value]*	2	
Antlr.TokenStreamException encountered (during unmarshalling). Exception *[supplied value]*	1	
Loop occurrence is more than maximum occurrence (during unmarshalling). Loop occurrence *[supplied value]*	4	

Table 57 e*Xchange Error Messages

e*Xchange Error Text	AK304 Error	AK403 Error
Segment occurrence is more than maximum occurrence (during unmarshalling). Segment occurrence *[supplied value]*	5	
Loop occurrence is less than minimum occurrence (during unmarshalling). Loop occurrence *[supplied value]*	3	
Segment occurrence is less than minimum occurrence (during unmarshalling). Segment occurrence *[supplied value]*	3	
Loop occurrence is more than maximum occurrence. Loop occurrence *[supplied value]*	4	
Segment occurrence is more than maximum occurrence. Segment occurrence *[supplied value]*	5	
Loop occurrence is less than minimum occurrence. Loop occurrence *[supplied value]*	3	
Segment occurrence is less than minimum occurrence. Segment occurrence *[supplied value]*	3	
Number of data elements inside the segment (during unmarshalling) exceeds *[supplied value]*		3
Number of data subelements inside the composite (during unmarshalling) exceeds *[supplied value]*		3
Data element is required but missing inside the segment (during unmarshalling)		1
Data subelement is required but missing inside the composite (during unmarshalling)		1
Repeatable element's occurrence is more than maximum occurrence (during marshalling). Element occurrence *[supplied value]*		5
Repeatable subelement's occurrence is more than maximum occurrence (during marshalling). Subelement occurrence *[supplied value]*		5
Repeatable element's occurrence is less than minimum occurrence. Element occurrence [supplied value]		4
Repeatable subelement's occurrence is less than minimum occurrence. Subelement occurrence *[supplied value]*		4
Invalid date format (during unmarshalling)		8
Unparsable date (during unmarshalling). Detail: *[supplied value]*		8
Invalid time format (during unmarshalling)		8
Unparsable time (during unmarshalling). Detail: *[supplied value]*		8
Invalid integer (during unmarshalling). Detail: *[supplied value]*		6

Table 57 e*Xchange Error Messages

e*Xchange Error Text	AK304 Error	AK403 Error
Invalid double (during unmarshalling). Detail: *[supplied value]*		6
Found trailing element separator(s) *[supplied value]*		6
Found trailing subelement separator(s) *[supplied value]*		6
Repeatable element's occurrence is more than maximum occurrence. Element occurrence *[supplied value]*		5
Repeatable subelement's occurrence is more than maximum occurrence. Subelement occurrence *[supplied value]*		5
Repeatable element's occurrence is less than minimum occurrence. Element occurrence *[supplied value]*		4
Repeatable subelement's occurrence is less than minimum occurrence. Subelement occurrence *[supplied value]*		4
Data has too many characters of *[supplied value]*		5
Data has too few characters of *[supplied value]*		4
Found character not in the HIPAA basic character set or extended character set: *[supplied value]*		6
Data has unnecessary trailing spaces		6
Data has too many bytes of *[supplied value]*		5
Data has too few bytes of *[supplied value]*		4
Found element marked not-used (during unmarshalling)		3
Found subelement marked not-used (during unmarshalling)		3
Code value is not in the code list of *[supplied value]*		7
Numeric or decimal data contains plus sign '+' at char position of *[supplied value]*		6
Numeric data contains decimal point '.' at char position of *[supplied value]*		6
Decimal data contains decimal point '.' at the rightmost end, which is at char position of *[supplied value]*		6
Numeric or decimal data contains minus sign '-' not at the beginning, but at char position of *[supplied value]*		6
Decimal data contains triad separator ',' at char position of *[supplied value]*		6
In numeric data element type field, found invalid character of *[supplied value]*		6
In decimal data element type field, found invalid character of *[supplied value]*		6
Found unnecessary leading zero(s) in numeric or decimal data field		6

C.1.2. Understanding Error Messages

The error text in Table 57 above refers to *supplied values*, which indicate text that is inserted into the standard error messages to give you information specific to each error. These values use certain abbreviations in the reason clause (that is, the text after "because") to indicate the condition in the rule that did not pass validation. The abbreviations are enclosed in single quotes. The reason clause has three different formats:

- Error location, 'relational abbreviation', constant value

  For example, in the error message "**2^1000A_PER_8 at 6 [(614)555-1212]: The value should be in format EM because 7'EQ'EM**", **7** indicates the element PER7 in loop 1000A, **'EQ'** is the relational abbreviation meaning equals, and **EM** is the value of the PER7 element. This message states that the data, (614)555-1212, is in the wrong format for the qualifier, EM, specified in PER7.

- Error location, 'conditional abbreviation'

  For example, in the message "**132^2300_CLM_5 at 55 [1]: Value can only be 6:7:8 because 2300_REF_5__Ord160'EXIST'**", **2300_REF_5__** indicates the fifth REF segment in loop 2300, and **'EXIST'** means the segment exists. This message states that the data 2300_CLM5, 1, is invalid because the fifth REF segment exists, which means the value for CLM5 can only be 6, 7, or 8.

- Conditional flag

  For example, in the message "**Value cannot be XX because $!MandateProviderId**", **$!MandateProviderId** is a conditional flag that indicates the Mandate Provider ID flag is not set to validate the nationally mandated provider ID.

In the above formats, the error location may be a fully qualified segment path, such as 2000B_HL_1_1, or may be reduced to a single number (the element number in the segment) if the condition and the constraint are in the same segment. If a segment is repeating within a loop, the repetition number is indicated by a double underscore following the repetition number. For example, 2300_DTP_2__Ord### (two underscores) indicates the second repetition of the DTP segment in loop 2300; 2300_DTP_2_Ord### (one underscore) indicates the second element in the DTP segment.

The abbreviations used in the error descriptions are listed and defined in Table 58 below.

Table 58 Error Message Abbreviations

Abbreviation	Definition
EQ	Is equal to
GT	Is greater than
GE	Is greater than or equal to
LT	Is less than
LE	Is less than or equal to
NE	Is not equal to

Table 58 Error Message Abbreviations

Abbreviation	Definition
EXISTS	Exists
NEXIST	Does not exist
$!MandatePlanId	The National Plan ID flag is set to allow alternative identifiers.
$MandatePlanId	The National Plan ID flag is set to validate for the mandated national identifier.
$!MandateProviderId	The National Provider ID flag is set to allow alternative identifiers.
$MandateProviderId	The National Provider ID flag is set to validate for the mandated national identifier.
$!MandateIndividualId	The National Individual ID flag is set to allow alternative identifiers.
$MandateIndividualId	The National Individual ID flag is set to validate for the mandated national identifier.
$!MandateEmployerId	The National Plan ID flag is set to allow alternative identifiers.
$MandateEmployerId	The National Plan ID flag is set to validate for the mandated national identifier.

C.1.3. 997 Functional Acknowledgment Error Codes

Table 59 below provides a description of applicable error messages that appear in the AK304 element of the 997 Functional Acknowledgment generated by e*Xchange, including the error number, description, and code. The error number appears in the fifth element of the e*Xchange validation error message. Note that AK304 error numbers 6, 7, and 8 are not currently used in the e*Xchange validation error message.

Table 59 Description of AK304 Errors used in e*Xchange Error Reporting

Error Number	Error Description	Error Code
1	Unrecognized segment ID	Err_UnreSegmID
2	Unexpected segment	Err_UnexSegm
3	Mandatory segment missing	Err_MandSegmMiss
4	Loop occurs over maximum times	Err_LoopOccuOverMaxiTime
5	Segment exceeds maximum use	Err_SegmExceMaxiUse
6	Segment not defined in transaction set	Err_SegmNotInDefiTranSet
7	Segment not in proper sequence	Err_SegmNotInPropSequ
8	Segment has data errors	Err_SegmHasDataElemErro

Table 60 below provides a description of applicable error messages that appear in the AK403 element of the 997 Functional Acknowledgment generated by e*Xchange, including the error number, description, and code. The error number appears in the ninth element of the e*Xchange validation error message. Note that AK403 error number 9 is not currently used in the e*Xchange message.

Table 60 Description of AK403 Errors used in e*Xchange Error Reporting

Error Number	Error Description	Error Code
1	Mandatory data element missing	Err_MandDataElemMiss
2	Conditional required data element missing	Err_CondRequDataElemMiss
3	Too many data elements	Err_TooManyDataElem
4	Data element too short	Err_DataElemTooShor
5	Data element too long	Err_DataElemTooLong
6	Invalid character in data element	Err_InvaCharInDataElem
7	Invalid code value	Err_InvaCodeValu
8	Invalid date	Err_InvaDate
10	Exclusion condition violated	Err_ExclCondViol

Index

I

J

L

M

N

O

P

R

S

T

U

V

W

X