

SUN B2B SUITE
**ASC X12 PROTOCOL MANAGER
USER'S GUIDE**

Release 5.1.0



Copyright © 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved. Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries. U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties. Sun, Sun Microsystems, the Sun logo, Java, Sun Java Composite Application Platform Suite, SeeBeyond, eGate, eInsight, eVision, eTL, eXchange, eView, eIndex, eBAM, eWay, and JMS are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd. This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

Copyright © 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés. Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays. L'utilisation est soumise aux termes de la Licence. Cette distribution peut comprendre des composants développés par des tierces parties. Sun, Sun Microsystems, le logo Sun, Java, Sun Java Composite Application Platform Suite, Sun, SeeBeyond, eGate, eInsight, eVision, eTL, eXchange, eView, eIndex, eBAM et eWay sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd. Ce produit est couvert à la législation américaine en matière de contrôle des exportations et peut être soumis à la réglementation en vigueur dans d'autres pays dans le domaine des exportations et importations. Les utilisations, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers les pays sous embargo américain, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exhaustive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

Part Number: 819-7947-10

Version 20070213144058

Contents

List of Figures	8
------------------------	----------

List of Tables	9
-----------------------	----------

Chapter 1

Introduction	10
About ASC X12 Protocol Manager	10
About This Document	11
What's in This Document	11
Scope	11
Intended Audience	11
Text Conventions	12
Screenshots	12
Related Documents	12
References	13
Sun Microsystems, Inc. Web Site	13
Documentation Feedback	13

Chapter 2

Overview of ASC X12 PM	14
About the ASC X12 Protocol	14
Introduction to ASC X12	14
ASC X12 Message Structure	15
Java CAPS Object Type Definitions	15
ASC X12 OTD Usage	15
Structure of ASC X12 Envelopes	16
Functional Groups (GS/GE)	17
Interchange Envelopes (ISA/IEA)	18
Transaction Sets (ST/SE)	19
Elements of ASC X12 Envelopes	20
Data	20
Segments	21
Loops	21
Delimiters	21

Control Numbers	22
ISA13 (Interchange)	22
GS06 (Functional Group)	23
ST02 (Transaction Set)	23
Backward Compatibility	23
Example of EDI Usage	23
Overview of EDI Payments Processing	23
Types of Information Exchanged Electronically	24
Types of Electronic Payment	24
Transfer of Funds	25
Payment-related EDI Transactions	26
Acknowledgment Types	26
TA1, Interchange Acknowledgment	26
997, Functional Acknowledgment	26
Application Acknowledgments	27
ASC X12 Protocol Manager Overview	27
Basic Operation	27
ASC X12 PM and eXchange	28
ASC X12 PM and the ASC X12 OTD Library	28
How ASC X12 PM Messaging Works	28
Key Parts of EDI Processing Logic	29
OTD Message Structures	29
Validations, Translations, Enveloping, and Acknowledgments	29
Message Information Levels	30
ASC X12 Version Support	30
Using the SEF Wizard	30
About eXchange Integrator	31
Understanding BPs	31
Trading Partner Overview	31
Process Overview	32
eXchange Partner Manager	32
B2B Suite, eXchange, and the Java CAPS	32

Chapter 3

Installing ASC X12 PM	34
Supported Operating Systems and Prerequisites	34
Supported Operating Systems	34
System Prerequisites	34
Getting Started With Installation	35
Installing ASC X12 PM Product Files	35
Basic Installation Procedures	36
After You Install	38

 Chapter 4

Configuring ASC X12 PM	39
Configuring eXchange Partner Manager: Overview	39
ePM, B2B Hosts, and Trading Partners	39
eXchange ePM	39
Using ePM: Overview	40
Configuring B2B Hosts and TPs in ePM	43
Configuring ASC X12 PM ePM Parameters	44
Interchange Envelope Parameters	45
ISA01 Author Info Qual	45
ISA02 Author Information	45
ISA03 Sec Info Qual	46
ISA04 Security Information	46
ISA05 IC Sender ID Qual	46
ISA06 Interchange Sender ID	47
ISA07 IC Rcvr ID Qual	47
ISA08 Interchange Rcvr ID	47
ISA11 IC Control Standard Identifier	47
ISA12 IC Version Number	48
ISA13 IC Control Number	48
ISA14 Acknowledgment Requested	48
ISA15 Usage Indicator	49
ISA16 Comp Elem Sep	49
Segment Terminator	49
Element Separator	50
Functional Group Parameters	50
GS01 Functional ID Code	50
GS02 Application Sender Code	51
GS03 Application Rcvr Code	51
GS04 Date Format	51
GS05 Time Format	51
GS06 Group Control Num	52
GS07 Resp Agency Code	52
GS08 Vers/Rel/Indust ID Code	52

 Chapter 5

Quick Start for ASC X12 PM	54
Using the Quick Start Procedures	54
Quick Start, Tutorial, or Both?	54
Overview of Basic Setup Steps	55
Atlanta and Berlin: Sample Business Scenario	55
Sample Scenario Business Description	56
Sample Scenario Projects	56
Getting Started	57
Before You Start	57
Exporting Sample Files	58

Exporting Data Files	58
Exporting ePM Files	59
Editing the Sample Data Files	59
Constructing the Environments	59
Creating External Systems	60
Configuring External Systems	60
Using Deployment Profiles	61
Locating the Projects	61
Deploying the Deployment Profiles	61
Special Considerations for Deployment Profiles	61
Importing Files for ePM	62
Running the ePM Interface	62
Importing B2B Hosts	62
Importing Schedules	62
Importing Trading Partners	62
Running the Sample Scenario	62

Chapter 6

ASC X12 PM Sample Scenario Tutorial	64
Using This Tutorial	64
Quick Start and Tutorial Approaches	64
Introduction to the Sample Implementation	65
Operational Diagram	65
Server Configurations	66
Preconfiguration for Atlanta and Berlin Environments	66
Creating and Starting the Domains	66
Adding a New User to ePM and Message Tracking	67
Adding the Application Server Instances	69
Initializing and Running Enterprise Designer	70
Editing the Sample Data .xml Files	71
Constructing the Environments	74
Using Environment Explorer	74
Setting up the Environments	74
Constructing the Projects	81
Constructing the B2B Host Project	81
eXchange Deployment Project	83
Creating the Validation Connectivity Map	83
Building and Deploying the Deployment Profiles	85
Constructing the 855_FromInt_850 Project	85
Updating the bp855 Business Process	86
Building and Deploying the Deployment Profile	87
Constructing the Remaining Projects' Deployment Profiles	87
Summary of Sample Scenario Projects	88
Importing and Configuring Components in ePM	88
Getting Started	89

Contents	Section
Running ePM	89
Importing B2B Hosts	90
Using Schedules	91
Importing TPs	92
Using Action Groups and Transaction Profiles	94
Configuring the Sample Scenario	94
Using Parameters in ePM	94
Interchange Envelope Parameters	96
Functional Group Parameters	97
Configuring Transports	98
Using Message Tracking	99
Before You Begin	99
Accessing Message Tracking	99
Glossary of Acronyms	101
Index	105

List of Figures

Figure 1	ASC X12 Envelope Schematic Diagram	16
Figure 2	ASC X12 997 (Functional Acknowledgment) Segment Table	17
Figure 3	Example of a Functional Group Header (GS)	18
Figure 4	Example of a Functional Group Trailer (GE)	18
Figure 5	Example of an Interchange Header (ISA)	19
Figure 6	Example of an Interchange Trailer (IEA)	19
Figure 7	Example of a Transaction Set Header (ST)	20
Figure 8	Example of a Transaction Set Trailer (SE)	20
Figure 9	Relationship of the B2B Suite, eXchange, and the Java CAPS	33
Figure 10	Java CAPS Installer	37
Figure 11	Sample Scenario Diagram	56
Figure 12	Sample Scenario Operation	65
Figure 13	Integration Server Administration Window	68
Figure 14	Enterprise Manager Window	70
Figure 15	Integration Server Properties Dialog Box: Environment	75
Figure 16	Oracle External System Properties Dialog Box: Environment	76
Figure 17	Sample Scenario's Atlanta Environment	80
Figure 18	Deployment of dpHost_A	82
Figure 19	OTD Validation Connectivity Map Linking	84
Figure 20	Business Process Editor: bp855	86
Figure 21	Initial ePM Window	90
Figure 22	ePM Window With Trading Partner Configuration Tab Selected	93
Figure 23	ASC X12 PM-specific Interchange Envelope Parameters: Example	96
Figure 24	ASC X12 PM-specific Functional Group Parameters: Example	97
Figure 25	Atlanta Host, Example Message Tracking Window	100

List of Tables

Table 1	Text Conventions	12
Table 2	Default Delimiters in the ASC X12 OTD Library	22
Table 3	Key Parts of EDI Processing	29
Table 4	Supported ASC X12 Versions	30
Table 5	ePM Override Inheritance Hierarchy	42
Table 6	Comparing Quick Start and Tutorial	55
Table 7	Sample Scenario Projects Summary	88
Table 8	Overrides for Atlanta B2B Host Configuration > ... > BatchLocalFile	98
Table 9	Overrides for Berlin B2B Host Configuration > ... > BatchLocalFile	98

Introduction

This document explains how to install, configure, deploy, and use the Sun Java™ Composite Application Platform Suite (Java CAPS) ASC X12 Protocol Manager (PM). This product is part of the Sun B2B Suite.

This chapter provides a brief introduction to the purpose, scope, and organization of the document plus additional reference information.

What's in This Chapter

- [About ASC X12 Protocol Manager](#) on page 10
- [About This Document](#) on page 10
- [Related Documents](#) on page 12
- [References](#) on page 13
- [Sun Microsystems, Inc. Web Site](#) on page 13
- [Documentation Feedback](#) on page 13

1.1 About ASC X12 Protocol Manager

This document provides instructions and background information for all users of the Accredited Standards Committee (ASC) X12 protocol and ASC X12 PM. The purpose of this document is to help you with the following information:

- The nature and function of ASC X12 and ASC X12 PM.
- The relationship of ASC X12 PM to other components of the Java CAPS, including eXchange™ Integrator and eGate™ Integrator, as well as the appropriate eWay™ Adapters.
- The ASC X12 PM components and editors and how to use them in your environment.
- How to use the ASC X12 manager features employed by ASC X12 PM.
- Importing and implementing a ASC X12 PM sample Project scenario provided with the product.

1.2 About This Document

This section explains information about this document.

1.2.1 What's in This Document

This document provides information about installing, configuring, and using ASC X12 PM and includes the following chapters:

- **Chapter 1 “Introduction”** provides a brief summary of ASC X12 PM and its operation, as well as an overview of this document.
- **Chapter 2 “Overview of ASC X12 PM”** gives an overview of ASC X12 PM, ASC X12, and eXchange.
- **Chapter 3 “Installing ASC X12 PM”** explains installation procedures, before and after installation, as well as system requirements.
- **Chapter 4 “Configuring ASC X12 PM”** explains the eXchange ePartner Manager (ePM) configuration steps necessary to allow ASC X12 PM to operate in your environment.
- **Chapter 5 “Quick Start for ASC X12 PM”** provides a brief overview of how to set up and run the ASC X12 PM Project sample scenario provided with the product.
- **Chapter 6 “ASC X12 PM Sample Scenario Tutorial”** explains in detail, how to implement and use the ASC X12 PM sample Project scenario.

1.2.2 Scope

This document describes how to install, set up, configure, run, and use ASC X12 PM to function as a component of Java CAPS applications, specifically eXchange and eGate Integrator.

1.2.3 Intended Audience

This document is intended for computer users who have the ability and responsibility of setting up and maintaining a fully functioning Java CAPS system.

These persons must also understand any operating systems on which Java CAPS is installed, for example, and must be thoroughly familiar with Windows-style user interface operations, as well as having a familiarity with the ASC X12 protocol,

1.2.4 Text Conventions

The following conventions are observed throughout this document.

Table 1 Text Conventions

Text Convention	Used For	Examples
Bold	Names of buttons, files, icons, parameters, variables, methods, menus, and objects	<ul style="list-style-type: none"> ▪ Click OK. ▪ On the File menu, click Exit. ▪ Select the eGate.sar file.
Monospaced	Command line arguments, code samples; variables are shown in <i>bold italic</i>	java -jar <i>filename</i> .jar
Blue bold	Hypertext links within document	See Screenshots on page 12
<u>Blue underlined</u>	Hypertext links for Web addresses (URLs) or email addresses	http://www.sun.com

1.2.5 Screenshots

Depending on what products you have installed, and how they are configured, the screenshots in this document may differ from what you see on your system.

1.3 Related Documents

Use the following related Sun SeeBeyond documents as a reference for additional information in using ASC X12 PM, if needed:

- *Java Composite Application Platform Suite Installation Guide*
- *Java Composite Application Platform Suite Deployment Guide*
- *Sun SeeBeyond eGate Integrator User's Guide*
- *Sun SeeBeyond eGate Integrator System Administration Guide*
- *Sun SeeBeyond eGate Integrator JMS Reference Guide*
- *Sun SeeBeyond File eWay Adapter User's Guide*
- *Sun SeeBeyond Batch eWay Adapter User's Guide*
- *Sun SeeBeyond Oracle eWay Adapter User's Guide*
- *Sun SeeBeyond LDAP eWay Adapter User's Guide*
- *Sun SeeBeyond HTTP(S) eWay Adapter User's Guide*
- *Sun SeeBeyond eInsight Business Process Manager User's Guide*
- *Sun SeeBeyond ASC X12 OTD Library User's Guide*

- *Sun B2B Suite eXchange Integrator User's Guide*
- *Sun B2B Suite eXchange Developer's Guide*
- B2B Suite Readme file for ASC X12 PM information

1.4 References

For more information on the ASC X12 protocol, see the following Web site:

- <http://www.x12.org/>

1.5 Sun Microsystems, Inc. Web Site

The Sun Microsystems web site is your best source for up-to-the-minute product news and technical support information. The site's URL is:

<http://www.sun.com>

1.6 Documentation Feedback

We appreciate your feedback. Please send any comments or suggestions regarding this document to:

CAPS_docsfeedback@sun.com

Please send any comments or suggestions regarding this product to:

b2b-feedback@sun.com

Overview of ASC X12 PM

This chapter provides a general overview of ASC X12 data structuring, as well as ASC X12 PM and its place in the Java CAPS, including system descriptions, general operations, and basic features.

What's in This Chapter

- [About the ASC X12 Protocol](#) on page 14
- [ASC X12 Protocol Manager Overview](#) on page 27
- [About eXchange Integrator](#) on page 31
- [B2B Suite, eXchange, and the Java CAPS](#) on page 32

2.1 About the ASC X12 Protocol

This section provides the following information:

- An overview of the ASC X12 protocol, including the structure of an ASC X12 envelope, data elements, and syntax.
- An explanation of how to use the generic message structures provided as an add-on to eGate and eXchange, to help you quickly create the structures you need for ASC X12 message transactions.
- An example of how ASC X12 is used in payment processing.

2.1.1 Introduction to ASC X12

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

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

An organization called 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.

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 messaging interchanges relating to business transactions, such as order placement and processing, shipping and receiving information, invoicing, and payment information.

ASC 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 instead of being held for batching).

2.1.2 ASC X12 Message Structure

ASC X12 messages have a message structure that indicates how data elements are organized and related to each other for a particular electronic EDI transaction.

Java CAPS Object Type Definitions

In the Java CAPS, message structures are defined as Object Type Definitions (OTDs). Each OTD consists of the following elements:

- **Physical Hierarchy**

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

- **Delimiters**

The specific predefined characters 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 part, for example, whether it is required, optional, or repeating.

Note: Included with the ASC X12 PM product is the ASC X12 OTD Library. See the *ASC X12 OTD Library User's Guide* for details.

ASC X12 OTD Usage

The Transaction Set structure of an invoice sent from one trading partner (TP) to another defines the header, trailer, segments, and data elements required by invoice transactions. The ASC X12 OTD for a specific version includes Transaction Set structures for each of the transactions available in that version. You can use these structures as provided, or customize them to suit your business needs.

eXchange uses OTDs based on ASC 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 ASC X12 transaction. The list of transactions provided is different for each version of ASC X12.

2.1.3 Structure of ASC X12 Envelopes

The rules for ASC X12 envelope structure ensure the integrity of the data and the efficiency of the information exchange. The actual ASC X12 message structure has primary levels that are hierarchical. From highest to the lowest, they are:

- Interchange Envelope
- Functional Group
- Transaction Set

A schematic structure of ASC X12 envelopes is shown in Figure 1. Each of these levels is explained in more detail in the remainder of this section.

Figure 1 ASC X12 Envelope Schematic Diagram

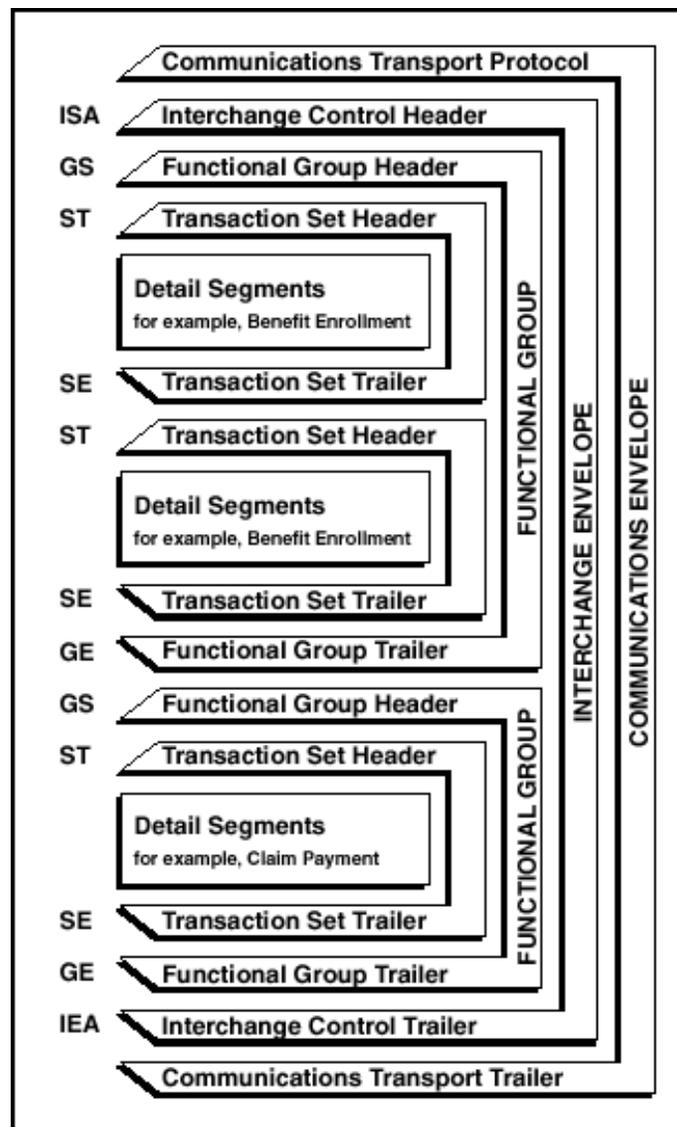


Figure 2 shows the standard segment table for an ASC X12 997 (Functional Acknowledgment) as it appears in the ASC X12 standard and in most industry-specific implementation guides.

Figure 2 ASC X12 997 (Functional Acknowledgment) 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	

Functional Groups (GS/GE)

Functional Groups, often referred to as the “inner envelope,” are made up of one or more Transaction Sets, all of the same type, which can be batched together into one transmission. The Functional Group is defined by the header and trailer segments.

The Functional Group Header (designated GS) segment appears at the beginning, and the Functional Group Trailer (designated GE) segment 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 constitute a specific Functional Group are identified by this functional ID code.

The GS segment contains:

- 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 GE segment contains:

- Number of Transaction Sets included
- Group control number (originated and maintained by the sender)

See Figure 3 and Figure 4.

Figure 3 Example of a Functional Group Header (GS)

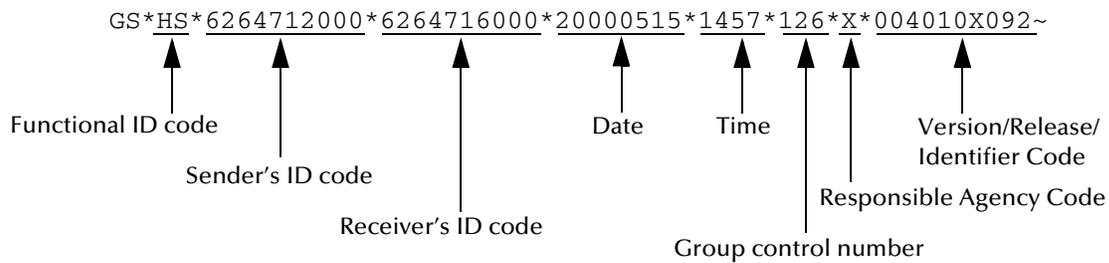
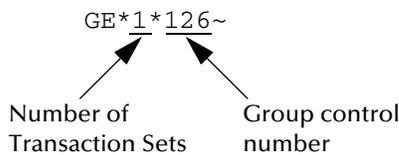


Figure 4 Example of a Functional Group Trailer (GE)



Interchange Envelopes (ISA/IEA)

The Interchange Envelope, often referred to as the “outer envelope,” is the wrapper for all the data to be sent in one transmission. It can contain multiple Functional Groups. This characteristic 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 (designated ISA) appears at the beginning, and the Interchange Control Trailer (designated IEA) appears at the end.

As well as enveloping one or more Functional Groups, the ISA and IEA segments include:

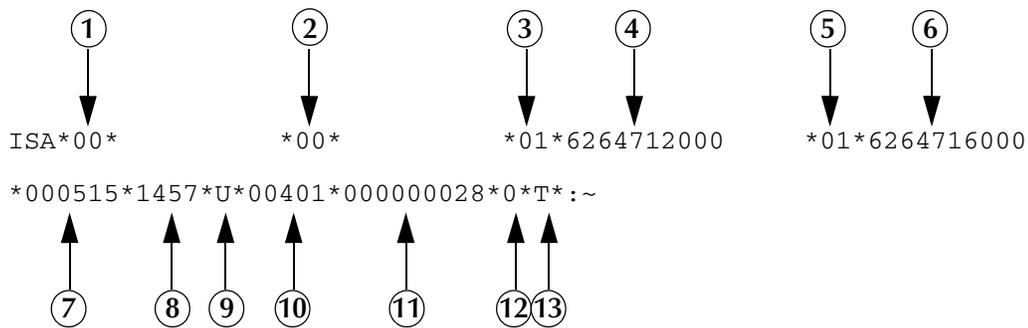
- Data element separators and data segment terminator
- Identification of sender and receiver
- Control information (used to verify message was correctly received)
- Authorization and security information, if applicable

The sequence of information transmitted is:

- ISA
- Optional interchange-related control segments
- Actual message information, grouped by transaction type into Functional Groups
- IEA

See Figure 5 and Figure 6.

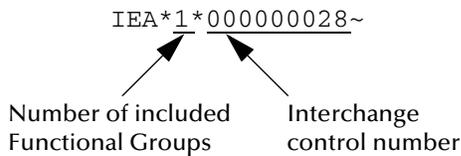
Figure 5 Example of an Interchange Header (ISA)



The following list describes the ISA segments shown in Figure 5:

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

Figure 6 Example of an Interchange Trailer (IEA)



Transaction Sets (ST/SE)

Each Transaction Set (also called a transaction) contains:

- Transaction Set header (designated ST)
- Transaction Set trailer (designated SE)
- Single message, enveloped within the header and footer

A Transaction Set has a three-digit code, a text title, and a two-letter code, for example, **997, Functional Acknowledgment (FA)**.

The Transaction Set is composed 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, postal code, and other geographical information. A Transaction Set may contain multiple segments. For example, the address segment might be used repeatedly to convey multiple sets of address information.

The ASC 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 can be compared to the relationship between records and fields in a database environment. See Figure 7 and Figure 8.

Figure 7 Example of a Transaction Set Header (ST)

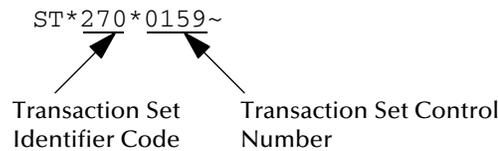
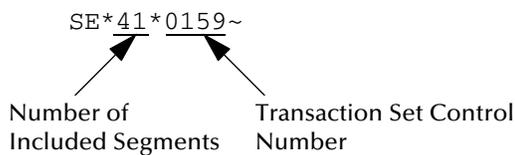


Figure 8 Example of a Transaction Set Trailer (SE)



2.1.4 Elements of ASC X12 Envelopes

The ASC X12 messages are all in ASCII text, with the single exception that the BIN segment is binary. Each ASC X12 message is made up of a combination of the following elements:

- Data
- Segments
- Loops

Elements are separated by delimiters. The remainder of this section explains these elements.

Data

The data element is the smallest named unit of information in the ASC X12 standard. Data elements can be broken down into types. The distinction between the types is strictly a matter of how they are used. The 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 ordinal member of a composite data structure, it is called a *composite* data element. A telephone number is a simple example of a composite. It has a three-digit area code, which must precede the three-digit central office code, which must precede the final four digits.

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

Segments

A segment is a logical grouping of data elements. In ASC X12, the same segment can be used for different purposes. This means that a field'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 ASC X12 enveloping segments, refer to [Structure of ASC X12 Envelopes](#) on page 16.

Loops

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

- Transaction Set (for example, 270 or 271)
- Loop (for example, "info. receiver loop")
- Occurrence of the loop
- Segment (for example, BGN)
- Field number (for example, 01)
- Occurrence of the segment (if it is a repeating segment)

Delimiters

In an ASC X12 message, the various delimiters act as syntax, dividing up the different elements of a message. The delimiters used in a 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 ASC X12 standards, but the industry-specific implementation guides do have recommended delimiters.

The default delimiters used by the ASC X12 OTD Library are the same as those recommended by the industry-specific implementation guides. These delimiters are shown in Table 2.

Table 2 Default Delimiters in the ASC X12 OTD Library

Type of Delimiter	Default Value
Segment terminator	~ (tilde)
Data element separator	* (asterisk)
Subelement (component) separator	: (colon)
Repetition separator (version 4020 and later)	+ (plus sign)

Within eXchange, delimiters are specified at the enveloping level. The delimiters defined for an envelope apply to all transactions in the same business service (a predefined dialog between the two parties).

If you do not specify delimiters and do not override them in the payload transactions fed into FROMINTERNAL, eXchange expects the default delimiters shown in Table 2.

Note: *It is important to note that errors could result if the transmitted data 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 ASC X12 and can cause problems with translation.*

See [Element Separator](#) on page 50 for information on the ePM delimiter parameter.

2.1.5 Control Numbers

The ASC X12 standard includes a control number for each enveloping layer, as follows:

- **ISA13:** Interchange
- **GS06:** Functional Group
- **ST02:** Transaction Set

The control numbers act as identifiers. These numbers are used for message identification and tracking. eXchange includes a flag for each control number. This feature allows you to choose not to assign control numbers to outgoing messages and not to store control numbers on incoming messages.

The remainder of this section provides a brief explanation of each of these control numbers' usage.

ISA13 (Interchange)

The ISA13 control number is assigned by the message sender. It must be unique for each interchange. This is the primary means used by eXchange Integrator to identify an individual interchange.

GS06 (Functional Group)

The GS06 control number is assigned by the sender. It must be unique within the Functional Group assigned by the originator for a Transaction Set.

eXchange ensures that 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)

The ST02 control number is assigned by the sender and is stored in the Transaction Set header. It must be unique within the Functional Group.

eXchange ensures that the control number in ST02 is identical with the SE02 element in the Transaction Set trailer and is unique within a Functional Group (GS-GE). Once you have defined a value for SE02, eXchange always uses that same value for SE02.

2.1.6 Backward Compatibility

Each version of ASC X12 is slightly different. Each new version has some new transactions. In addition, existing transactions can change from version to version.

New versions of ASC X12 are usually backward-compatible. However, this compatibility is not a requirement of the ASC X12 rules. You cannot assume that different versions of ASC X12 are always completely backward-compatible.

You can expect that when you analyze the differences, only a few minor changes are required in the message structures. Therefore, it is recommended that, when you are using a new version or messaging between different versions, you make sure to be thoroughly familiar with any possible compatibility issues.

Note: In this context, "backward compatibility" means that software that parses one version can, in some circumstances, be unable to parse the next version, even if the software ignores any unexpected new segments, data elements at the end of segments, and sub-elements at the end of composite data elements. No backward compatibility means that required segments can disappear entirely, data elements can change format and usage, and required data elements can become optional.

2.1.7 Example of EDI Usage

This section provides, as an example, an overview of the normal processes involved in EDI payment processing, that is, how electronic payments processing is used. Not everything in this example applies to the use of ASC X12 in all possible payments processing scenarios.

Overview of EDI Payments Processing

EDI payments processing encompasses both collection and disbursement transactions. The exchange of funds is accomplished by means of credit and debit transfers. Transfers can also include a related bank balances, as well as transactions and account-analysis reporting mechanisms.

Most nonmonetary EDI TP communications are handled either directly between the parties or indirectly through their respective value-added networks (VANs). However, the exchange of funds requires a financial intermediary. This “third party” is normally the bank or banks that hold deposit accounts of the two parties.

The EDI operations involve the exchange of remittance information along with the order to pay. The remittance information, which acts as an electronic check stub, can be sent in any of the following ways:

- Directly between TPs or through their respective EDI VAN mailboxes
- Through the banking system, with the beneficiary’s bank sending notice of payment to the beneficiary
- By the originator to the originator’s bank as an order to pay, with the originator’s bank notifying the beneficiary

The TPs and the capabilities of their respective banks determine:

- Routing of the electronic check stub
- Whether payment is of the type:
 - ♦ Debit authorized by the payor and originated by the beneficiary
 - ♦ Credit transfer originated by the payor

Types of Information Exchanged Electronically

The following types of information can be exchanged electronically between bank and customer:

- Daily reports of balances and transactions
- Reports of lock-box and electronic funds transfer (EFT) remittances received by the bank
- Authorizations issued to the bank to honor debit transfers
- Monthly customer account analysis statements
- Account reconciliation statements
- Statements of the demand-deposit account

The electronic payment mechanism, which is a subset of EDI, involves the following separate activities:

- Exchange of payment orders, causing value to transfer from one account to another
- Exchange of related remittance information in standardized machine-processable formats

Types of Electronic Payment

An electronic payment can be either:

- Credit transfer, initiated by the payor
- Debit transfer, initiated by the payee as authorized by the payor

Regardless of how a credit transfer was initiated, the payor sends a payment order to its bank in the form of an ASC X12 Payment Order/Remittance Advice (Transaction Set 820).

The bank then adds data in a format required by the United States by the National Automated Clearing House Association (NACHA) and originates the payment through the Automated Clearing House (ACH) system. A corporate-to-corporate payment then performs the following functions:

- Transfers actual monetary value
- Transfers notification of payment from payor to payee

When a credit transfer occurs, these two functions are sometimes treated as one, and sometimes treated separately. These functions can travel in either of the following ways:

- Together through the banking system
- Separately and by different routes

ASC X12 Transaction Set 820 is a data format for transporting a payment order from the originator to its bank. This payment order can be:

- Instructions to the originator's bank to originate a credit transfer
- Instructions to the TP to originate a debit transfer against the payor's bank account

Once this decision has been made, Transaction Set 820 transports the remittance information to the beneficiary. The transfer can either be through the banking system or by a designated route separate from the transport of funds.

Note: *Whenever the Transaction Set 820 remittance information is not transferred with the funds, it can be transmitted directly from the originator to the beneficiary. It can also be transmitted through an intermediary, such as a VAN.*

Transfer of Funds

Before funds can be applied against an open accounts-receivable account, the beneficiary must reconcile the following data streams:

- Payment advice from the receiving bank
- Remittance information received through a separate channel

These data streams were separated during the transfer operation.

If this reconciliation does not take place, and if the amount of funds received differs from the amount indicated in the remittance advice, the beneficiary might have problems balancing the accounts-receivable ledger.

The value transfer begins when the originator issues a payment order to the originator's bank. If a credit transfer is specified, the originator's bank charges the originator's bank account and pays the amount to the beneficiary's bank for credit to the beneficiary's account.

If the payment order specifies a debit transfer, the originator is the beneficiary. In this case, the beneficiary's bank originates the value transfer, and the payor's account is debited (charged) for a set amount, which is credited to the originator's (beneficiary's) bank account.

The payor must issue approval to its bank to honor the debit transfer, either before the beneficiary presents the debit transfer or at the same time. This debit authorization or approval can take one or more of the following forms:

- Individual item approval
- Blanket approval of all incoming debits with an upper dollar limit
- Blanket approval for a particular TP to originate any debit

Payment-related EDI Transactions

ASC X12 uses an end-to-end method to route the 820 Payment Order/Remittance Advice from the originator company through the banks to the beneficiary. Use of this method means there can be several relay points between the sender and the receiver.

The Transaction Set 820 is wrapped in an ACH banking transaction for the actual funds transfer between the banks.

2.1.8 Acknowledgment Types

ASC X12 includes the following types of acknowledgments:

- TA1 Interchange Acknowledgment
- 997 Functional Acknowledgment

Note: See [Sample Scenario Business Description](#) on page 56.

TA1, Interchange Acknowledgment

The TA1 Interchange 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 Gs and transactions.

997, Functional Acknowledgment

The Transaction Set 997 Functional Acknowledgment includes much more information than the TA1 (see [Figure 2 on page 17](#)). This Transaction Set was designed to allow TPs to establish a comprehensive control function as part of the business exchange process.

There is a one-to-one correspondence between Transaction Set 997 and a Functional Group. Segments within this Transaction Set 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, Transaction Set 997 is used as a functional acknowledgment to a Functional Group that was transmitted previously.

Important: *Transaction Set 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 (Transaction Set 997) from the beneficiary of the payment. Transaction Set 997 is created using the data about the identity and address of the originator found in the ISA and/or GS segments.

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 (Transaction Set 824) is a generic application acknowledgment that can be used in response to any ASC X12 transaction. However, it has to be set up as a response transaction; only TA1 and Transaction Set 997 transactions are sent out automatically.

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

2.2 ASC X12 Protocol Manager Overview

Because ASC X12 PM integrates with eGate, eInsight, eXchange, and the ASC X12 OTD Library, the product enables you to design Java CAPS Projects that process and validate ASC X12 messages.

Note: *As you use this document, refer to the **eXchange Integrator User's Guide** for information relating directly to eXchange operation.*

2.2.1 Basic Operation

ASC X12 PM handles the basic operations necessary for ASC X12 messaging, such as:

- Interchange and acknowledgment processing
- Business message correlation
- Enveloping and de-enveloping
- Document batching and splitting
- Event archiving

ASC X12 PM and eXchange

eGate and eXchange enable you to build Java CAPS Projects that process standard B2B business communication and enveloping protocols, such as ASC X12. ASC X12 PM works during message processing with eXchange to provide the following features:

- Message transport
- Message tracking
- Error handling
- Structural message validation

ASC X12 PM and the ASC X12 OTD Library

ASC X12 PM provides packaged Business Protocols with rules that process and validate ASC X12 messages, which are called OTDs in the Java CAPS. The Java CAPS provides packaged ASC X12 OTDs as part of the ASC X12 OTD Library. You can also build your own OTDs using the SEF OTD wizard, which is supplied with eGate.

2.2.2 How ASC X12 PM Messaging Works

Total operation of ASC X12 PM happens in the following functional layers:

- **View:** The eXchange Protocol Designer provides the ability to create ASC X12 Projects, and the eXchange Message Tracking feature allows the searching and viewing of ASC X12 messages.
- **Services Orchestration:** You use the eXchange Service Designer to design a Transaction Set. Then, the ASC X12 Project prepares and returns the interchange and functional acknowledgments (TA1 and 997) to the TP, as well as performing message correlation to associate the business response with the appropriate request.
- **Integration Services:** ASC X12 Projects validate ISA and GS envelopes from incoming messages, prepare ISA and GS envelopes for outgoing messages, batch together documents to be delivered as a single transaction (ISA), and record the activities in Message Tracking.

2.2.3 Key Parts of EDI Processing Logic

The key parts of EDI processing logic are listed in Table 3.

Table 3 Key Parts of EDI Processing

Term	Description	Language Analogy	eGate Component
Structure	Format, segments, loops	Syntax rules	OTD elements and fields
Validation	Data contents “edit” rules	Semantic rules	Validation methods
Translation (also called mapping)	Reformatting or conversion	Translation	Collaborations, Java Collaboration Definitions (JCDs)
Enveloping	Header and trailer segments	Envelope for a written letter	Special “envelope” OTDs: FunctionalGroupEnv and InterchangeEnv
Ack	Acknowledgments	Return receipt	Specific acknowledgment elements in the OTD

eGate uses the structures, validations, translations, enveloping, and acknowledgments, as explained in the remainder of this section, to support the ASC X12 standard.

OTD Message Structures

The ASC X12 OTD Library includes pre-built OTDs for all supported ASC X12 versions. These OTDs can be viewed in the OTD Editor, but cannot be modified.

To customize the OTD structure, for example, to add a segment or loop, you must first generate a .sef file (typically using a third-party application). You then use the SEF OTD Wizard to generate the OTD.

Validations, Translations, Enveloping, and Acknowledgments

Within each ASC X12 OTD are Java methods and Java bean nodes for handling validation. The marshal and unmarshal methods of the two *envelope* OTDs handle enveloping and de-enveloping.

Note: For more information on these OTDs, see the *ASC X12 OTD Library User’s Guide*.

No prebuilt translations are supplied with the ASC X12 OTD Library. You build data translations with an eGate Enterprise Designer user interface called the Java Collaboration Editor (JCE).

Note: In eGate, ASC X12 translations are called *Java Collaboration Definitions (JCDs)*.

Message Information Levels

The following levels of information guide the final format of a specific ASC X12 transaction:

- **ASC X12 Standard:** The Accredited Standards Committee publishes a standard structure for each ASC X12 transaction.
- **Industry-specific Implementation Guides:** Specific industries publish Implementation Guides customized for that industry. Normally, these guidelines are provided as recommendations only. However, for the sake of uniformity and compatibility, it is extremely important to follow these guidelines.
- **TP Agreements:** It is normal for TPs to have individual agreements that supplement the standard guides. The specific processing of the transactions in each TP's individual system can vary from one site to another. Because of this, additional documentation providing information about the differences is helpful to the site's TPs and simplifies implementation. For example, although a certain code might be valid in an implementation guide, a specific TP might not use that code in transactions. In such a case, it could be important to include that information in a TP agreement.

2.2.4 ASC X12 Version Support

ASC X12 PM provides support for the ASC X12 versions shown in Table 4.

Table 4 Supported ASC X12 Versions

▪ 4010	▪ 4020	▪ 4030	▪ 4040	▪ 4050	▪ 4060
▪ 4011	▪ 4021	▪ 4031	▪ 4041	▪ 4051	▪ 4061
▪ 4012	▪ 4022	▪ 4032	▪ 4042	▪ 4052	

2.2.5 Using the SEF Wizard

You can use this product with custom SEF OTDs built with the SEF OTD wizard. The wizard supports the most current SEF versions.

The SEF OTD wizard does not handle the following information and sections:

- In the .SEMREFS section, semantic rules with its type of the "exit routine" are ignored as per SEF specification. An exit routine specifies an external routine (such as a COM-enabled server program supporting OLE automation) to run for translators or EDI data analyzers.
- The .TEXT sections, for example, .TEXT,SETS, .TEXT,SEGS, .TEXT,COMS, and .TEXT,ELMS, are ignored, because these sections store information about changes in a standard's text, such as notes, comments, names, purposes, descriptions, titles, semantic notes, explanations, and definitions.

2.3 About eXchange Integrator

eXchange Integrator provides an open Business Protocol framework to support standard EDI and B2B protocols, as well as packaging protocols. The eXchange product supports existing standard protocols, using an extensive set of prebuilt eInsight Business Processes (BPs). It also provides the tools and framework to create and adopt new protocols and to build custom BPs.

B2B modeling semantics are exposed so that eInsight Business Rules can be added and tailored to address the particular needs of providing eBusiness solutions. The tight integration with the rest of Java CAPS provides validation, logging, and reporting capabilities. Because each logical step within any Business Rule is accessible anywhere along the entire eInsight BP, the design tools provide complete end-to-end visibility.

Note: For a complete explanation of eXchange and eInsight, as well as their operation, see the **eXchange Integrator User's Guide** and **eInsight Business Process Manager User's Guide**.

2.3.1 Understanding BPs

An eInsight BP is a collection of actions or operations that take place in your company, revolving around a specific business practice. These processes can involve a variety of participants and may include internal and external computer systems or employees.

In eXchange, you create a graphical representation of a business process called a BP model. When you are using the sample for a PM's implementation scenario, the system uses the BPs necessary for scenario's operation. The BPs specific to the sample scenario provided with the product have already been created for this scenario.

2.3.2 Trading Partner Overview

The architecture of eXchange centers around the concept of sending and receiving messages relative to one or more TPs. Each TP that you import or create and then configure corresponds to one of your business trading partners.

These TPs contain configurable transaction profiles for each individual TP relationship. You can configure TPs with Transaction Profiles and Schedules, within ePM, for use by run-time components.

Each Transaction Profile specifies which one or more BPs to use for the current transaction, where and how to receive inbound messages, how to configure and secure messages in their channels, and how and where to deliver outbound messages.

2.3.3 Process Overview

Using eXchange to create a business solution consists of the following phases:

- Design phase within Enterprise Designer
- Configuration/design phase within ePM
- Run-time phase

2.3.4 eXchange Partner Manager

ePM is a feature of eXchange you can use as a tool that allows you to configure eXchange for use with any of your PMs. You must set specific parameter values within ePM to ensure the correct operation of your Projects, for each protocol. This guide explains how to use and configure ePM, to set parameters relevant to this guide's PM.

For more information on how to use ePM, see the *eXchange Integrator User's Guide*.

2.4 B2B Suite, eXchange, and the Java CAPS

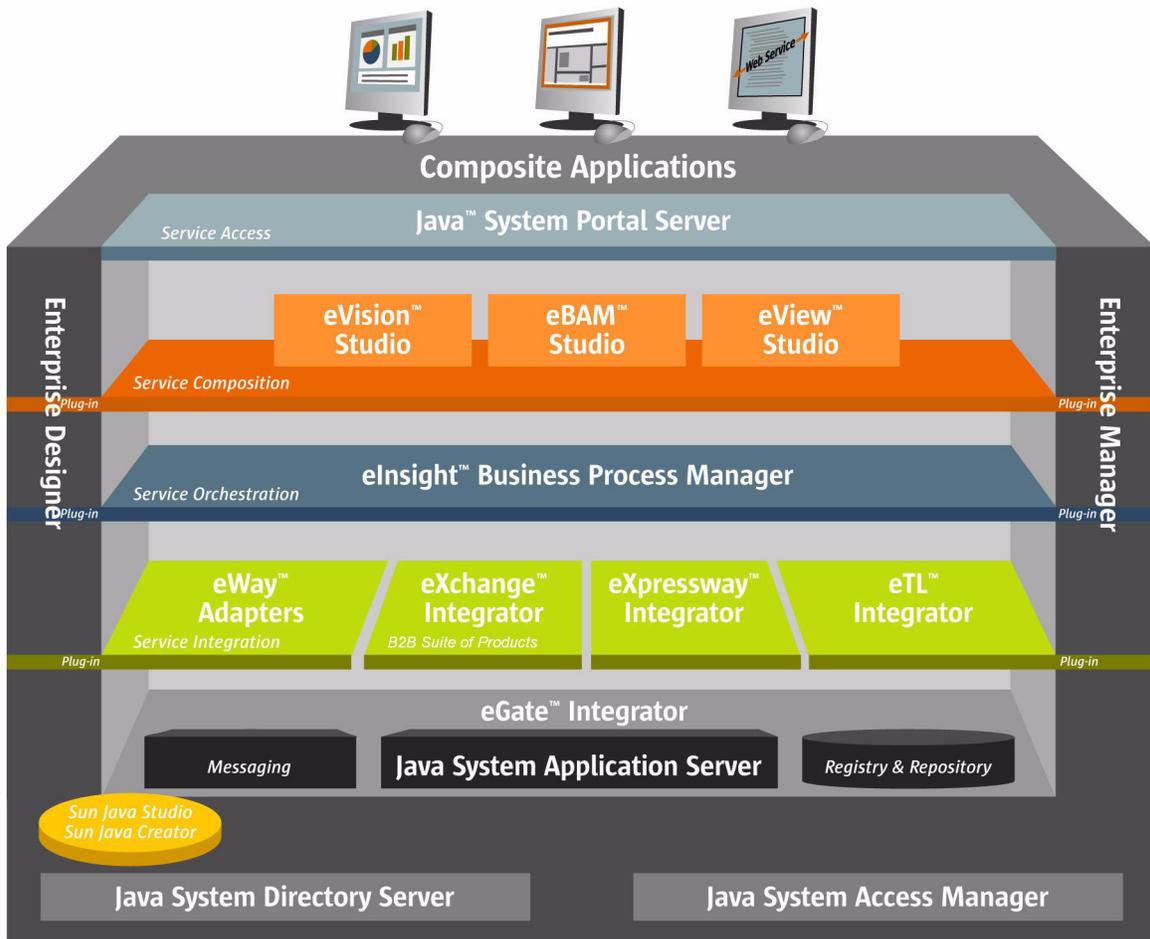
eXchange is one of the products that make up the the B2B Suite within the Java CAPS. The B2B Suite products, including eXchange, provide a Web-based TP configuration and management solution for automating and securely managing business partner relationships. The products also facilitate real-time interaction between the enterprise and its TPs, suppliers, and customers.

As a part of the Java CAPS, the B2B Suite provides the following benefits and features:

- B2B services via eXchange
- Protocol managing, specifically by the PMs
- Protocol formats contained in the OTD Libraries
- Trading partner management facility, that is, the ePM interface
- Archiving tool, the Message Tracking feature

The B2B Suite is tightly integrated with the Java CAPS and runs as a group of components within the Java CAPS environment. Figure 9 illustrates how the B2B Suite, eXchange, and other Java CAPS components work together, including eInsight.

Figure 9 Relationship of the B2B Suite, eXchange, and the Java CAPS



Installing ASC X12 PM

This chapter explains how to install ASC X12 PM, as well as pre- and post-installation procedures.

What's in This Chapter

- [Getting Started With Installation](#) on page 35
- [Installing ASC X12 PM Product Files](#) on page 35

3.1 Supported Operating Systems and Prerequisites

This section describes the ASC X12 PM supported operating systems and the prerequisites you need for its use.

3.1.1 Supported Operating Systems

ASC X12 PM supports all operating systems supported by eGate and eXchange versions compliant with this release. See the *eXchange Integrator User's Guide* for a complete list.

3.1.2 System Prerequisites

ASC X12 PM requires the following Sun SeeBeyond products (compliant with this release) for its correct operation:

- Core Products
 - ♦ eGate
 - ♦ eXchange
 - ♦ eInsight
 - ♦ SEF OTD Wizard

- eWays
 - ♦ Batch eWay
 - ♦ File eWay
 - ♦ HTTP eWay
 - ♦ LDAP eWay
 - ♦ Oracle eWay
- OTD Library
 - ♦ ASC X12 OTD Library

See the eXchange Readme file for a list of the Sun SeeBeyond products compliant with this release.

Third-party Application Requirements

There are no third-party application requirements.

3.2 Getting Started With Installation

Open and review the Readme file for the Java CAPS to gain current information you may need, for example for eGate or eInsight, before installing ASC X12 PM. You can find this file in the root directory of the Java CAPS installation's Repository CD-ROM.

Also, ASC X12 PM has its own **ASC_X12_Manager_Readme.txt** file that contains additional information specific to this application, including required ESRs.

Note: See the *SeeBeyond Java CAPS Installation Guide* for details on how to obtain the Readme and documentation files.

This version of ASC X12 PM is compatible with Java CAPS version 5.1.2 (eGate and eInsight).

Configuring eGate Projects for Large Messages

If an eGate Project uses Sun SeeBeyond JMS (Java Messaging Service) IQ Manager and is estimated to process messages or transactions over 8 MB for Windows, or 16 MB for UNIX, you must increase the **Segment Size** property of JMS IQ Manager as explained in the *eGate Integrator JMS Reference Guide*.

3.3 Installing ASC X12 PM Product Files

During the Java CAPS installation operation, use Enterprise Manager, a Web-based application, to select and upload ASC X12 PM and add-on application .sar files from the Java CAPS installation CD-ROM to the Repository.

When the Repository is running on a UNIX operating system, you must still install ASC X12 PM using Enterprise Manager and Microsoft Internet Explorer on a Windows computer connected to the Repository server.

3.3.1 Basic Installation Procedures

Follow the general instructions for installing the Java CAPS, which you can find in the *Java CAPS Installation Guide*. You must begin by installing eGate. For more information, see the *eGate Integrator System Administration Guide*.

Note: Create and configure the eXchange database using the **Oracle510.zip file**. For complete information on how to perform this and other eXchange setup operations, see the *eXchange Integrator User's Guide*.

In this document, the examples and illustrations show a Repository named **repB2B**, installed under the base installation directory and using the default ports (12000-12009).

To install the Repository and product .sar files

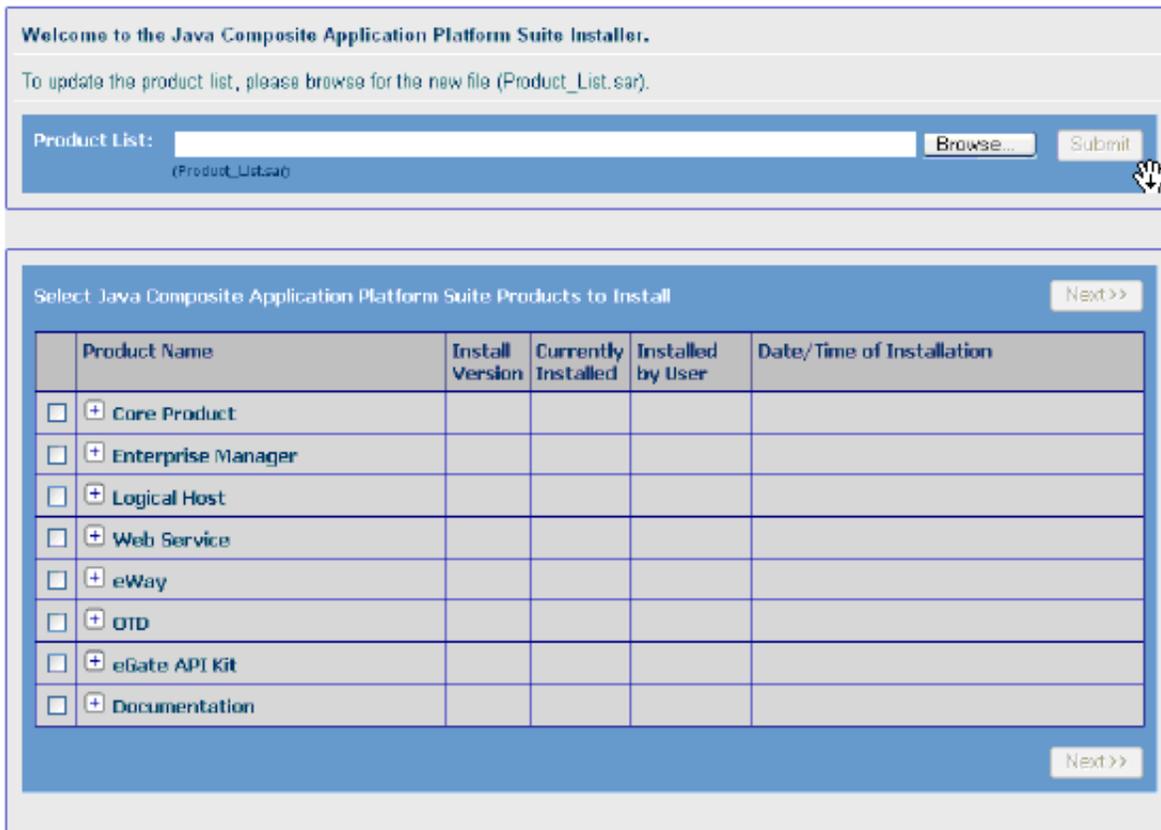
- 1 Install the Repository by running either of the following scripts:
 - ♦ (for Windows) ... \Repository\install.bat
 - ♦ (for UNIX) sh ... /Repository/install.sh
- 2 Upon completion of the installation, start the Repository by running the following script:

```
C:\ ... repository\startserver.bat
```
- 3 Start a new browser session and point your browser at this URL:

```
http://localhost:12000
```
- 4 In the **Java CAPS Login** window of the Java CAPS Installer, supply the following values:
 - ♦ **username:** Administrator
 - ♦ **password:** STC

See Figure 10.

Figure 10 Java CAPS Installer



- 5 Use the Java CAPS Installer's **Administration** tab to install the following file:

- ♦ **eGate.sar**

This installation may require approximately 15 minutes.

Note: Ensure that, to duplicate the sample shown in this document, you have named your Repository *repB2B*.

- 6 Click the link to install additional products, and then browse to the current product list and submit it:

- ♦ **Product_List.sar**

- 7 Stage and install the files listed under this step (installing both together may require approximately 15 minutes).

From the Core Product category:

- ♦ **eInsight.sar**

From the Enterprise Manager category:

... \<yourOS-type>\Enterprise_Manager-<yourOS-Platform>.sar

- 8 Stage and install the following product file, from the Logical Host category:

... \<yourOS-type>\logicalhost-<yourOS-platform>.sar

- 9 Stage and install the following product files, from the eWay category:

- ♦ BatcheWay.sar
- ♦ FileeWay.sar
- ♦ HTTPeWay.sar
- ♦ LDAPeWay.sar
- ♦ OracleeWay.sar

- 10 Stage and install the following product files from the Core Product category:

- ♦ eXchange.sar
- ♦ SEF_OTD_Wizard.sar

Note: You must finish installing *eXchange.sar* before you start to install *X12_Manager.sar*.

- 11 Stage and install the following product file, from the OTD category:

- ♦ X12_v4010_OTD.sar

Note: You must finish installing this file before you start to install *X12_Manager.sar*.

- 12 Stage and install the following product file, from the Core Product category:

- ♦ X12_Manager.sar

- 13 Finally, stage and install the following product file, from the OTD category:

- ♦ X12_v4010_OTD_Validation_BP.sar

3.3.2 After You Install

Once ASC X12 PM is installed and configured, it must then operate in conjunction with an eGate Project before it can perform its intended functions. You must create these Projects specifically for ASC X12 PM, or you must import one or more ASC X12 PM Projects.

See the *eXchange Integrator User's Guide* and *eGate Integrator User's Guide* for detailed information on incorporating these types of Projects into eGate. See [Chapter 5](#) (overview and quick start) and [Chapter 6](#) (detailed tutorial) for information on a sample business scenario with Projects already created, using eXchange and ASC X12 PM.

Configuring ASC X12 PM

This chapter explains how to configure eXchange Partner Manager (ePM) parameters for use with ASC X12 PM and ASC X12.

What's in This Chapter

- [Configuring eXchange Partner Manager: Overview](#) on page 39
- [Configuring ASC X12 PM ePM Parameters](#) on page 44

4.1 Configuring eXchange Partner Manager: Overview

This chapter explains the configuration parameters required for ASC X12 PM Projects and their operation with ASC X12 and other Java CAPS applications. You can configure these parameter values for ASC X12 PM using the eXchange ePM user interface.

Note: *For more information on how to do these operations in ePM, including default values for general eXchange parameters and how to override them, see the [eXchange Integrator User's Guide](#).*

4.1.1 ePM, B2B Hosts, and Trading Partners

You may use ePM to set up and configure ASC X12 PM parameters at the B2B Host configuration level for the Projects in a business scenario. In ePM, B2B Host components can be created from scratch or imported. These components derive their default parameter properties from the B2B Host you built in Enterprise Designer, which contains ASC X12 PM configuration parameters.

4.1.2 eXchange ePM

The eXchange ePM interface allows you to set essential parameter properties for your ASC X12 PM eXchange Projects. This tool also allows you to configure the specific business and messaging functions you want implemented by your B2B Hosts and Trading Partners (TPs).

Ensuring that you have configured the appropriate values in ePM allows the B2B Hosts and TPs you configure to operate seamlessly with eXchange, ASC X12 PM, and ASC X12 within your B2B scenario. Also, TPs in ePM can be either created from scratch or imported.

For more information on B2B Hosts and TPs, see [Constructing the B2B Host Project](#) on page 76 and [Importing and Configuring Components in ePM](#) on page 84.

Using ePM: Overview

In eXchange, each TP contains information identifying the values for using ASC X12 PM with eXchange, as well as communication with the ASC X12 B2B Host and TP delivery and transport information used for sending and receiving B2B information.

Categories of Configurable Properties

There are three categories of configurable parameter properties in ePM, as follows:

- Business Protocols
- Delivery Protocols
- Transports

In general, you may use parameters under Business Protocols to configure data payload-related operations within your business. Also generally, parameters under Delivery Protocols determine data payload-unrelated messaging operations. Parameters under Transports are directly related to eXchange and remain the same regardless of which PM you are using.

Note: See the *eXchange Integrator User's Guide* for more information.

You may locate the current B2B Host or TP in **ePM Explorer**, by clicking the **B2B Host Configuration** or **Trading Partner Configuration** tab. The B2B Host acts as a top-level “parent” component that supplies all default parameter properties to the components under it, including the TP. These components include Action Groups and Transaction Profiles.

Business Actions

You define Business Actions within the B2B Host, as constructed in Enterprise Designer. Business Actions are already a part of the B2B Host in ePM when you begin to configure in ePM. They are the message type, inbound or outbound, for example, **270 FromPartner**.

Action Groups

Under B2B Hosts, you may create Action Groups. They function as “child” components that inherit parameter properties from their “parent” B2B Host. By associating one or more Action Groups with a TP, you define the TP’s general operation.

In terms of usage, for example, you might want to place all of your Transaction Profiles for purchase order requests and responses in one Action Group and give it a name that represents its function.

Note: You cannot create an Action Group within a TP.

Transaction Profiles

Each Transaction Profile enables a specific messaging function and is associated with a B2B Host. For example, you may create Transaction Profiles at the B2B Host level (in the **B2B Host Configuration** tab in ePM), whose parameter values are inherited at the TP level (in the **Trading Partner Configuration** tab in ePM).

A Transaction Profile consists of a Business Protocol Action Group, a Delivery Protocol Action Group (if necessary), and a Transport. Therefore, a given Transaction Profile inherits parameter values from each of its constituent Action Group and Transport components.

At the TP level, a B2B Host Transaction Profile may be used as a part of the configuration of a TP component, that is, one of the TP's Transaction Profiles. In these cases, the TP is said to "inherit" the configuration values of the B2B Host Transaction Profile it is using.

Defaults and Overrides

ePM allows you to override the default parameter properties at any "parent" or "child" component level. Overrides inherit from "parent" to "child" components. Default overrides cascade from B2B Hosts to TPs. You can also configure specific overrides for individual TPs.

Note: For information on Lookup parameters and how they operate in ePM, including how they are inherited and overridden, see the *eXchange Integrator User's Guide*.

The current TP configuration inherits the current B2B Host configuration. Additionally, ePM allows you to override any inherited parameter values at this level or at any lower level in the TP, if necessary (see [Table 5 on page 42](#)). For example, a TP's Action Group's overrides are inherited from the current B2B Host's Action Groups.

Review the previous example of the B2B Host Transaction Profile used as a part of the TP configuration. Since this Transaction Profile also belongs to the current TP, the inherited parameter values may be overridden at this level, effectively "customizing" the Transaction Profile for the TP.

Inheritance and Override Hierarchy

Table 5 illustrates this ePM hierarchy of default override inheritance. Keep in mind that parameter categories only inherit from the same categories, for example, B2B Host Action Group Business Protocols from TP Action Group Business Protocols, and so on.

Table 5 ePM Override Inheritance Hierarchy

Major Component Being Configured	Row in This Table	Selected in ePM Explorer	Parameter Categories in ePM Canvas	Parameter Values Inherit Overrides as Follows:
B2B Host component: May have one or more Action Groups, which in turn may have one or more Transaction Profiles.	Row 1	B2B Host under B2B Host Configuration tab	Business Protocols	Top level; no inheritance
			Delivery Protocols	
			Transports	
	Row 2	Action Group; "child" component relation to "parent" B2B Host above	Business Protocols	From B2B Host Configuration tab values set on ePM canvas
			Delivery Protocols	
	Row 3	Transaction Profile; "child" component relation to "parent" Action Group (and "grandparent" B2B Host) above	Business Protocols	From B2B Host > Action Group configuration values set on ePM canvas
Delivery Protocols				
Transports				
Trading Partner (TP) component: May have one or more Action Groups, which in turn may have one or more Transaction Profiles.	Row 4	TP under Trading Partner Configuration tab	Business Protocols	This Row 4 inherits from Row 1 above
			Delivery Protocols	
			Transports	
	Row 5	TP Action Group; same as Row 2 above (cannot be created at this level)	Business Protocols	This Row 5 inherits from Row 2 above
			Delivery Protocols	
	Row 6	Transaction Profile	Business Protocols	This Row 6 inherits from Row 3 above
Delivery Protocols				
Transports				

It is recommended that you set your necessary configurations at the "highest" level possible, according to the hierarchy shown in the previous table. For example at the B2B Host Business Protocol level or at the TP Business Protocol level. See the sample scenario ePM configuration for an example of these recommended configuration practices.

Note: For more information parameter override inheritance in ePM, see the *eXchange Integrator User's Guide*.

Configuring B2B Hosts and TPs in ePM

Before you can use a TP, you must configure its associated parameters specifically for eXchange, ASC X12 PM, and your B2B operation.

Tabs on ePM Canvas

You configure your ASC X12 PM Projects under the following tabs in ePM:

- **B2B Host Configuration**
- **Trading Partner Configuration**

These tabs offer you the following sets of properties in the Host Explorer tree (left column):

- **Business Protocols**
- **Delivery Protocols**
- **Transports**
- **Transaction Profiles**
- **Private Keys**
- **Contacts**
- **Schedules**

Configuration parameters for your ASC X12 PM Projects are located under the **Business Protocols**, **Delivery Protocols**, and **Transports** categories of properties. See [Categories of Configurable Properties](#) on page 40 for a description of these categories.

Parameter Types

Each set of ePM parameters contains the following parameter types:

- General eXchange parameters common to all PMs, for example, Transports
- ASC X12 PM-specific parameters present only for this PM

Additional Information

This document contains a sample implementation scenario with specific ePM settings configured. You may refer to this sample scenario for ePM implementation examples.

The rest of this chapter explains the available parameters for ASC X12 PM configuration and how to set them.

4.2 Configuring ASC X12 PM ePM Parameters

This section explains how to configure ASC X12 PM-specific parameter values in ePM.

Configuring ePM: ToPartner and FromPartner Messaging Model

The following list explains the outbound and inbound messaging model used for Environments and TPs during ePM configuration:

- When you are configuring ePM for a component related to the current TP's Environment, you must take the viewpoint of that Environment.

For example, Company A is the current TP with its own Environment. You are configuring a component related to Company A's Environment **envA**. Therefore, in terms of the companies, **ToPartner** means from Company A (outbound from **envA**) and **FromPartner**, to Company A (inbound to **envA**).

- Following the same model, when you are configuring ePM for a component related to a *different* TP's Environment, you must take the viewpoint of *that* Environment.

For example, Company B is a TP with an Environment outside of Company B. You are configuring a component related to Company B's Environment **envB**. Therefore, in terms of the companies, **ToPartner** means from Company B (outbound from **envB**) and **FromPartner**, to Company B (inbound to **envB**).

- Therefore, if TPs in the previous examples are named as follows:
 - ♦ **tpB** is Company A's TP.
 - ♦ **tpA** is Company B's TP.

The following relationship holds true:

- ♦ **tpA** is the TP for **envB**.
- ♦ **tpB** is the TP for **envA**.

ASC X12 PM-specific Parameter Types

ASC X12 PM-specific parameters are of two basic types as explained under:

- **Interchange Envelope Parameters** on page 45
- **Functional Group Parameters** on page 50

Note: When ePM displays both sets of parameters in the previous list, parameters appear, which are generic to eXchange and are not explained in this chapter. See the *eXchange Integrator User's Guide* for details on these parameters.

For additional information on how to configure ePM parameters, see **Importing and Configuring Components in ePM** on page 88, as well as the *eXchange Integrator User's Guide*.

4.2.1 Interchange Envelope Parameters

This section explains how to configure Interchange Envelope (outer envelope) ePM parameters for ASC X12 PM. [Figure 23 on page 96](#) shows an ePM example with these parameters displayed. This section describes the following parameters:

- [ISA01 Author Info Qual](#) on page 45
- [ISA02 Author Information](#) on page 45
- [ISA03 Sec Info Qual](#) on page 46
- [ISA04 Security Information](#) on page 46
- [ISA05 IC Sender ID Qual](#) on page 46
- [ISA06 Interchange Sender ID](#) on page 47
- [ISA07 IC Rcvr ID Qual](#) on page 47
- [ISA08 Interchange Rcvr ID](#) on page 47
- [ISA11 IC Control Standard Identifier](#) on page 47
- [ISA12 IC Version Number](#) on page 48
- [ISA13 IC Control Number](#) on page 48
- [ISA14 Acknowledgment Requested](#) on page 48
- [ISA15 Usage Indicator](#) on page 49
- [ISA16 Comp Elem Sep](#) on page 49
- [Segment Terminator](#) on page 49
- [Element Separator](#) on page 50

ISA01 Author Info Qual

Description

Allows you to enter a value representing the code used to identify the type of information in the Authorization Information; required.

Required Values

The appropriate valid integer that represents the Authorization Information.

Default

None

ISA02 Author Information

Description

Allows you to enter a value representing the information used for additional identification or authorization of the interchange sender or the data in the interchange. The type of information is set by the Authorization Information Qualifier (ISA01).

Required Values

The appropriate valid integer that represents the Author Information.

Default

None

ISA03 Sec Info Qual

Description

Allows you to enter a value representing the code used to identify the type of information in the Security Information; required.

Required Values

The appropriate valid integer that represents the Security Information type.

Default

None

ISA04 Security Information

Description

Allows you to enter a value representing security information about the interchange sender or the data in the interchange. The type of information is set by the Security Information Qualifier (ISA03) value.

Required Values

The appropriate valid integer that represents the Security Information.

Default

None

ISA05 IC Sender ID Qual

Description

Allows you to enter a value representing the qualifier used to designate the system/method of information code (IC) structure employed to designate the sender ID element being qualified.

Required Values

The appropriate valid integer that represents the IC sender ID qualifier.

Default

None

ISA06 Interchange Sender ID

Description

Allows you to enter a value representing the IC published by the sender for other parties to use as the receiver ID to route data to them. The sender always codes this value for the sender ID element; required.

Required Values

The appropriate valid integer that represents the interchange sender ID.

Default

None

ISA07 IC Rcvr ID Qual

Description

Allows you to enter a value representing the qualifier used to designate the system/method of IC structure used to designate the receiver ID element being qualified; required.

Required Values

The appropriate valid integer that represents the IC receiver ID qualifier.

Default

None

ISA08 Interchange Rcvr ID

Description

Allows you to enter a value representing the IC published by the receiver of the data. When you are sending, this value is used by the sender as their sending ID, thus other parties sending to them use this as a receiving ID to route data to them; required.

Required Values

The appropriate valid integer that represents the interchange receiver ID.

Default

None

ISA11 IC Control Standard Identifier

Description

Allows you to enter a value representing the IC used to identify the agency responsible for the control standard used by the message that is enclosed by the current interchange header and trailer; required.

Required Values

The appropriate valid integer that represents the IC control standard identifier.

Default

U

ISA12 IC Version Number

Description

Allows you to enter a value representing the current IC version number. This number covers all the current interchange control segments; required.

Required Values

The appropriate valid integer that represents the current IC version number.

Default

00401

ISA13 IC Control Number

Description

Allows you to enter a value representing the data interchange IC control number. This value is a setting that starts calculating the unique identifier for outbound messages. The identifier increments by 1 from this number, for each additional message; required.

Required Values

The appropriate valid integer that represents the IC control number for the first message. It is recommended that you use the default, unless your system needs dictate otherwise. This number *cannot* be 0.

Default

1

ISA14 Acknowledgment Requested

Description

Allows you to enter a value representing the code sent by the sender to request an interchange acknowledgment (TA1); required.

Required Values

The appropriate valid integer that represents the acknowledgment code.

Default

1

ISA15 Usage Indicator

Description

Allows you to enter a value representing the code used to indicate whether data enclosed by the current interchange envelope is for testing, production, or information only; required.

Required Values

P, T, or I.

Default

P

ISA16 Comp Elem Sep

Description

Allows you to enter a value that changes the component element default delimiter.

The data element type is not applicable. The component element separator is a delimiter and not a data element. This field provides the delimiter used to separate component data elements within a composite data structure. This value must be different from the data element separator and the segment terminator.

Required Values

Enter the appropriate delimiter override character.

If you use *nondefault* delimiters (for example, if you use “!” for segment terminator in v4060), you must ensure that your business rules manually pass the nondefault delimiters into the **ExStdEvent/PayloadSection/Envelopes/BusinessProtocol/** location, that is, pass the ISA into **.../Batch/Header**, the IEA into **.../Batch/Trailer**, the GS into **.../Group/Header**, and the GE into **.../Group/Trailer**.

To use a control character as a delimiter, pass the escaped Unicode UTF-16 representation of the character (**\uXXXX**). For example, if you wanted to use a carriage return (ASCII **0x0d**) as a delimiter, you would pass the string **\u000d**.

Default

Colon (:) character

Segment Terminator

Description

Allows you to change the segment terminator default delimiter. See [ISA16 Comp Elem Sep](#) on page 49 for more details.

Required Values

Enter the delimiter override character.

Default

Tilde (~) character

Element Separator

Description

Allows you to change the element separator default delimiter. See [ISA16 Comp Elem Sep](#) on page 49 for more details.

Note: XML reserved characters, for example &, <, or >, cannot be used as delimiters. See the appropriate XML information source for a complete list of these characters.

Required Values

Enter the delimiter override character.

Default

Asterisk (*) character

4.2.2 Functional Group Parameters

This section explains how to configure Functional Group (inner envelope) ePM parameters for ASC X12 PM. [Figure 24 on page 97](#) shows an ePM example with these parameters displayed. This section contains the following parameters:

- [GS01 Functional ID Code](#) on page 50
- [GS02 Application Sender Code](#) on page 51
- [GS03 Application Rcvr Code](#) on page 51
- [GS04 Date Format](#) on page 51
- [GS05 Time Format](#) on page 51
- [GS06 Group Control Num](#) on page 52
- [GS07 Resp Agency Code](#) on page 52
- [GS08 Vers/Rel/Indust ID Code](#) on page 52

GS01 Functional ID Code

Description

Allows you to set a value to match the Group Name attribute of this business action, set in the B2B Host's Business Service; required.

Required Values

For example, a Group Name of "HS" is for a 270 action; "HB" for a 271; "PO" for an 850; "PR" for an 855; "FA" for a 997.

Default

None (depends on the current business action)

GS02 Application Sender Code

Description

Allows you to enter a value representing the code identifying the party sending a message transmission. These codes are predefined and agreed upon by the TPs.

Required Values

The appropriate valid integer that represents the application sender code.

Default

None

GS03 Application Rcvr Code

Description

Allows you to enter a value representing the code identifying the party receiving a message transmission. These codes are predefined and agreed upon by the TPs; required.

Required Values

The appropriate valid integer that represents the application sender code.

Default

None

GS04 Date Format

Description

Allows you to set the date structure for this transaction set, in a format specified under "Required Values;" required.

Required Values

CCYYMMDD or **CCYYYYMMDD**; choose from the list to use a four-digit or two-digit year format, for example:

- 20041201 = December 1, 2004
- 050112 = January 12, 2004

Default

CCYYMMDD

GS05 Time Format

Description

Allows you to set the time structure for this transaction set, in a format specified under "Required Values;" required.

Required Values

HHMM, HHMMSS, HHMMSSD, or HHMMSSDD.

Choose from the list to specify seconds and degrees of accuracy, for example:

- 2359 = 11:59PM
- 235959 = 11:59:59PM
- 23595999 = 11:59:59.99PM

The time is expressed in 24-hour (military) clock time as follows: HHMM, or HHMMSS, or HHMMSSD, or HHMMSSDD, where H = hours (00-23), M = minutes (00-59), S = integer seconds (00-59); and DD = decimal seconds; decimal seconds are expressed as follows: D = tenths (0-9) and DD = hundredths (00-99).

Default

HHMM

GS06 Group Control Num

Description

Allows you to enter the group control number assigned and maintained by the sender.

Required Values

The appropriate valid integer that represents the group control number.

Default

0

GS07 Resp Agency Code

Description

Allows you to enter a value representing the code used in conjunction with data element (DE) 480 used to identify the issuer of the current ASC X12 standard; required.

Required Values

The appropriate correct transaction set code (X is recommended, but you may have to use others, for example, T, as required by the current ASC X12 standard being used.

Default

X

GS08 Vers/Rel/Indust ID Code

Description

Allows you to enter a value representing the code indicating the ASC X12 version, release, subrelease, and industry identifier of the EDI standard being used, including the GS and GE segments; required.

Required Values

If the responsible agency code under the GS07 segment is X (recommended), enter values as follows in the appropriate DE 480 positions:

- 1 through 3 = the current version number
- 4 through 6 = the release and subrelease levels of the current version
- 7 through 12 = the current industry or trade association identifiers (optionally assigned by the user)

If code entered for the GS07 segment is different, for example, T, then other formats are allowed, as required by the current ASC X12 standard being used.

Default

None

Quick Start for ASC X12 PM

ASC X12 PM comes with a sample implementation scenario that includes Projects, B2B Hosts, TPs, and data files. This chapter provides basic overview and “quick-start” procedures. Use this chapter as a ASC X12 PM setup overview and/or a quick way to get started, using the sample scenario.

What’s in This Chapter

- [Using the Quick Start Procedures](#) on page 54
- [Atlanta and Berlin: Sample Business Scenario](#) on page 55
- [Constructing the Environments](#) on page 59
- [Using Deployment Profiles](#) on page 61
- [Importing Files for ePM](#) on page 62
- [Running the Sample Scenario](#) on page 62

5.1 Using the Quick Start Procedures

This chapter provides an overview of the ASC X12 PM sample implementation scenario and basic procedures that describe how to import the necessary files, then efficiently set up, run, and monitor the business scenario. The remainder of this section explains the purpose and content of this chapter in greater detail.

5.1.1 Quick Start, Tutorial, or Both?

The ASC X12 PM product includes a complete sample implementation, included in the **ASC_X12_ManagerDocs.sar** file, that allows you to see the end results without having to go through all the design steps.

If you import and set up this sample scenario, as described in this chapter, you can see run-time results quickly without having to read detailed instructions.

The tutorial in [Chapter 6](#), on the other hand, provides a detailed hands-on guide to creating all the sample components, including some procedures that are not specific to ASC X12 PM. This chapter gives detailed, instructional procedures and more specific examples.

Table 6 compares the purposes and tasks of the two approaches.

Table 6 Comparing Quick Start and Tutorial

Approach	Purpose	Tasks
Quick Start	This “load and go” method provides the quickest route to seeing ASC X12 PM in action with eXchange.	Import the sample Projects; create the DPs and build/deploy the Projects; view initial results; run the sample scenario; experiment with passing and monitoring message data.
Tutorial: Chapter 6	This “up close and detailed” method provides complete steps for creating, configuring, and monitoring the working ASC X12 PM business scenario provided in the sample.	Create the necessary Environments and Projects; add and configure all components to be used (for example, OTDs, ASC X12 PM components, BPs, and Connectivity Maps), build/deploy the Projects; view initial results; experiment with passing and monitoring message data.

If you use both the quick-start and tutorial approaches, do the procedures contained in this chapter *first*. In this way, you can use this chapter as a general guide and reference to gain essential background knowledge, before you begin the tutorial in [Chapter 6](#).

5.1.2 Overview of Basic Setup Steps

The basic setups steps, after installation, for setting up, running, and monitoring the sample implementation scenario provided in this chapter are:

- [Getting Started](#) on page 57
- [Constructing the Environments](#) on page 59
- [Using Deployment Profiles](#) on page 61
- [Importing Files for ePM](#) on page 62

For complete information on how to use eXchange, see the *eXchange Integrator User’s Guide*.

5.2 Atlanta and Berlin: Sample Business Scenario

The sample ASC X12 PM implementation scenario demonstrates inbound and outbound message processing between the following parties:

- Atlanta Company
- Berlin Company

In the sample’s business scenario, each company has an eXchange installation, and the two companies trade data.

5.2.1 Sample Scenario Business Description

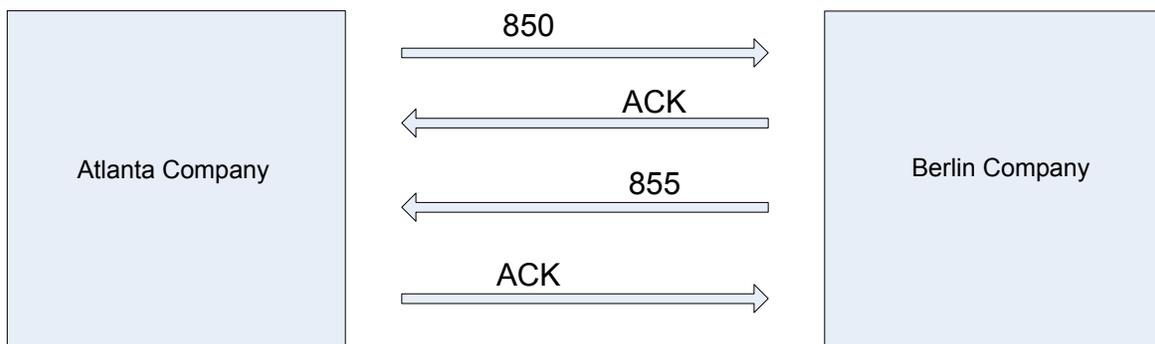
This sample scenario and its Projects demonstrate the configuration of eXchange to support ASC X12. The scenario involves the ASC X12 PM and two TPs, an Atlanta company and a Berlin company. The current viewpoint is assumed to be Atlanta. The resulting B2B solution functions as follows:

- X12 850 Purchase Order Request payloads are read from a local (internal) file.
- The individual payloads are wrapped in the inner and outer ASC X12 envelopes and sent to the Berlin Trading Partner.
- Berlin replies with a TA1 and 997 Functional Acknowledgments.
- Berlin delivers an 855 Purchase Order Response and receives TA1 and 997 acknowledgments.
- eXchange prepares and writes the individual response payloads to an internal file and displays the messages in Message Tracking.

You can change the sample’s scenario to reverse the companies’ sender and receiver roles, if you want (see [Chapter 6](#) for more details). See [Using Message Tracking](#) on page 99 for details on this feature.

Figure 11 shows a diagram of the ASC X12 PM sample’s basic operation.

Figure 11 Sample Scenario Diagram



For a more detailed diagram, see [Figure 12 on page 65](#).

5.2.2 Sample Scenario Projects

The sample scenario is installed with the ASC X12 PM product and contains sample Projects, available upon first use of Enterprise Designer. You may locate these components in the Enterprise Designer’s **Project Explorer**.

The scenario utilizes the following Projects under **eXchange** in **Project Explorer**:

- **B2BHosts**
- **Deployment**
- **GUI**
- **Error**
- **Samples > X12**

Important: You must create the Environments to be used by the sample scenario, as explained under **Constructing the Environments** on page 74.

For a list of files used by these Projects, see **Exporting Sample Files** on page 58.

5.3 Getting Started

This section explains basic information you need to begin using the sample implementation scenario.

Note: See the *eXchange User's Guide* for more information on the subject matter covered by this section.

5.3.1 Before You Start

Before you start using the sample Projects, ensure you have completed the following tasks:

- Finish the installation, as explained in **Chapter 3**.
- Make sure your LDAP and Oracle systems are installed, configured, and operating correctly.
- Your Repository must be running.
- You must have two Domains installed and running, one for Atlanta (**dmnA**) and one for Berlin (**dmnB**).
- Using Enterprise Manager (port 15000), you must add two Integration Servers for the Domains, **dmnA** on 18000 for Atlanta and **dmnB** on 28000 for Berlin (**Host Name: localhost; User Name: Administrator; Password: STC**).

- Also for **dmnA**, add a user with the following parameters:
 - ♦ **username: userA**
 - ♦ **password: userA**
 - ♦ **Group List: PartnerManager, MessageTracking**
- For **dmnB**, add a user with the following parameters:
 - ♦ **username: userB**
 - ♦ **password: userB**
 - ♦ **Group List: PartnerManager, MessageTracking**
- You must be logged on to Enterprise Designer.

Note: If your Repository already has a Project at the root level whose name is identical to any of the Projects you are importing, you must delete or rename such Projects before you start.

5.3.2 Exporting Sample Files

Files are supplied with the sample scenario, which support the general operation of the sample. These files are for the transport of data and for the operation of ePM. You must export these files using Project Explorer in Enterprise Designer.

Note: For a list of the sample scenario's Projects, see [Sample Scenario Projects](#) on page 56.

These files are:

- Data
- ePM

Exporting Data Files

The data files are:

- For Atlanta under **eXchange > Samples > X12 > RecvFromInt > Files:**
 - ♦ **850-SendingOutbound.dat.~in**
 - ♦ **850.dat.**
- For Berlin under **eXchange > Samples > X12 > 855_FromInt_850 > Files:**
 - ♦ **X12_4010_855_template.st**
 - ♦ **X12_dlg_850_In_Atlanta_850_In.xml**

It is recommended that you set up an export folder structure to contain these files like:

```
C:\temp\exchange\Sample\X12\Data\Atlanta Or Berlin
```

The ePM export files are located in **Project Explorer** under **eXchange**, as follows:

- **ePMImport > X12 > Hosts > envA_X12.exp** and **envB_X12.exp**

- **ePMImport > X12 > Scheduler > envA_X12_S1.exp** and **envB_X12_S1.exp**
- **ePMImport > X12 > TP_Profiles > envA_X12_TP_Berlin.exp** and **envB_X12_TP_Atlanta.exp**

It is recommended that you set up an export folder structure to contain these files like:

`C:\temp\exChange\Sample\X12\TP_Profiles` OR `B2B Hosts` OR `Schedules`

Exporting ePM Files

The Atlanta ePM files are:

- **envA_ASC X12**: For the B2B Host.
- **envA_X12_S1.exp**: For the Schedule.
- **envA_X12_TP_Berlin.exp**: For the **Berlin** TP.

The Berlin ePM files are:

- **envB_ASC X12**: For the B2B Host.
- **envB_X12_S1.exp**: For the Schedule.
- **envB_X12_TP_Atlanta.exp**: For the **Atlanta** TP.

5.3.3 Editing the Sample Data Files

Files are supplied with the sample scenario, which reference path locations enclosed between the XML tags: `<dir> ... </dir>`. You must edit these files to reflect the location where the data files are to be used by the Atlanta and Berlin systems.

The files you need to edit are:

- For Atlanta:
 - ♦ **850-SendingOutbound.dat.~in**
- For Berlin
 - ♦ **X12_dlg_850_In_Atlanta_850_In.xml**

The previous section lists the locations of where to find these files, if you have not already exported them. If you need information on exactly how to edit these files, see [Editing the Sample Data .xml Files](#) on page 71.

If you have already run the sample and you want to experiment with other differences from the sample (such as using a payload data file with a different file name, or using a TP with a different name), be sure these differences are also reflected in these files, as necessary.

5.4 Constructing the Environments

This section contains Enterprise Designer procedures for constructing the required Environments for the sample scenario. Make sure that properties you configure under

these procedures match your system's configuration, including system configurations you must check before you start (see [Getting Started](#) on page 57).

Before You Begin

- Make sure you have completed all the operations, as explained under [Before You Start](#) on page 57.

You must create these two Environments:

- For Atlanta: **envA**
- For Berlin: **envB**

5.4.1 Creating External Systems

Create, construct, and if necessary, configure the following external systems for **envA** and **envB**:

- **esOracle**
- **esBLF**
- **esHTTP**
- **esHTTPserver**
- **esFileA** (for Atlanta) and **seFileB** (for Berlin)
- **esLDAP**
- **esB2BService**

5.4.2 Configuring External Systems

Using their **Properties** dialog boxes, configure the following external systems for **envAtlanta**, as necessary for your setup:

- Oracle
- LDAP

Make sure to configure the File eWays, as necessary for your system:

- **esFileA**
- **esFileB**

Using their **Properties** dialog boxes, configure the following additional components, as necessary for your setup:

- B2B Configuration Service
- JMS settings
- Sun SeeBeyond Integration Server

For additional information on these and additional necessary setup operations, see [Setting up the Environments](#) on page 74.

5.5 Using Deployment Profiles

This section describes how to build and deploy the Projects' Deployment Profiles in the ASC X12 PM sample scenario. Building a Deployment Profile creates the application .ear file for the Project. After creating this file, you must deploy it for all Deployment Profiles *except* the B2B Host.

For more information on these operations, see [Constructing the Projects](#) on page 81.

5.5.1 Locating the Projects

On the **Project Explorer** tree, you may open the sample scenario's Projects to display their components.

Note: Make sure you have all of the Projects, as listed under [Sample Scenario Projects](#) on page 56.

5.5.2 Deploying the Deployment Profiles

This section provides a procedure that describes how to construct and deploy the sample scenario's Project Deployment Profiles.

To construct and deploy the Deployment Profiles

- 1 Open the Project you want to work with, using **Project Explorer**.
- 2 Create and name one or more Deployment Profiles for each Project, as shown in [Table 7 on page 88](#).
- 3 Automap and deploy each Deployment Profile, except that you do *not* deploy the Deployment Profiles for the Host Project.

Note: Make sure the B2B Host Deployment Profiles create an instance of the eXchange Service for each current Deployment Profile and Environment (one each for Atlanta and one each for Berlin).

- 4 Make sure you click **Save All** after you are finished with each individual operation.

5.5.3 Special Considerations for Deployment Profiles

Ensure that you take the following necessary considerations into account while constructing your Deployment Profiles:

- Do *not* deploy the B2B Host Deployment Profiles.
- Be sure that you create the validation Connectivity Map for the eXchange Deployment Project's Deployment Profiles; see [Creating the Validation Connectivity Map](#) on page 83 for details.
- Be sure that you include the correct path location in **bp855**; see [Updating the bp855 Business Process](#) on page 86 for details.

5.6 Importing Files for ePM

This section explains ePM procedures for importing the sample scenario files supplied for the B2B Hosts and Trading Partners.

Note: For a general description of the outbound and inbound messaging **ToPartner** and **FromPartner** model used by ePM, see [Configuring ePM: ToPartner and FromPartner Messaging Model](#) on page 44.

If you need more detailed procedures for any of the operations described under this section, see [Importing and Configuring Components in ePM](#) on page 88.

5.6.1 Running the ePM Interface

Start running ePM as explained under [Running ePM](#) on page 89.

5.6.2 Importing B2B Hosts

Your next step is importing the following B2B Host files:

- **envAtlanta_X12**: For Atlanta.
- **envBerlin_X12**: For Berlin.

5.6.3 Importing Schedules

Next, you must import the following Schedule files for the B2B Hosts.

- **envA_X12_S1.exp**: For Atlanta.
- **envB_X12_S1.exp**: For Berlin.

5.6.4 Importing Trading Partners

Next, you must import the TPs for the B2B Hosts.

- **envB_X12_TP_Atlanta.exp**: For the **Atlanta** TP (**envB** for Berlin)
- **envA_X12_TP_Berlin.exp**: For the **Berlin** TP (**envA** for Atlanta)

5.7 Running the Sample Scenario

This section explains how to run the sample scenario and transport data between the two TPs in the scenario, Atlanta and Berlin.

To transport data between the TPs

- 1 Locate the folder on the machine running the domains, where you have stored the data to be transported, for example, `C:\temp\exchange\Sample\X12\Data`.

- 2 Rename file **850-SendingOutbound.dat.~in** located in the ... X12\Data\Atlanta\ folder to **850-SendingOutbound.dat.in**.

The message transport operation occurs.

- 3 Check Message Tracking (see [Figure 25 on page 100](#)) and make sure you see the following messages:
 - ♦ <Transaction ID> Berlin X12 PurchaseOrder_AG <Dialog ID> 855 Inbound
 - ♦ <Transaction ID> Berlin X12 PurchaseOrder_AG <Dialog ID> 850 Outbound

Result: If you are able to view the previous messages in Message Tracking, the sample scenario is running correctly and the data transport operation is successful.

Monitoring Messages

You may monitor overall message activity using the eXchange Message Tracking features. See [Using Message Tracking](#) on page 99 for information on how to access and begin using this feature.

ASC X12 PM Sample Scenario Tutorial

This chapter provides a basic ASC X12 PM tutorial, explaining how to create and implement a sample scenario, as well as how you can use eXchange to achieve B2B solutions using the ASC X12 protocol.

What's in This Chapter

- [Using This Tutorial](#) on page 64
- [Preconfiguration for Atlanta and Berlin Environments](#) on page 66
- [Editing the Sample Data .xml Files](#) on page 71
- [Constructing the Environments](#) on page 74
- [Constructing the Projects](#) on page 81
- [Importing and Configuring Components in ePM](#) on page 88

6.1 Using This Tutorial

This chapter provides a tutorial for the ASC X12 PM sample implementation scenario and detailed procedures that describe how to construct, run, and monitor the sample, which employs the Atlanta and Berlin Companies described in [Chapter 5](#).

6.1.1 Quick Start and Tutorial Approaches

See [Using the Quick Start Procedures](#) on page 54 for detailed instructions on how to use the quick-start and tutorial approaches to implementing the sample scenario. In summary, the instructions are:

- You can use [Chapter 5](#) for a quick, general overview of the basic setup steps needed to import, set up, run, and monitor the sample ASC X12 PM eXchange Projects.
- You can use this chapter as a step-by-step, more detailed tutorial approach to the same sample implementation.

Important: *Many of the procedures in this chapter must be done for both the Atlanta and Berlin companies. Unless otherwise stated, procedures are given once, with dual branching where procedures for the two Environments differ. It is recommended that you use a given procedure to create the Atlanta component (for example, the Environment) first, then use the procedures again for Berlin, with the appropriate substitutions.*

6.1.2 Introduction to the Sample Implementation

To perform the sample scenario implementation, you need to set up the sample Projects, their Environments, and components using the eGate Enterprise Designer with eInsight and eXchange. See the major heading overviews in [Chapter 5](#) for summary explanations of ASC X12 PM implementation.

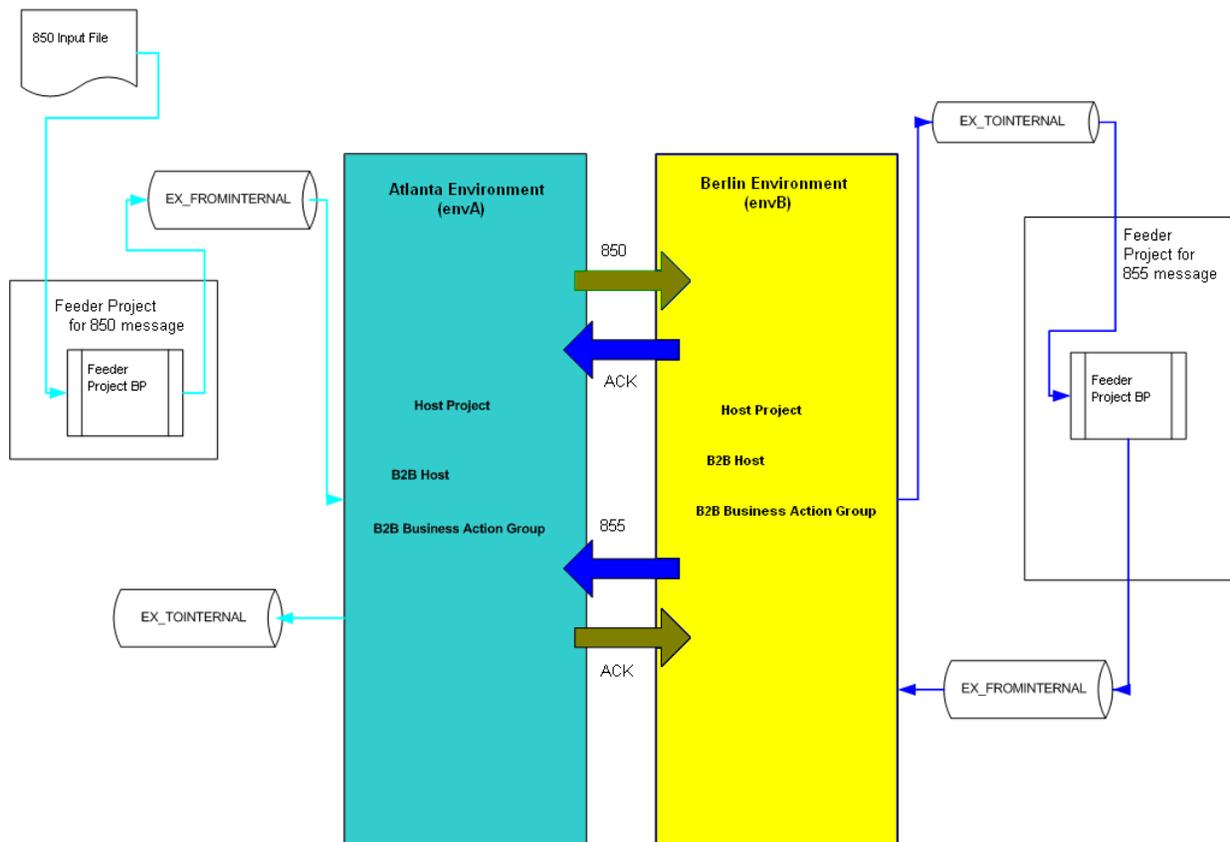
[Atlanta and Berlin: Sample Business Scenario](#) on page 55 contains a description of the sample scenario and how it operates, including the eXchange solutions provided for the business problems contained in the sample.

For more information solving business problems using eXchange with eInsight and eGate, including additional details on implementation, see the *eXchange Integrator User's Guide*, *eInsight Business Process Manager User's Guide*, and *eGate Integrator User's Guide*.

6.1.3 Operational Diagram

Figure 12 shows an operational diagram of the sample scenario.

Figure 12 Sample Scenario Operation



6.1.4 Server Configurations

The sample assumes you use default configurations for all servers, where possible, and that you make any changes in Enterprise Designer, where needed, for example:

- **Oracle:** You must create a new outbound Oracle external system instance for each Environment and configure it for your system, even if you imported the sample Environments. Sample parameters are for reference only. Any Oracle database used by eXchange must be accessible to eGate, and you must know its Oracle SID, user name, and password. Create and configure the eXchange database using the **Oracle510.zip** file. For more information see the *Oracle eWay Adapter User's Guide*, for eWay settings and Integration Server configuration
- **LDAP:** You must create a new outbound LDAP external system instance for each Environment and configure it for your system, even if you imported the sample Environments. Sample parameters are for reference only. Any LDAP application used by eXchange must be accessible to eGate. For more information see the *LDAP eWay Adapter User's Guide*, for eWay settings and Integration Server configuration
- **HTTPS:** For information on how to configure your HTTP server or client to use SSL, see the *HTTP(S) eWay Adapter User's Guide*, for eWay settings and Integration Server configuration.

6.2 Preconfiguration for Atlanta and Berlin Environments

This section explains the preconfiguration operations you must perform for the Atlanta and Berlin Environments.

6.2.1 Creating and Starting the Domains

This section explains how to create and start the Logical Host Domains for the Atlanta and Berlin Environments.

See the *eGate Integrator User's Guide* for more information on eGate Logical Hosts, Domains, and the Domain Manager feature.

To create the Atlanta Domain

- Create a new domain (under the **\logicalhost** directory, that is, the Logical Host) by running the following script:

```
C:\ ... logicalhost\createdomain --dname dmna
```

This script creates the domain name **dmna** and retains the default ports 1800x.

To start the Atlanta Domain

- With the Repository running, start (if not already started) the newly created Domain as follows:

```
C:\ ... logicalhost\start_dmna.bat
```

Note: Starting the first domain can require approximately 7 minutes.

To create the Berlin Domain

- Create another new domain by running the following script:

```
C:\ ... logicalhost\createdomain --dname dmnB --startingport 28000
```

This script creates the domain name **dmnB** and designates the default ports 2800x.

To start the Berlin Domain

- With the Repository running, start (if not already started) the newly created Domain as follows:

```
C:\ ... logicalhost\start_dmnB.bat
```

You may use the eGate Domain Manager interface to make sure the new Domains are started and running.

6.2.2 Adding a New User to ePM and Message Tracking

You must add two new ePM users via the eGate Integration Server Security Gateway, one for Atlanta and one for Berlin. Doing this operation means using the set of procedures in this section twice, with appropriate changes.

To add a new user to the ePM and Message Tracking Groups

- 1 With the Repository running, start a new browser session and point it at the following URL:

```
http://localhost:18000
```

For Berlin, use:

```
http://localhost:28000
```

- 2 Log in to Integration Server Security Gateway using **username/password = Administrator/STC**.

- 3 In the Integration Server Administration window, click the **User Management** tab. See Figure 13.

Figure 13 Integration Server Administration Window

Integration Server Administration

Configuration Agent | User Management

Current realm for user management is

[Users List](#) > [Add/Edit User](#)

Specify the details for this User.

User Name:*

Password:*

Confirm Password:*

Group List:

Separate multiple groups with commas.

- 4 In the **User Management** tab, click **Add New User** and supply the required values, for example:
 - ◆ **User Name:** **userA** (Atlanta); **userB** (Berlin)
 - ◆ **Password:** **userA** (Atlanta); **userB** (Berlin)
 - ◆ **Confirm Password:** **userA** (Atlanta); **userB** (Berlin)
 - ◆ **Group List:** **PartnerManager,MessageTracking**

This step provides the following user privileges:

- ◆ The **PartnerManager** role allows the specified user to log in to and use ePM.
- ◆ The **MessageTracking** role allows the specified user to use the Message Tracking Web client.

In this example, both roles are granted to a newly created user named **userA**. If you prefer, you can set up several users that have one privilege or the other, or both. Or, instead of creating new users, you can confer **PartnerManager** and/or **MessageTracking** privileges upon an existing user, such as **Administrator**.

- 5 When you are finished, click **Submit**.

- 6 Log out of Integration Server Administration and close the window.
- 7 Repeat these procedures for Berlin, using the appropriate changes.

6.2.3 Adding the Application Server Instances

You must add two new instances of the Application Server via the eGate Enterprise Manager, one for Atlanta and one for Berlin. Doing this operation means using the set of procedures in this section twice, with appropriate changes.

To add two new Application Server instances

- 1 With the Repository running, install Enterprise Manager by running the following script:

```
C:\ ... \emanager\install.bat
```

- 2 In the installation wizard, follow the prompts and accept the license agreement and default port (15000).

- 3 After the installation is complete, start Enterprise Manager server by running the following script:

```
C:\ ... \emanager\startserver.bat
```

- 4 Start a new browser session and point it at the following URL:

```
http://localhost:15000
```

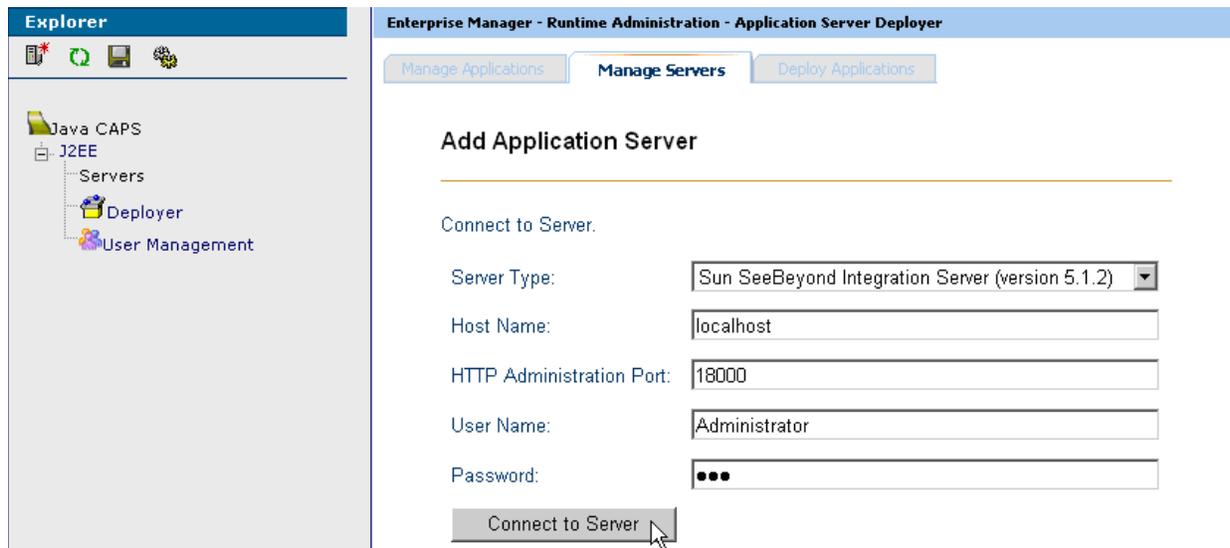
- 5 Log in to Enterprise Manager using **username/password = Administrator/STC**.

- 6 Click **J2EE** then, in the **Manage Servers** tab, add a new Application Server for Atlanta using the following settings:

- ♦ **Server Type: Sun SeeBeyond Integration Server**
- ♦ **Host Name: localhost**
- ♦ **Administration Port: 18000**
- ♦ **User Name: Administrator**
- ♦ **Password: STC**

See Figure 14.

Figure 14 Enterprise Manager Window



- 7 Click **Connect to Server**.
- 8 Use these same procedures to add another Application Server instance for Berlin, entering **28000** for the **HTTP Administration Port**.
- 9 Save your changes and exit the window.

6.2.4 Initializing and Running Enterprise Designer

This section describes operations it is recommended that you perform upon, and directly after, initializing and running Enterprise Designer.

Note: Using Enterprise Designer, you must make sure to increase the *eDesigner_heap_size* property to **1024**. For more information, see the *eXchange Integrator User's Guide*.

To initialize and run Enterprise Designer

- 1 With the Repository running, start Enterprise Designer by running the following script:

```
C:\ ... \edesigner\bin\runed.bat
```
- 2 In the installation wizard, accept the license agreement.
- 3 Log in to Enterprise Designer using **username/password = Administrator/STC**.
- 4 On the **Tools** menu, click **Update Center**.

- 5 In the **Update Center** wizard, follow the steps to check for updates, and to add all available updates and new modules.
- 6 When you are done, restart Enterprise Designer (referred to as IDE in the user interface).

Note: For more information, see the *eGate Integrator User's Guide*.

Final Result: You have now finished preparing eGate to run the sample scenario.

6.3 Editing the Sample Data .xml Files

Data files are supplied with the sample scenario, which reference a path enclosed between the XML tags: `<dir> ... </dir>`. You must first export these files using Enterprise Designer's **Project Explorer**. Then, edit these files to reflect the path location where you actually export the sample scenario files.

Note: For a list of the sample scenario's files, see [Exporting Sample Files](#) on page 58.

To export the sample data files

- 1 Locate the export files for the Atlanta sample data in Enterprise Designer's **Project Explorer** under **eXchange > Samples > X12 > RecvFromInt > Files**

These files are:

- ♦ **850-SendingOutbound.dat.~in**
- ♦ **850.dat.**

- 2 Export these files to a folder on your C drive. It is recommended that you set up a folder structure to contain these files, under `C:\temp`, for example:

```
C:\temp\exChange\Sample\X12\Data\Atlanta
```

- 3 Locate the export files that ensure Berlin returns the correct messages to Atlanta in Project Explorer, under **eXchange > Samples > X12 > 855_FromInt_850 > Files:**

These files are:

- ♦ **X12_4010_855_template.st**
- ♦ **X12_dlg_850_In_Atlanta_850_In.xml**

- 4 Export these files to a folder on your C drive. It is recommended that you set up a folder structure to contain these files, under `C:\temp`, for example:

```
C:\temp\exChange\Sample\X12\Data\Berlin
```

You must make sure that the **850-SendingOutbound.dat.~in** file is updated to reflect the appropriate Atlanta data path location. You may do this operation using a text editor.

To edit the Atlanta 850-SendingOutbound.dat.~in file

- 1 Make sure you have exported the sample data files.

- 2 Use the Enterprise Designer's Export feature to export the sample data files to a specified location, for example:

```
cd C:\temp\exchange\Sample\X12\Data\Atlanta
```

- 3 Change directories to the subdirectory of the location where you exported the sample data files.
- 4 Use a text editor open the following file:

850-SendingOutbound.dat.~in

You see text that resembles the following text:

```
<?xml version="1.0" encoding="UTF-8"?><!--Sample XML file
generated by XMLSPY v2004 rel. 3 U (http://www.xmlspy.com)--
><TestInput xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\X12Sample\xsd\ServiceInput.xsd">
<dir><path></dir>
<filename>850.dat</filename>
<tradingPartner>Berlin</tradingPartner>
<service>PurchaseOrder_AG_Profile</service>
<action>850</action>
<createnumberofmsgs>1</createnumberofmsgs>
</TestInput>
```

The lines preceding the last line `</TestInput>` have the following meaning:

- ♦ The `<dir>...</dir>` line supplies the path of the directory that holds the payload data file to be processed.
 - ♦ The `<filename>...</filename>` line supplies the file name of the payload data file for the current transaction.
 - ♦ The `<tradingpartner>...</tradingpartner>` line supplies the name of the current TP.
 - ♦ The `<service>...</service>` line supplies the name of the Transaction Profile Group of the current TP; used in ePM.
 - ♦ The `<action>...</action>` line supplies the transaction number, in this case, 850.
- 5 If necessary, in the line `<dir><path></dir>` (shown in the previous example), change the string represented by `<path>`, to the actual path of the directory that holds your local copy of the `850.dat` file.

For example:

```
<dir>C:\temp\exchange\Sample\X12\Data\Atlanta</dir>
```

Or:

```
<dir>/~myname/exchange/Sample/X12/Data/Atlanta</dir>
```

- 6 If you have already run the sample and you want to experiment with other changes (such as using a payload file with a different file name, or using a TP with a different name), be sure they are also reflected here.
- 7 When you are finished, save your changes and exit the text editor.

You must make sure that the `Berlin X12_dlg_850_In_Atlanta_850_In.xml` file is updated to reflect the appropriate Berlin data path location. You may do this operation using a text editor.

To edit the Berlin X12_dlg_850_In_Atlanta_850_In.xml file

- 1 Make sure you have exported the sample data files.
- 2 Use the Enterprise Designer's Export feature to export the sample data files to a specified location, for example:

```
cd C:\temp\exchange\Sample\X12\Data\Berlin
```

- 3 Change directories to the subdirectory of the location where you exported the sample data files.
- 4 Use a text editor open the following file:

♦ **X12_dlg_850_In_Atlanta_850_In.xml**

You see text that resembles the following text:

```
< ?xml version="1.0" encoding="UTF-8"? >
<FileAndService>
  <TradingPartner>Atlanta</TradingPartner>
  <TPProfileID></TPProfileID>
  <Service>PurchaseOrder_Inb_AG</Service>
  <Action>855</Action>
  <Files Directory="<path>">
    <Name>X12_4010_855_template.st</Name>
  </Files>
</FileAndService>
```

- 5 If necessary, in the line `<dir><path></dir>` (shown in the previous example), change the string represented by `<path>`, to the actual path of the directory that holds your local copy of the `X12_dlg_850_In_Atlanta_850_In.xml` and `X12_4010_855_template.st` files.

For example:

```
<dir>C:\temp\exchange\Sample\X12\Data\Berlin</dir>
```

Or:

```
<dir>/~myname/exchange/Sample/X12/Data/Berlin</dir>
```

- 6 If you have already run the sample and you want to experiment with other changes (such as using a payload file with a different file name, or using a TP with a different name), be sure they are also reflected here.
- 7 When you are finished, save your changes and exit the text editor.

Note: For more information, see the *eXchange Integrator User's Guide*.

Result: You have now finished preparing the data files to run within the sample scenario.

6.4 Constructing the Environments

In implementing ASC X12 PM Projects, you must set up at least one Environment for each eXchange installation. The sample scenario is set up to operate on one machine but mimic two TPs. As a result, for the sample scenario, you need to construct the Environments as explained under the following sections:

- [Using Environment Explorer](#) on page 74
- [Setting up the Environments](#) on page 74

6.4.1 Using Environment Explorer

You perform these operations using Enterprise Designer's **Environment Explorer** and its canvas windows. The sample Environments contain the following types of components:

- Instances for external systems accessed by eWays
- Instance for the B2B Service Configurator external system
- Instance for the Logical Host

For example, the Oracle external system must be configured to reference your Oracle setup. Other external systems (for example, the File and Batch eWays) have configurations that may differ depending on your system setup, and so forth. Also, you may be using nonstandard ports or user name/password combinations.

The remainder of this section describes the procedures to construct the sample scenario's Environments.

Note: Before you begin, make sure you have followed the preliminary instructions provided under [Getting Started](#) on page 57.

6.4.2 Setting up the Environments

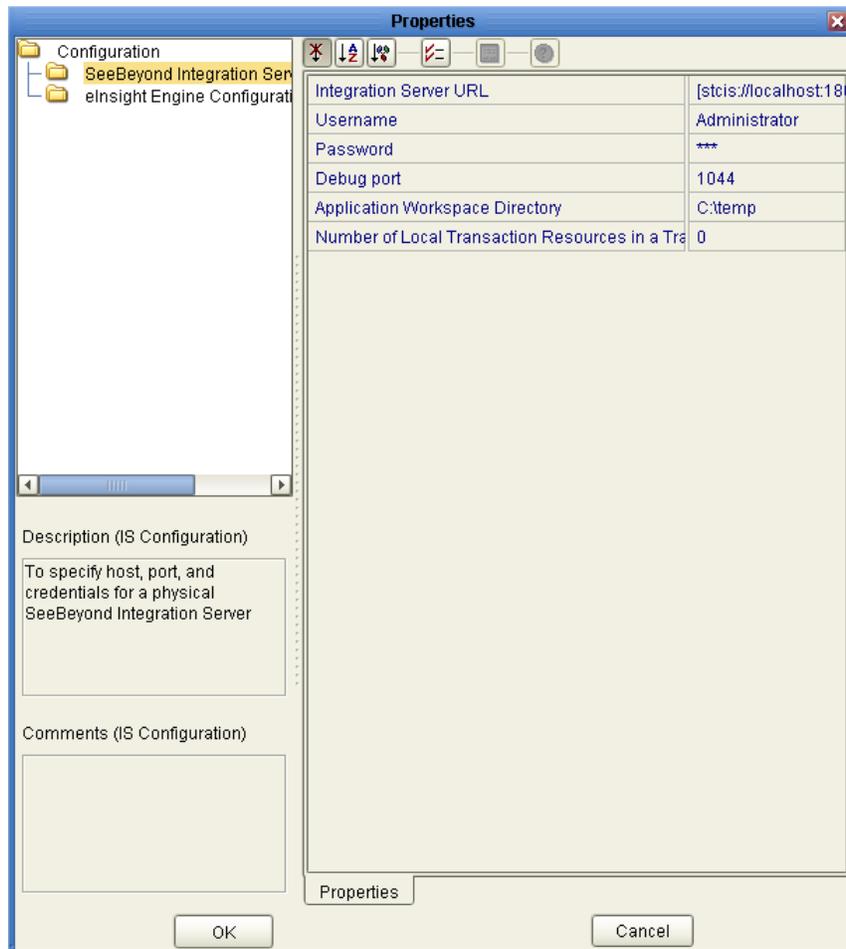
This section explains how to create the sample's Environments for Atlanta and Berlin. Use the procedures to set up Atlanta first, then Berlin.

To create the basic components

- 1 On Enterprise Designer, near the lower left of the window, click the **Environment Explorer** tab.
- 2 On the **Environment Explorer** tree, right-click the Repository and, on the context menu, click **New Environment**.
- 3 Name the newly created Environment **envA** (**envB** for Berlin).
- 4 Right-click **envA** (**envB** for Berlin) and, on the menu, click **New Logical Host** and name the Logical Host **lhA** (**lhB** for Berlin).
- 5 Right-click **IntegrationSvr1** and select **Properties** from the context menu.

The **Properties** dialog box for **IntegrationSvr1** appears. See Figure 15.

Figure 15 Integration Server Properties Dialog Box: Environment



- 6 For **lhA > IntegrationSvr1** on Environment Explorer, set the **Configuration > SeeBeyond Integration Server** properties as follows:
 - ♦ **Integration Server URL:** Points to the Integration Server, for example:
 - ♦ `http://localhost:18000` (for **envA**)
 - ♦ `http://localhost:28000` (for **envB**)
 - ♦ **Username:** Administrator
 - ♦ **Password:** STC (masked)
 - ♦ **Debug port:** 1044
 - ♦ **Application Workspace:** Blank, for this sample.
 - ♦ **Number of Local Transactions:** 0

- 7 For the rest of the **IntegrationSvr1** properties settings, accept the defaults.
- 8 Create a Sun SeeBeyond JMS IQ Manager under **IntegrationSvr1** and name it **SBJMSIQMgr1**.

To create and configure the Oracle external system

- 1 On Enterprise Designer, on the **Environment Explorer** tree, right-click **envA** (**envB** for Berlin) and, on the context menu, click **New Oracle External System**.
- 2 Name the new component **esOracle** and click **OK**.

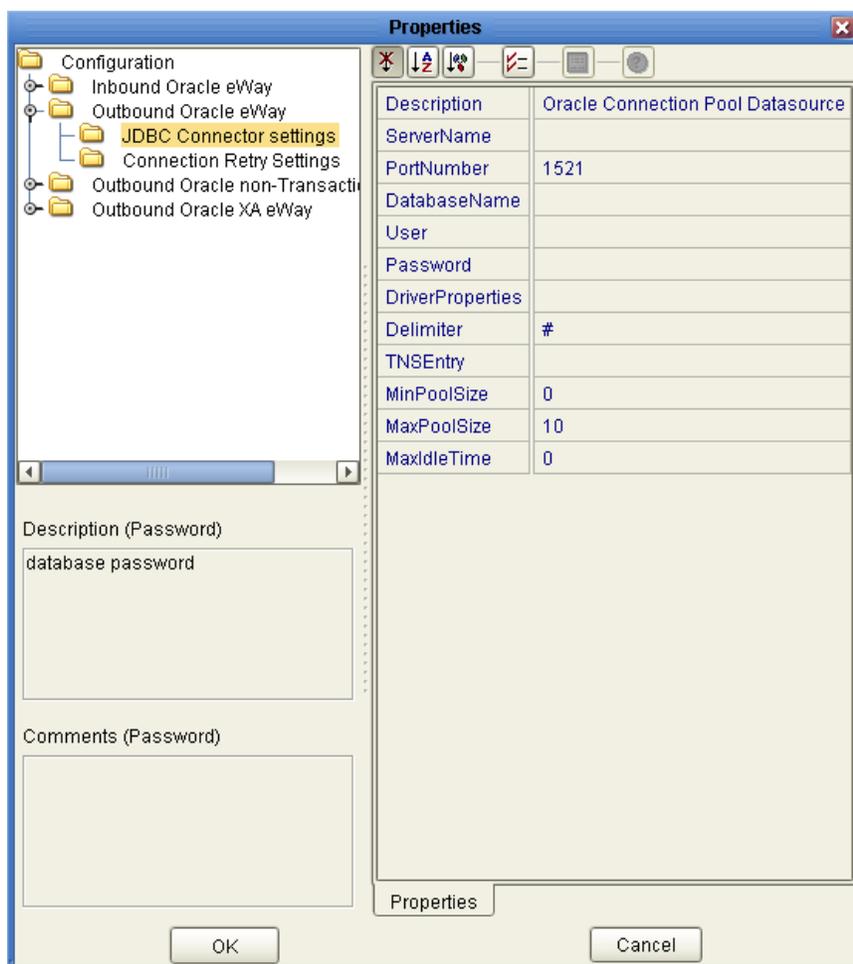
These actions create, for the current Environment, an external system instance for the Oracle eWay in outbound mode.

Note: The eXchange database uses Oracle. For more information on Oracle requirements for eXchange and ASC X12 PM, see the Readme file that accompanies ASC X12 PM.

- 3 Right-click **esOracle** and select **Properties** from the context menu.

The **Properties** dialog box for the external system appears. See Figure 16.

Figure 16 Oracle External System Properties Dialog Box: Environment



- 4 Configure the **Configuration > Outbound Oracle eWay > JDBC Connector settings** properties for **esOracle** as follows:
 - ◆ **Description: Oracle Connection Pool Datasource**
 - ◆ **ServerName:** Host name of the Oracle server machine
 - ◆ **PortNumber: 1521** (change this value only if your Oracle system administrator changed the default)
 - ◆ **DatabaseName:** SID for your current Oracle system
 - ◆ **User:** Valid user ID for the current Oracle system
 - ◆ **Password:** Valid password for the current Oracle system (masked)
 - ◆ **Driver Properties:** Blank, for this sample
 - ◆ **Delimiter:** #
 - ◆ **TNS Entry:** Blank, for this sample
 - ◆ **MinPoolSize: 0**
 - ◆ **MaxPoolSize: 10**
 - ◆ **MaxIdleTime: 0**
- 5 When all properties have been configured correctly for your site, click **OK**.

To create and configure the LDAP external system

- 1 Create a new LDAP eWay (**New > LDAP External System**) under **envA** (**envB** for Berlin) and name it **esLDAP**.
- 2 Set the **Environment Configuration > Connection** properties for **esLDAP** as follows:
 - ◆ **Initial Context Factory:** **com.sun.jndi.ldap.LdapCtxFactory**
 - ◆ **Provider URL:** As necessary for your system, according to the provided syntax
 - ◆ **Authentication:** **simple**
 - ◆ **Principal:** As necessary for your system
 - ◆ **Credentials:** Password, as necessary for your system

Note: A separate LDAP external system instance is required for each B2B Host.

- 3 For all other LDAP eWay (**esLDAP**) properties, accept the defaults.

To create and configure the B2B Configurator Service external system

- 1 Create a B2B Service Configurator (**New > B2B Configurator Service**) external system under **envA** and name it **esB2BService**.

- 2 Set the **environment-configuration > Database Settings** properties for **esB2BService** as follows:
 - ♦ **Type:** Oracle
 - ♦ **URL:** Points to the eXchange database; an example of the URL syntax is: **jdbc:oracle:thin:@<hostname>:<port>:exchange**
 - ♦ **UserName:** Valid user ID for the current Oracle system
 - ♦ **Password:** Valid password for the current Oracle system (masked)
- 3 Set the **environment-configuration > JMS Settings** properties for **esB2BService** as follows:
 - ♦ **JMS Server URL:** Points to the IQ Manager port. The **envA** Logical Host is on 18000 (**envB** on 28000), the IQ Manager port for **envA** is 18007 (28007 for **envB**). This port number is listed in the Domain Manager for the current Logical Host; an example of the URL syntax is: **stcms://<hostname>:<port>**
 - ♦ **Security Principal:** Administrator
 - ♦ **Security Credentials:** STC (masked)
 - ♦ **Connection Factory:** connectionfactories/topicconnectionfactory
 - ♦ **Resend Topic:** topics/EX_TODELIVERY
 - ♦ **Timeout Topic:** topics/EX_ERROR
 - ♦ **Business Protocol Topic for Batching:** topics/EX_BATCHER
 - ♦ **Delivery Protocol Topic for Batching:** topics/EX_DELIVERYBATCHER
- 4 For the rest of the **esB2BService** properties settings, accept the defaults.

To create and configure the File eWays

- 1 Create a new File eWay (**New > File External System**) in inbound mode under **envA** and name it **esFileA**.
- 2 Set the **Configuration > Inbound File eWay > Parameter Settings** property for **esFileA** as follows:
 - ♦ **Directory:** C : /temp/eXchange/Sample/X12/Data/Atlanta

Note: Make sure this folder and the folder for **envB** are correctly configured under the appropriate parameter in ePM. It is recommended that you create an additional folder under X12 named **Errors**.

- 3 Set the **Configuration > Outbound File eWay > Parameter Settings** property (for DLQ and processing errors) for **esFileA** as follows:
 - ♦ **Directory:** C : /temp/eXchange/Sample/X12/Errors/Atlanta
- 4 For all other File eWay (**esFileA** and **esFileB**) properties, accept the defaults.
- 5 Create a new File eWay (**New > File External System**) in inbound mode under **envB** and name it **esFileB**.

- 6 Set the **Configuration > Inbound File eWay > Parameter Settings** property for **esFileB** as follows:
 - ♦ **Directory:** C : /temp/eXchange/Sample/X12/Data/Berlin
- 7 Set the **Configuration > Outbound File eWay > Parameter Settings** property (for DLQ and processing errors) for **esFileB** as follows:
 - ♦ **Directory:** C : /temp/eXchange/Sample/X12/Errors/Berlin

To create and configure the additional external systems

- 1 In Enterprise Designer, on the **Environment Explorer** tree, right-click **envA** (**envB** for Berlin) and, on the context menu, click **New Batch Local File System**.
- 2 Name the new external system **esBLF**, and click **OK**.
These operations create, for the Environment, an external system for the Batch eWay in local file mode.
- 3 In the new external systems' **Properties** dialog boxes and accept the defaults.

Note: When you are finished with each **Properties** dialog box, click **OK**.

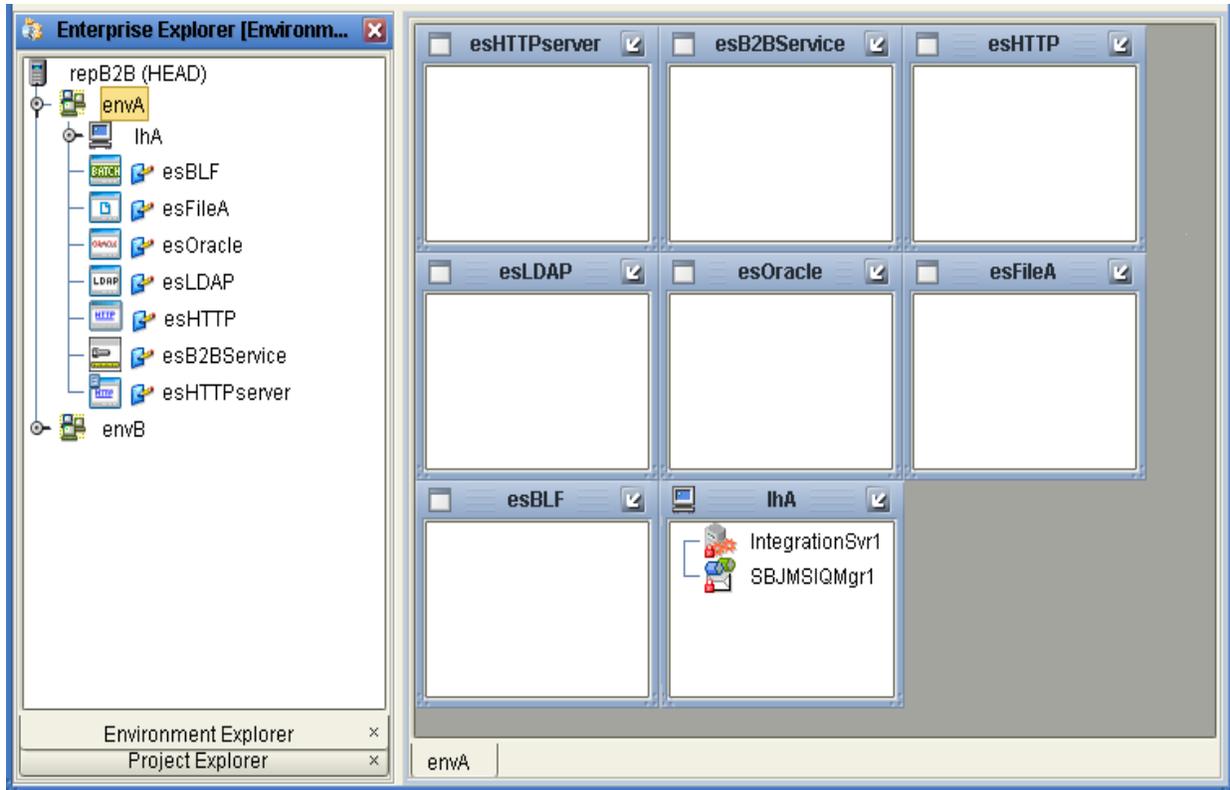
- 4 Create an HTTP eWay (client mode) external system under **envA** (**envB** for Berlin) and name it **esHTTP**.
- 5 Open the **Properties** dialog boxes for **esHTTP** and accept the defaults.
- 6 Create an HTTP eWay (server mode) external system under **envA** (**envB** for Berlin) and name it **esHTTPserver**.
- 7 Open the **Properties** dialog boxes for **esHTTPserver** and accept the defaults.

Note: When you build a B2B Host Deployment Profile, eXchange automatically creates another external system on the chosen Environment. This external system is called an eXchange service. For more information on this service, see the **eXchange Integrator User's Guide**. Also, see **Constructing the B2B Host Project** on page 81.

When You Are Finished for Atlanta

Result: You have set up the Environment for Atlanta, **envA**. See Figure 17.

Figure 17 Sample Scenario's Atlanta Environment



- Collapse the **envA Environment Explorer** tree, click **Save All**, and close all canvases.

When You Are Finished for Berlin

- *Result:* You have set up the Environment for Berlin, **envB**, which appears directly under **envA** on the **Environment Explorer** tree. See [Figure 17 on page 80](#).
- Collapse the **envB Environment Explorer** tree, click **Save All**, and close all canvases.

Final Result: You have now finished setting up the Environments for Atlanta and Berlin, including the external system components to be used by both.

6.5 Constructing the Projects

This section explains how to construct the Projects in the ASC X12 PM in order to run the sample scenario. This operation includes:

- Using installed Projects (in **Project Explorer**) for the sample scenario
- Setting up and configuring Project components
- Associating each Project with the appropriate Environment(s)
- Mapping and building the Project's Deployment Profile
- Deploying the Deployment Profile, if necessary

Building a Deployment Profile creates the application **.ear** file for the Project. After creating this file, you must deploy it for all Deployment Profiles except the B2B Host.

You perform these operations using Enterprise Designer's **Project Explorer** and its canvas windows. The remainder of this section describes the necessary procedures under the following sections:

- [Constructing the B2B Host Project](#) on page 81
- [eXchange Deployment Project](#) on page 83
- [Constructing the 855_FromInt_850 Project](#) on page 85
- [Constructing the Remaining Projects' Deployment Profiles](#) on page 87

6.5.1 Constructing the B2B Host Project

This section explains how to set up, and configure the ASC X12 PM sample scenario's B2B Host Project, **X12Host**. Constructing the B2B Host Project creates an eXchange service that acts as a channel manager and provides a connection to the eXchange database. You must build two Deployment Profiles, one for each company, Atlanta and Berlin. However, you do *not* deploy these Deployment Profiles.

To build the B2B Host's Deployment Profile for Atlanta

- 1 On Enterprise Designer's **Project Explorer** tree, right-click **X12Host** under **eXchange > B2BHosts** and, on the context menu, choose **New** and click **Deployment Profile**.
- 2 In the resulting dialog box, name the new Deployment Profile **dpHost_A**, point it at **envA**, make sure it is using only the **cmX12** Connectivity Map, and click **OK**.

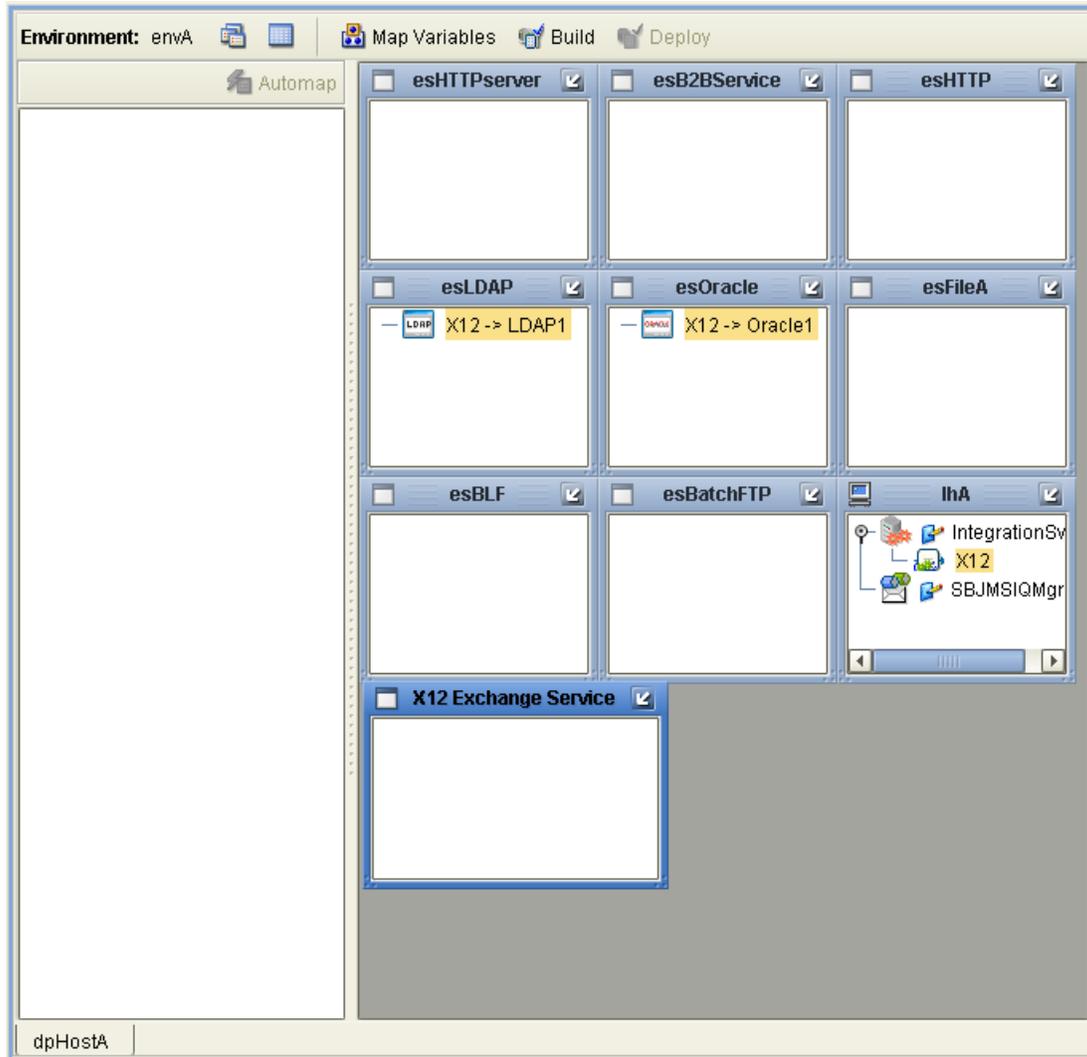
Deployment Editor opens. Its left pane shows the X12 B2B Host instance, the **Oracle1** external application, and the **LDAP1** external application. These are the components created in the Connectivity Map **cmX12**.

The Deployment Editor right pane contains windows representing the Logical Host and external systems created in **envA**.

- 3 Click **Automap** to map the components.

The components in the left pane automatically map to the appropriate windows on the right pane of Deployment Editor for **dpHost_A**. See Figure 18.

Figure 18 Deployment of dpHost_A



- 4 Click **Save**.
- 5 Click **Build** to build the Deployment Profile.

A dialog box appears, indicating the status of the build operation. A new service, X12 eXchange Service, is created and assigned to the current Deployment Profile and Environment.

You may view this X12 eXchange Service on **envA**, in the Environment Explorer tree, as well as in the right pane of Deployment Editor for **dpHost_A**.

Note: Do *not* deploy the B2B Host Project.

If the build operation is not successful, repeat the steps in this procedure, carefully rechecking every action. When the build is successful, go to the next step.

Note: *Building Deployment Profiles for large Projects may take approximately 10 to 15 minutes or more.*

- 6 When you are finished, click **Save All** and close all canvases.

To build the B2B Host's Deployment Profile for Berlin

- 1 On Enterprise Designer's **Project Explorer** tree, right-click **X12Host** under **eXchange > B2BHosts** and, on the context menu, point at **New** and click **Deployment Profile**.
- 2 In the resulting dialog box, name the new Deployment Profile **dpHost_B**, point it at **envB**, make sure it is using **cmX12**, and click **OK**.
- 3 In Deployment Editor, click **Automap** to map the components.
- 4 Click **Save**.
- 5 Build the Deployment Profile for Berlin in the same way as you did for Atlanta, except make sure to use **envB** and **cmHostB**.

The X12 eXchange Service is created and assigned to the current Deployment Profile and Environment. Do not deploy the B2B Host Project. Once more, if there are any errors, troubleshoot until you are ready to go to the next step.

- 6 When you are finished, click **Save All** and close all canvases.

Result: You have now finished constructing the B2B Host Project, including creating, mapping, and building Deployment Profiles for Atlanta and Berlin.

6.5.2 eXchange Deployment Project

This section explains how to set up, build, and deploy Atlanta and Berlin Deployment Profiles in the eXchange **Deployment** Project. This Project makes all of the core B2B services and processing available to the application .ear files built from the Deployment Profiles.

To complete this operation, you must set up a Connectivity Map for ASC X12 OTD validation and Deployment Profiles for both Atlanta and Berlin. You must map, build, and deploy both Deployment Profiles.

Creating the Validation Connectivity Map

To complete this operation, you must You must deploy OTD validation BPs via a Connectivity Map, to allow you to configure the Trading Partner Profiles to specify the custom validation handlers. You must then map, build, and deploy two Deployment Profiles, one each for Atlanta and Berlin.

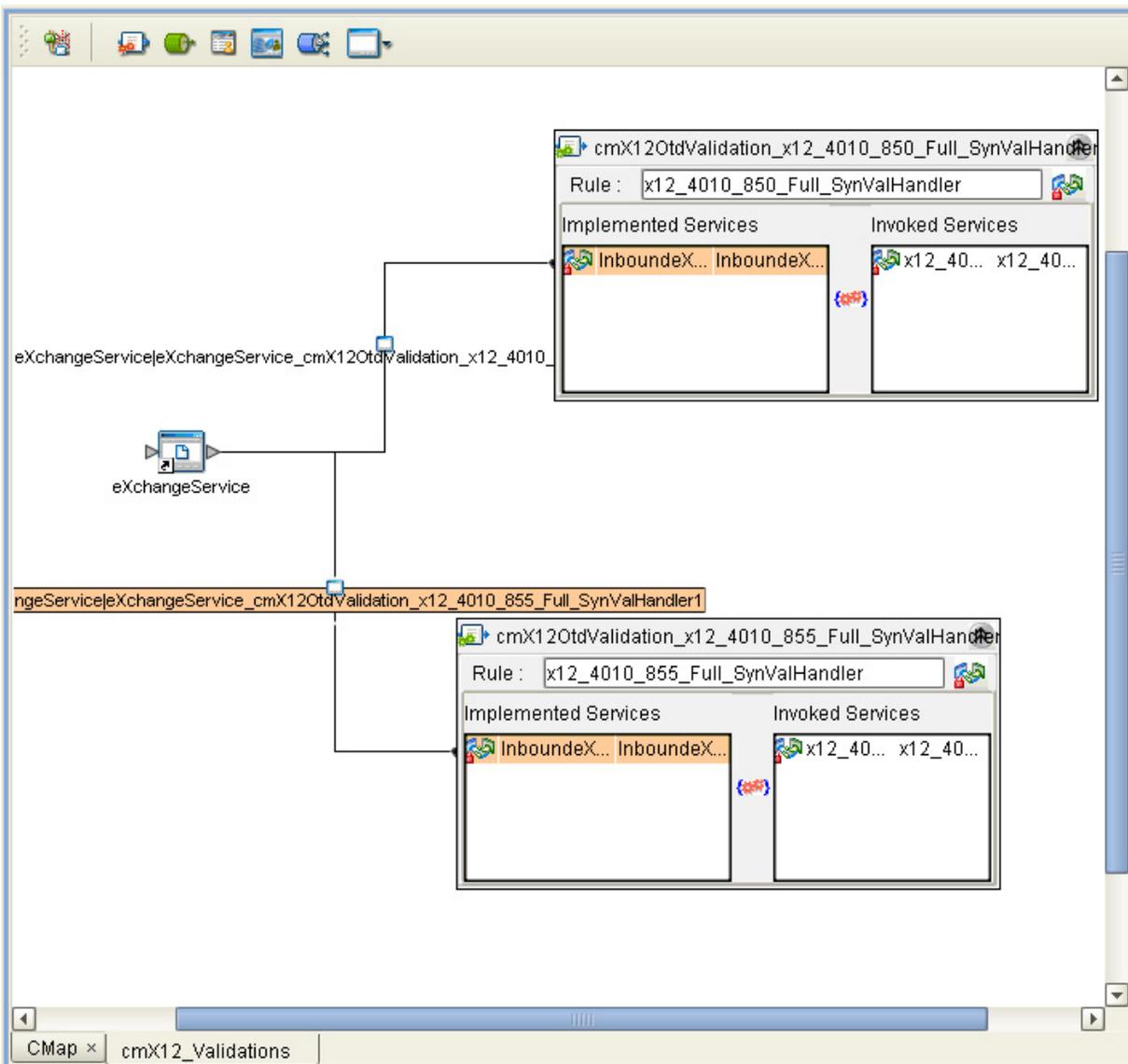
To create the OTD validations Connectivity Map

- 1 Create a Connectivity Map using **Project Explorer**, under **eXchange > Deployment**, named, for example, **cmX12OtdValidation** (or a convenient name for your system, with fewer characters).

- 2 From **Project Explorer**, drag **Sun SeeBeyond > eXchange > User Components > OTD Validations > X12 > v4010 > X12_4010_850_FullSynValhandler** onto the Connectivity Map canvas.
- 3 From the same **Project Explorer** location, drag the **X12_4010_855_FullSynValhandler** component onto the Connectivity Map canvas.
- 4 Drag one **eXchangeService** from **eXchange > Deployment** folder onto the canvas
- 5 Double-click **bpX12_850_FullSynValhandler1** and link **InboundExchangeService** to **eXchangeService**.
- 6 Double-click **bpX12_855_FullSynValhandler1** and link **InboundExchangeService** to **eXchangeService**.

For the results, see Figure 19.

Figure 19 OTD Validation Connectivity Map Linking



- 7 Open and save all the default properties for the eWays on the Connectivity Map.
- 8 Click **Save All** then close the Connectivity Map.

Building and Deploying the Deployment Profiles

When you are finished, you must create Deployment Profiles for both Atlanta and Berlin. These Deployment Profiles are for the eXchange Deployment Project.

To build and deploy the eXchange Deployment Project's Deployment Profile for Atlanta

- 1 On Enterprise Designer's **Project Explorer** tree, right-click **Deployment** and, on the context menu, point at **New** and click **Deployment Profile**.
- 2 In the resulting dialog box, name the new Deployment Profile **dpeXDep_A**, point it at **envA**, make sure it is using all the checked Connectivity Maps, and click **OK**.
- 3 On Deployment Editor, click **Automap** to map the components.
- 4 Click **Build** to build the Deployment Profile for Atlanta.
- 5 Click **Deploy** to deploy the Deployment Profile.

Note: After each of the operations, map, build, and deploy, you receive a status message. If you receive any errors, troubleshoot your previous procedures, as necessary. Deploying Deployment Profiles may take as long or longer than building them.

- 6 When you are finished, click **Save All** and close all canvases.

To build and deploy the eXchange Deployment Project's Deployment Profiles for Berlin

- 1 On Enterprise Designer's **Project Explorer** tree, right-click **Deployment** and, on the context menu, point at **New** and click **Deployment Profile**.
- 2 In the resulting dialog box, name the new Deployment Profile **dpeXDep_B**, point it at **envB**, make sure it is using all the checked Connectivity Maps, and click **OK**.
- 3 On Deployment Editor, click **Automap** to map the components.
- 4 Click **Build** to build the Deployment Profile for Berlin.
- 5 Click **Deploy** to deploy the Deployment Profile.
- 6 When you are finished, click **Save All** and close all canvases.

Result: You have now finished constructing the eXchange Deployment Project, including creating, mapping, building, and deploying Deployment Profiles for Atlanta and Berlin.

6.5.3 Constructing the 855_FromInt_850 Project

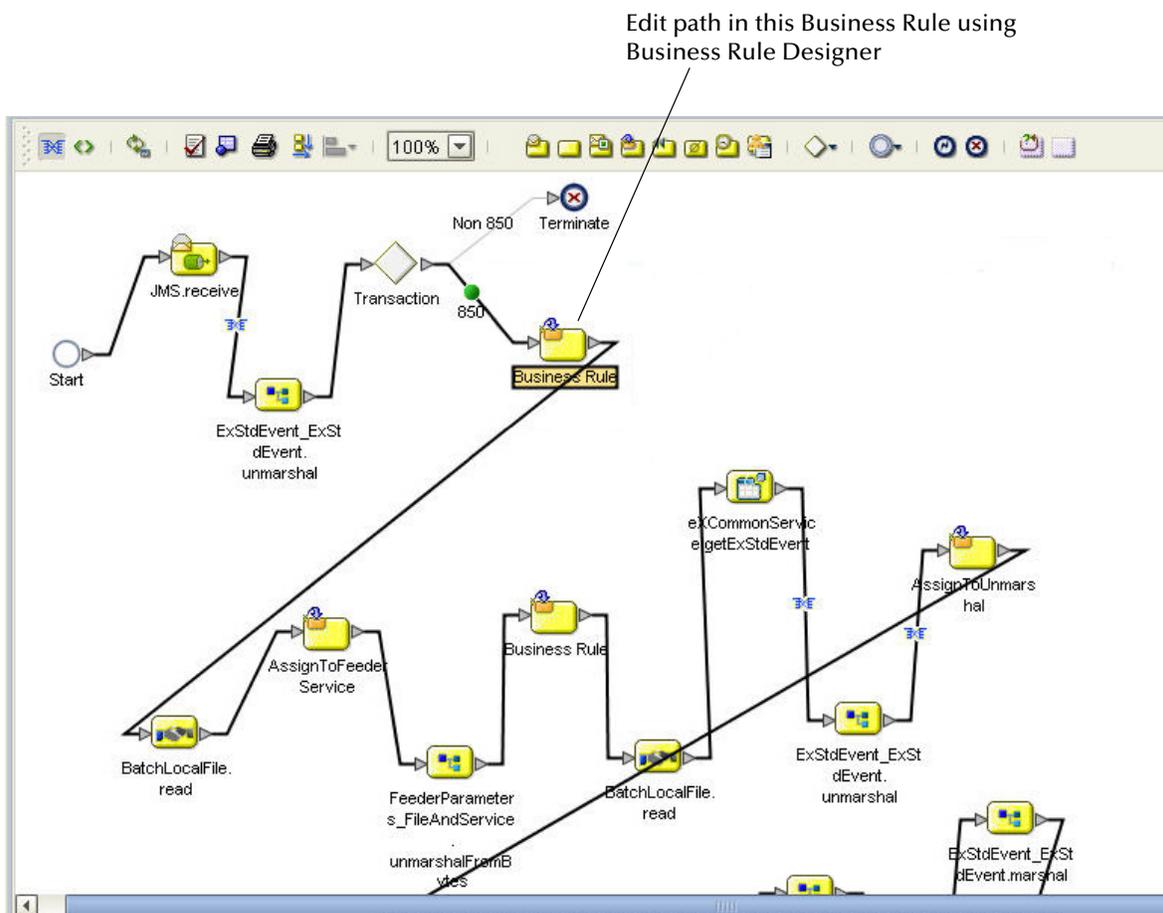
This section describes how to set up, build, and deploy the Berlin Deployment Profile for the **855_FromInt_850** Project. This Project operates with **envB** and makes sure the X12 855 message is returned from Berlin to Atlanta.

Updating the bp855 Business Process

You must make sure the Berlin system is using the data file path listed under [Editing the Sample Data .xml Files](#) on page 71 by checking the bp855 BP in 855_FromInt_850. To do this action, open eXchange > Samples > X12 > 855_FromInt_850 > BPs > bp855 in Project Explorer.

The BP structure appears in Business Process Editor, in Enterprise Designer's right pane. See Figure 20.

Figure 20 Business Process Editor: bp855



Make sure that the path given in the indicated component, in the previous figure, reflects your Berlin data path, for example:

```
C:\temp\exchange\Sample\X12\Data\Berlin
```

This is also the path location of the **X12_dlg_850_In_Atlanta_850_In.xml** and **X12_4010_855_template.st** files.

Note: This setup has been created for the purpose of the sample scenario only. It is recommended that, when creating your own inbound BPs, you configure the BP to read this type of path information from the inbound eWay.

Building and Deploying the Deployment Profile

This section describes how to build and deploy the **855_FromInt_850** Project's Deployment Profile for Berlin.

To build and deploy the **855_FromInt_850** Project's Deployment Profile for Berlin

- 1 On Enterprise Designer's **Project Explorer** tree, right-click **Deployment** and, on the context menu, point at **New** and click **Deployment Profile**.
- 2 In the resulting dialog box, name the new Deployment Profile **dp855_B**, point it at **envB**, make sure it is using all the checked Connectivity Maps, and click **OK**.
- 3 On Deployment Editor, click **Automap** to map the components.
- 4 Click **Build** to build the Deployment Profile for Berlin.
- 5 Click **Deploy** to deploy the Deployment Profile.
- 6 When you are finished, click **Save All** and close all canvases.

Result: You have now finished setting up the **855_FromInt_850** Project, including creating, mapping, building, and deploying Deployment Profile for Berlin only (there is no Atlanta Deployment Profile for this Project).

6.5.4 Constructing the Remaining Projects' Deployment Profiles

Construct the following Projects's Deployment Profiles, as shown under **eXchange** in **Project Explorer**, in the same way you have done for the previous Projects in the sample scenario:

- **ePM**
- **Tracker**
- **Sub_DLQ**
- **Sub_ProcErrors**
- **RecvFromInt**
- **RecvFromTP**; use only the **BatchLocalFile** sub-Project.
- **SendToInt**

To construct the remaining Projects' Deployment Profiles

- Locate, name, and deploy Deployment Profiles for the Projects shown in the previous list, as depicted in **Table 7 on page 88**.

Final Result: You have constructed, built, and deployed (if necessary) all the Projects' Deployment Profiles for the sample scenario.

6.5.5 Summary of Sample Scenario Projects

Table 7 provides a summary list of the sample scenario's Projects, their Deployment Profiles, and corresponding Environments. It is recommended that you construct the Projects and their Deployment Profiles in the order shown in the table.

Table 7 Sample Scenario Projects Summary

Location Under eXchange	Projects	Deployment Profiles	Environments
B2BHosts	X12Host	dpHost_A; not deployed	envA
		dpHost_B; no deployed	envB
Deployment	eXchange Deployment	dpeXDep_A	envA
		dpeXDep_B	envB
Samples > X12	855_FromInt_850	dp855_B	envB
GUI	ePM	dpePM_A	envA
		dpePM_B	envB
	Tracker	dpTrack_A	envA
		dpTrack_B	envB
Error	Sub_DLQ	dpSDLQ_A	envA
		dpSDLQ_B	envB
	Sub_ProcErrors	dpSPErrors_A	envA
		dpSPErrors_B	envB
Samples > X12	RecvFromInt	dpRecvInt_A	envA
	RecvFromTP	dpRecvTP_A	envA
		dpRecvTP_B	envB
	SendToInt	dpSendInt_B	envB

6.6 Importing and Configuring Components in ePM

This section explains how import, create, and configure TPs, Action Groups, Transaction Profiles, and Schedules in the ASC X12 PM sample scenario, using the eXchange ePM. Additionally, the section describes how to use ePM to set the configuration parameter values for the Transaction Profiles and their related components.

Note: For detailed procedures on how to use TPs, Action Groups, Transaction Profiles, and Schedules, see the *eXchange Integrator User's Guide*.

6.6.1 Getting Started

Before you begin, it is recommended that you do all the procedures given in the previous sections of this chapter. Also, your Integration Server, Repository, and Logical Host must be running. Also, your LDAP system and eXchange database (Oracle) must be running and accessible. However, Enterprise Designer does not need to be running.

Note: For a general description of the outbound and inbound messaging **ToPartner** and **FromPartner** model used by ePM, see [Configuring ePM: ToPartner and FromPartner Messaging Model](#) on page 44.

You *must* do the procedures given under [Running ePM](#) on page 89 to run ePM. In addition, after you run ePM, it is recommended that you do the procedures given under [Importing B2B Hosts](#) on page 90 and [Importing TPs](#) on page 92.

Note: If you want, you can create your own B2B Hosts and TPs using the procedures given in this section and using the sample scenario B2B Hosts and TPs as models. For more details, see the *eXchange Integrator User's Guide*.

The remainder of this section explains these operations.

Note: For detailed information on configuring ePM, see [Chapter 4](#).

Exporting the Necessary ePM Files

The ePM export files for the B2B Hosts, TPs, and Schedules are located in **Project Explorer** under **eXchange**, as follows:

- **ePMImport > X12 > Hosts > envA_X12.exp** and **envB_X12.exp**
- **ePMImport > X12 > Schedulers > envA_X12_S1.exp** and **envB_X12_S1.exp**
- **ePMImport > X12 > TP_Profiles > envA_X12_TP_Berlin.exp** and **envB_X12_TP_Atlanta.exp**

For more information, see [Exporting ePM Files](#) on page 59. It is recommended that you set up a folder structure to contain these files, which reflects this organization, for example:

```
C:\temp\exchange\Sample\X12\TP_Profiles
```

Note: For information on how to use ePM, see the *eXchange Integrator User's Guide*.

6.6.2 Running ePM

This section explains how to start running ePM.

To run eXchange ePM

- 1 Start a browser session.
- 2 Enter the Logical Host name and ePM port number with `epm` appended, as follows:

```
http://<logicalhost>:<port+1>/epm
```

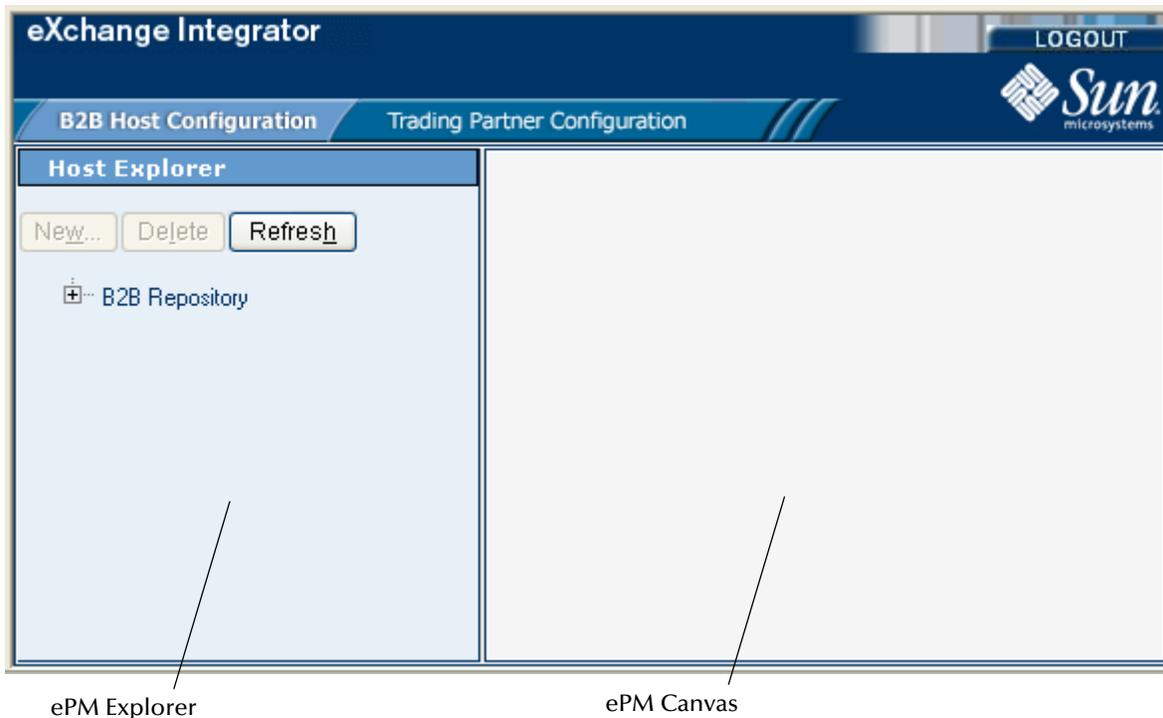
For example:

`http://localhost:18001/epm`

- 3 When the sign-in window appears, enter your Enterprise Manager user name (or the new user described under [Adding a New User to ePM and Message Tracking](#) on page 67), as well as the appropriate password, and click **Sign In**.

The initial ePM window appears. See Figure 21.

Figure 21 Initial ePM Window



The ePM window has the following sections:

- **ePM Explorer**
 - ♦ **B2B Host Configuration** tab
 - ♦ **Trading Partner Configuration** tab
- **ePM Canvas**

Note: For complete instructions on how to use ePM, see the *eXchange Integrator User's Guide*.

6.6.3 Importing B2B Hosts

Your next step is importing the Atlanta and Berlin B2B Host files, as explained under this section.

This sample scenario has the following B2B Hosts:

- **envA_X12**: For Atlanta.
- **envB_X12**: For Berlin.

To import the envA B2B Host

- 1 Log in to ePM for **envA (dmnA)**.
- 2 Click the **B2B Host Configuration** tab, if **Host Explorer** is not already displayed.
- 3 In **Host Explorer**, click and expand **B2B Repository**.
- 4 Select the B2B Host **envA_X12**.
- 5 At the bottom of **ePM Canvas**, click **Import**.
- 6 When you are finished, click **Save**.

To import the envB B2B Host

- 1 Log in to ePM for **envB (dmnB)**.
- 2 Click the **B2B Host Configuration** tab, if **Host Explorer** is not already displayed.
- 3 In **Host Explorer**, click and expand **B2B Repository**.
- 4 Select the B2B Host **envB_X12**.
- 5 At the bottom of **ePM Canvas**, click **Import**.
- 6 When you are finished, click **Save**.

6.6.4 Using Schedules

Next, you must import the Schedules for the B2B Hosts.

To import a Schedule

- 1 Log in to ePM for **envA (dmnA)** for Atlanta; use **envB (dmnB)** for Berlin.
- 2 Click the **B2B Host Configuration** tab.
- 3 **ePM Explorer** with this tab selected appears.
- 4 Click the **Schedule** icon in **ePM Explorer**.
- 5 Select the Schedule file you want to import (**envA_X12_S1.exp** for **envA** and **envB_X12_S1.exp** for **envB**).
- 6 Click **Import**.
- 7 When you are finished, click **Save**.

To modify an existing Schedule

- 1 Log in to ePM for **envA (dmnA)** for Atlanta; use **envB (dmnB)** for Berlin.
- 2 Click the **B2B Host Configuration** tab.
- 3 **ePM Canvas** with this tab selected appears.
- 4 Click the **Settings** tab.

- 5 In ePM Canvas, modify the scheduling information for the current Schedule, as necessary. This information is for inbound only.
- 6 When you are finished, click **Save**.

6.6.5 Importing TPs

Your next step is importing or creating the Atlanta and Berlin TP files, as explained under this section. Keeping track of the TPs, where they are sent from, and where they are received depends on which company you consider to be your current company. See [Configuring ePM: ToPartner and FromPartner Messaging Model](#) on page 44.

This sample scenario has the following TPs:

- **Berlin:** For Atlanta.
- **Atlanta:** For Berlin.

Note: Also, you may create and construct these TPs yourself, using the sample TPs as models.

This sample scenario has the following TP files (under TP_Profiles):

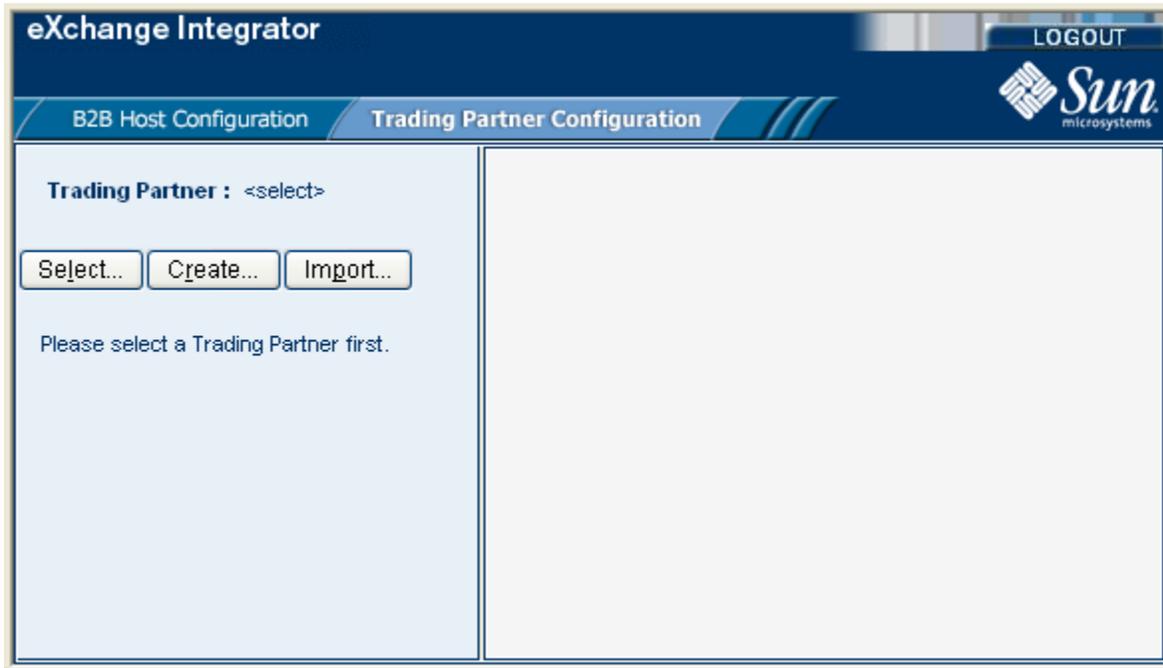
- **envA_X12_TP_Berlin.exp:** For Atlanta.
- **envA_X12_TP_Atlanta.exp:** For Berlin.

To import the Berlin TP to envA

- 1 Click the **Trading Partner Configuration** tab.

The ePM window with this tab selected appears. See Figure 22.

Figure 22 ePM Window With Trading Partner Configuration Tab Selected



- 2 From this window, click **Import**.

The **Import a Trading Partner - Step 1 of 2** window appears in **ePM Canvas**.

- 3 Name the TP **Berlin**.

- 4 Browse to the folder where you have stored your TP files and select **envA_X12_TP_Berlin.exp**, then click **Next**.

The **Import a Trading Partner - Step 2 of 2** window appears.

Note: See [Exporting Sample Files](#) on page 58 for a list of the sample scenario's data files.

- 5 Choose **envA_X12**, from the pull-down menu.
- 6 Click **Finish**.

To locate the Berlin TP in the ePM window

- 1 In the upper left side of the ePM window, click **Select**.

The **Select the Trading Partner to Configure** window appears in **ePM Canvas**.

- 2 Click **Search** on the canvas.

Any available TPs appear directly below.

- 3 In this case, you are looking for **Berlin**, which appears.
- 4 Click the TP name, in this case **Berlin**, to configure the TP.

To import the Atlanta TP to envB

- 1 Click the **Trading Partner Configuration** tab.
The ePM window with this tab selected appears.
- 2 From this window, click **Import**.
The **Import a Trading Partner - Step 1 of 2** window appears in ePM Canvas.
- 3 Name the TP **Atlanta**.
- 4 Browse to the folder where you have stored your TP files and select **envB_X12_TP_Atlanta.exp**, then click **Next**.
The **Import a Trading Partner - Step 2 of 2** window appears.
- 5 Choose **envB_X12**, from the pull-down menu.
- 6 Click **Finish**.

To locate the Atlanta TP in the ePM window

- Use the same procedure as explained previously.

6.6.6 Using Action Groups and Transaction Profiles

You do the actual configuration of Action Groups and Transaction Profiles using parameters available using the following levels of the **ePM Explorer** tree:

- **Business Protocols**
- **Delivery Protocols**
- **Transports**

More information on how to configure **Transport** parameters is available in the *eXchange Integrator User's Guide*. For information on how to configure these parameters in the sample scenario, see [Configuring Transports](#) on page 98.

6.6.7 Configuring the Sample Scenario

You may use the ePM for the Projects in the sample scenario as a model to complete the configuration of ePM. Enter information in ePM as shown in the sample. For more information, see [Chapter 4](#) and the *eXchange Integrator User's Guide*.

Using Parameters in ePM

Many of the Business and Delivery Protocol parameters are the same, regardless of the PM and business communication protocol you are using. However, some of them are ASC X12 PM-specific and are only present for ASC X12.

This section describes these parameters under:

- [Interchange Envelope Parameters](#) on page 96
- [Functional Group Parameters](#) on page 97

For more information on these parameters, see [Configuring ASC X12 PM ePM Parameters](#) on page 44.

Note: For more information on parameters not described under this section, see the *eXchange Integrator User's Guide*.

To enter an override value for a parameter

- 1 Click the check box next to the parameter you want to override, under the **Override** column (see [Figure 23 on page 96](#) for an example).
- 2 Enter the appropriate override in the text box for the parameter.
- 3 Click **Save**.

ePM Configuration General Operation

In the sample scenario, you import preconfigured B2B Hosts and TPs. When you are configuring your own (for example, if you are creating B2B Hosts and TPs from scratch), it is recommended that you use the following general order of configuration operations when setting values in ePM:

- On the **B2B Host Configuration** tab, create a new B2B Host Transaction Profile.
- Create a new Action Group (ePM here calls this a Business Action Group) for the B2B Host. Be sure to choose the correct Delivery Action and External Transport for the Action Group's Business Actions.
- On the **Trading Partner Configuration** tab, create a new TP.
- For the new TP, select the Transaction Profile from the B2B Host.
- Choose the necessary settings (**Settings** tab) for the current TP Transaction Profile.
- Open any applicable Business Actions and set appropriate overrides, as necessary.

Interchange Envelope Parameters

The ASC X12 PM-specific Interchange Envelope (outer envelope or ISA) parameters appear in **ePM Canvas** as shown in Figure 23. The example shown is for the **Atlanta TP's 850 FromPartner** Transaction Profile.

Figure 23 ASC X12 PM-specific Interchange Envelope Parameters: Example

Property	Current Value	Override?	Inherits From	Original Value
ISA01 AUTHOR INFO QUAL: *	00	<input type="checkbox"/>	Host Business Protocol	00
ISA02 AUTHOR INFORMATION:		<input type="checkbox"/>	Host Business Protocol	
ISA03 SEC INFO QUAL: *	00	<input type="checkbox"/>	Host Business Protocol	00
ISA04 SECURITY INFORMATION:		<input type="checkbox"/>	Host Business Protocol	
ISA05 IC SENDER ID QUAL: *	01	<input type="checkbox"/>	Host Business Protocol	01
ISA06 INTERCHANGE SENDER ID: *	AtlantaIID	<input type="checkbox"/>	Host Business Protocol	AtlantaIID
ISA07 IC RCVR ID QUAL: *	01	<input type="checkbox"/>	Host Business Protocol	01
ISA08 INTERCHANGE RCVR ID: *	Set Value	<input type="checkbox"/>	Host Business Protocol	Set Value
ISA11 IC CONTROL STANDARD IDENTIFIER: *	U	<input type="checkbox"/>	Host Business Protocol	U
ISA12 IC VERSION NUMBER: *	00401	<input type="checkbox"/>	Host Business Action Group (4010_270-271_ActionGroup)	00401
ISA13 IC CONTROL NUMBER:	1	<input type="checkbox"/>	Host Business Protocol	1
ISA14 ACKNOWLEDGMENT REQUESTED: *	1	<input type="checkbox"/>	Host Business Protocol	1
ISA15 USAGE INDICATOR: *	P	<input type="checkbox"/>	Host Business Protocol	P
ISA16 COMP ELE SEP:	:	<input type="checkbox"/>	Host Business Protocol	:
SEGMENT TERMINATOR:	~	<input type="checkbox"/>	Host Business Protocol	~
ELEMENT SEPARATOR:	*	<input type="checkbox"/>	Host Business Protocol	*

Save Last modified at 9/19/06 2:56 PM by aa

You may display the example in Figure 23 by using **ePM Explorer**, under **B2B Repository > envA_X12 > Business Protocols > X12**.

As you can see, there are no override values entered in Figure 23, and the configuration accepts all the defaults. You may override the defaults for any parameter by clicking a check box under **Override** and entering a different value, as necessary.

You may, of course, do the same override operation with other ePM parameters, as described under **Configuring eXchange Partner Manager: Overview** on page 39.

Functional Group Parameters

The ASC X12 PM-specific Functional Group (inner envelope or GS) parameters appear in **ePM Canvas** as shown in Figure 24. The example shown is for the **Atlanta TP's 850 FromPartner** Transaction Profile.

Figure 24 ASC X12 PM-specific Functional Group Parameters: Example

Property	Current Value	Override?	Inherits From	Original Value
GS01 FUNCTIONAL ID CODE: *	H5	<input checked="" type="checkbox"/>	Host Business Protocol	Set Value
GS02 APPLICATION SENDER CODE: *	Set Value	<input type="checkbox"/>	Host Business Protocol	Set Value
GS03 APPLICATION RCVR CODE: *	AtlantaACd	<input type="checkbox"/>	Host Business Protocol	AtlantaACd
GS04 DATE FORMAT: *	CCYYMMDD	<input type="checkbox"/>	Host Business Protocol	CCYYMMDD
GS05 TIME FORMAT: *	HHMM	<input type="checkbox"/>	Host Business Protocol	HHMM
GS07 RESP AGENCY CODE: *	X	<input type="checkbox"/>	Host Business Protocol	X
GS08 VERS/REL/INDUST ID CODE: *	004010	<input type="checkbox"/>	Host Business Action Group (4010_270-271_ActionGroup)	004010
Business - Duplication Checking: *	No	<input type="checkbox"/>	Host Business Protocol	No
Business - Use Decryption: *	No	<input type="checkbox"/>	Host Business Protocol	No
Business - Verify Signature: *	No	<input type="checkbox"/>	Host Business Protocol	No
Business - Use Decompression: *	No	<input type="checkbox"/>	Host Business Protocol	No
Business - Send Acknowledgements: *	Yes	<input type="checkbox"/>	Host Business Protocol	Yes
Business - Character Set Encoding:		<input type="checkbox"/>	Host Business Protocol	
Business - Message Encoding:		<input type="checkbox"/>	Host Business Protocol	
Business - Content Type:		<input type="checkbox"/>	Host Business Protocol	

Batch - Inbound FromPartner

Save Last modified at 9/19/06 2:55 PM by aa

You may display the example in Figure 23 by using **ePM Explorer**, under **B2B Repository > envA_X12 > Business Protocols > X12**.

As you can see, one override value has been entered in Figure 24, and the configuration uses the defaults for all other displayed parameters. As explained previously, you may override the default values, as necessary.

Configuring Transports

Table 8 lists the **Transports** parameter override values you must enter to configure **B2B Host Configuration > envA_X12 > Transports > BatchLocalFile**.

Table 8 Overrides for Atlanta B2B Host Configuration > ... > BatchLocalFile

Category	Parameter	Override Value
Outbound ToPartner	TargetDirectoryName	C:\temp\Exchange\Sample\X12\Data\Atlanta
Inbound FromPartner	TargetFileName	ToAtlanta_[a-zA-Z0-9]*\.msg
ACK - Outbound ToPartner	TargetDirectoryName	C:\temp\Exchange\Sample\X12\Data\Atlanta
ACK - Inbound FromPartner	TargetFileName	ToAtlanta_[a-zA-Z0-9]*\.msg

Table 9 lists the **Transport** parameter override values you must enter to configure **B2b Host Configuration > envB_X12 > Transports > BatchLocalFile**.

Table 9 Overrides for Berlin B2B Host Configuration > ... > BatchLocalFile

Category	Parameter	Override Value
Outbound ToPartner	TargetDirectoryName	C:\temp\Exchange\Sample\X12\Data\Berlin
Inbound FromPartner	TargetFileName	ToBerlin_[a-zA-Z0-9]*\.msg
ACK - Outbound ToPartner	TargetDirectoryName	C:\temp\Exchange\Sample\X12\Data\Berlin
ACK - Inbound FromPartner	TargetFileName	ToBerlin_[a-zA-Z0-9]*\.msg

Running the Sample

See [Running the Sample Scenario](#) on page 62 for information on how to complete the passing of data between the two TPs.

6.7 Using Message Tracking

eXchange provides a special feature, Message Tracking, allowing you to monitor the status of messages as they are received and processed through eXchange and ASC X12 PM.

6.7.1 Before You Begin

- You must already have deployed the appropriate Projects' Deployment Profiles for in the sample scenario.

Note: *The B2B Host Project is not deployed.*

- Your Oracle and LDAP systems for eXchange must already be running, and you must already have begun running a Logical Host before you can run Message Tracking.
- The inbound and outbound scenarios for the sample must be running.
- For Message Tracking to be useful, there must be one or more messages that have already been picked up by the current Logical Host's Integration Server.

6.7.2 Accessing Message Tracking

This section explains how to access Message Tracking.

To access Message Tracking

- 1 Start a browser session.
- 2 Point your browser at the following URL:

```
http://<logicalhost>:<port+1>/<objname>/tracker
```

Where:

- ♦ *logicalhost*: The host name or IP address of a Logical Host running your Project, that is, the current Logical Host.
- ♦ *port*: is The Web server connector port configured in your Integration Server. To discover this information, use **Environment Explorer** to open the current Logical Host. Right-click the Integration Server and select **Properties**. Open **IS Configuration > Sections > Web Container > Web Server > Default Web Server**; *port* is the value set for **Connector Port**. If you have several Web server configurations, check them also.

The default port-number value is 18001 for the first Integration Server in the first-created Logical Host. (or 28001 for the first Integration Server in the second-created Logical Host, and so on). For the sample scenario, use 18001 for Atlanta (**envA**) and 28001 for Berlin (**envB**).

- ♦ *objname*: The name of the Message Tracking instance as it appears on the current Connectivity Map. For the sample scenario, this name is `tracker`.

Example: To access Message Tracking for the sample scenario, use the following URLs:

- <http://localhost:18001/tracker>

Note: You can only use the same port number for different Message Tracking instances if they reside on different machines.

As stated previously, the sample must be running before you access Message Tracking, and messages must have been transported before any become accessible.

Message Tracking Window

When you first run Message Tracking, the **Message Tracking** window appears. After you perform a search, as necessary, in the window's left pane, message information results appear in the right pane. See Figure 25.

Figure 25 Atlanta Host, Example Message Tracking Window

The screenshot shows the eXchange Integrator Message Tracking interface. The left pane is titled 'Search Criteria' and includes a 'Host' dropdown set to 'envA_X12', a 'Trading Partner' dropdown set to 'ALL', a 'Protocols' list with 'ALL' and 'X12', and 'Filters' for 'Error Type' (ALL), 'Direction' (Both), and 'Date Filter' (Process date). The right pane is titled 'Search Results' and displays a table with the following data:

Transaction ID	Trading Partner	Protocol	Dialog	Dialog ID	Action	Direction	Error Data	Response Req
f13010000-D6C83455100100-8199A718-011X128503247	Berlin	X12	PurchaseOrder_AG	{13010000-D6C83455100100-8199A718-011X128503247	855	Inbound	No	Yes
f13010000-D6C83455100100-8199A718-011X128503247	Berlin	X12	PurchaseOrder_AG	{13010000-D6C83455100100-8199A718-011X128503247	850	Outbound	No	Yes

See the *eXchange Integrator User's Guide* for information on how to use Message Tracking.

Glossary of Acronyms

AD, *AD, xAD

In eXchange an *Attributes Definition* defines the metadata attributes of parameters used in a business protocol, delivery protocol, or transport. Examples of *xADs* include: BPAD=BAD+EAD; DPAD=MAD+PAD; and TAD.

AS2

Applicability Statement 2 (AS2) is an Internet Draft security standard defined by the IETF (Internet Engineering Task Force), designed to allow business transactions to move securely over the Internet.

B2B

Business-to-business (B2B) interactions are those that occur between business partners in the context of e-commerce.

BAD

In eXchange, *Business Attribute Definitions (BADs)* define the metadata attributes of message payload parameters used in business protocols such as X12, HIPAA, EDIFACT, or CIDX. Each BAD combines with one EAD to constitute a BPAD.

BPAD

In eXchange, *Business Protocol Attribute Definitions (BPADs)* define metadata for business protocols such as X12, HIPAA, EDIFACT, or CIDX. A BPAD consists of one Business Attributes Definition (BAD) and one Enveloping Attributes Definition (EAD).

CAPS

The Sun Java *Composite Application Platform Suite (Java CAPS)* includes eGate Integrator, eInsight Business Process Manager eXchange Integrator, eWay Intelligent Adapters, OTD Libraries, and Protocol Managers, as well as many other products.

CIDX

The *Chemical Industry Data Exchange (CIDX)* is a non-profit organization dedicated to improving the ease, speed and cost of securely conducting business electronically in the chemical industry. CIDX focuses on the development of eBusiness standards, called Chem eStandards.

DPAD

In eXchange, *Delivery Protocol Attribute Definitions* (DPADs) define metadata for delivery protocols such as AS2, ebXML, or RNIF. A DPAD consists of one Messaging Attributes Definition (MAD) and one Packaging Attributes Definition (PAD).

EAD

In eXchange, *Enveloping Attribute Definitions* (EADs) define the metadata attributes of message envelope parameters used in business protocols such as X12, HIPAA, EDIFACT, or CIDX. Each EAD combines with one BAD to constitute a BPAD.

ebXML

A well-recognized *e-business XML* (extensible markup language; see “XML”) whose implementation includes specifications for messaging, collaboration profiles, business processes, and metadata registry.

ePM

eXchange Partner Manager (ePM) is a Web-based GUI for defining and managing Trading Partner (TP) information.

FTP

File Transport Protocol (FTP) is a transport protocol for sending and receiving files. Specifications for FTP include RFCs 959, 1635, 2228, and 2577.

HTTP

Hypertext Transport Protocol (HTTP) is a transport protocol for transmitting information referenced in a URL of the form **http://<hostname>:<port>/.../...** Specifications for HTTP include RFCs 2068, 2616, 2617, 2660, and 3310.

ICAN

Before Java CAPS 5.1.x, SeeBeyond offered an *Integrated Composite Application Network* (ICAN) Suite that included eGate Integrator, eXchange Integrator, various eWay Intelligent Adapters, OTD Libraries, and Protocol Manager Composite Applications, as well as many other products.

LDAP

The *Lightweight Directory Access Protocol* is a standard networking protocol for querying and modifying information stored as a distributed nonrelational database in directory servers (informally called “LDAP servers”) accessed via TCP/IP. Specifications for LDAP include RFCs 1777-1779 and 2251-2255.

MAD

In eXchange, *Messaging Attribute Definitions* (MADs) define the metadata attributes of messaging parameters used in delivery protocols such as AS2, ebXML, or RNIF. Each MAD combines with one PAD to constitute a DPAD.

MIME

Multipurpose Internet Mail Extensions (MIME) extends the format of basic Internet mail to allow non-textual messages, multipart message bodies, and so forth. Specifications for MIME include RFCs 2045–2049.

OTD

In Java CAPS, an *Object Type Definition* (OTD) contains the data structure and rules that define an object. OTDs are used in Java collaborations to transform data interface with external systems.

PAD

In eXchange, *Packaging Attribute Definitions* (PADs) define the metadata attributes of packaging parameters used in delivery protocols such as AS2, ebXML, or RNIF. Each PAD combines with one MAD to constitute a DPAD.

RNIF

The purpose of the *RosettaNet Implementation Framework* (RNIF) is to allow trading partners to configure their business processes in such a way as to operate with other trading partners adhering to the same framework, allowing electronic business transactions to be conducted securely over the Internet.

SME

In Java CAPS, *Secure Messaging Exchange* (SME) uses advanced cryptographic techniques to ensure security, verifiability, and nonrepudiation of messages exchanged electronically.

S/MIME

Secure/Multipurpose Internet Mail Extensions (S/MIME) provides a consistent way to send and receive secure MIME data, using digital signatures for authentication, message integrity and non-repudiation and encryption for privacy and data security. Specifications for S/MIME version 2 include RFCs 2311–2315.

SMTP

Simple Mail Transfer Protocol (SMTP) is a transport protocol for transmitting e-mail messages between servers or from client to server. Specifications for SMTP include RFCs 1651, 2821, and 3461.

TAD

In eXchange, *Transport Attribute Definitions* (TADs) define the metadata attributes of parameters used in transport protocols such as FTP or HTTP.

TCP/IP

The *Transmission Control Protocol/Internet Protocol* is a standard suite of communication protocols for connecting hosts and transmitting data over the Internet.

TP, TPP

In eXchange, a *Trading Partner* (TP) has one or more *Trading Partner Profiles* (TPPs) that contain information identifying the values of messaging, enveloping, and/or transport parameters to be used for sending and receiving B2B information.

URL

A *Uniform Resource Locator* (URL) is a string that identifies information, such as a particular piece of information shared by a particular host.

XML

An *Extensible Markup Language* (XML) is a language whose syntax obeys an official schema, called “the XML schema”, but whose semantics (“vocabulary”) are open.

Index

A

- Accredited Standards Committee 10
- acknowledgement types 26
- acknowledgments
 - as part of EDI logic 29
 - Functional Acknowledgment (997) 26
 - Interchange Acknowledgment (TA1) 26
- adding application server instances 69
- adding new user to ePM and Message Tracking 67
- American National Standards Institute 14
- ANSI 14
- application acknowledgements 27
- application server instances, adding 69
- ASC 10
- ASC X12 parameters, configuring 44
- ASC X12 PM overview
 - ASC X12 OTD Library 28
 - basic operation 27
 - EDI processing 29
 - introduction 27
 - message information levels 30
 - messaging 28
 - OTD structures 29
 - PM and eXchange 28
 - translations, enveloping, and acknowledgments 29
 - version support 30
- ASC X12 Protocol Manager 10
- ASC X12 protocol, about
 - components 20
 - control numbers 22
 - data elements 20
 - envelope structure 16
 - Functional Groups 17
 - Interchange Envelopes 18
 - introduction 14
 - loops 21
 - message body 14
 - message structure 15
 - schematic structure 16
 - segments 21
 - Transaction Sets 19
 - what is it? 14

B

- Business Process (BP) 31

C

- compatibility of versions, backward 23
- configurations, server 66
- configuring
 - HTTPS 66
 - LDAP 66
 - Oracle eWay 66
- configuring external systems, brief overview 60
- configuring TPs in ePM
 - Environments 44
 - parameter types 44
- constructing Environments, brief overview 59, 60
- conventions, text 12
- creating and starting the Domains 66

D

- data element separator 22
- data elements 20
- Data Interchange Standards Association 14
- delimiters 15, 21, 22
- Deployment Profiles, deploying
 - brief overview 61
 - special considerations 61
- DISA 14
- Domains, creating and starting 66

E

- EDI 14, 23
 - payment processing overview 23
 - usage example 23
- editing sample data files 59, 71
- electronic payment types 24
- Element Separator 50
- Enterprise Designer
 - initializing 70
 - Oracle configuration 76
 - running 70
- enveloping
 - as part of EDI logic 29
- Environment Explorer, using 74
- Environment, see "setting up sample Environments"
- Environments, constructing
 - brief overview 59, 60
- ePM files, importing, brief overview 62
- ePM parameters, see "parameters, ePM"
- ePM, adding new user 67
- ePM, brief overview

- B2B Hosts and TPs 43
 - components 40
 - defaults and overrides 41
 - features 39
 - hierarchy of inheritance and overrides 42
 - parameters and properties 40
- ePM, using
 - Action Groups and Transaction Profiles 94
 - configuration, general operation 95
 - configuring transports 98
 - exporting ePM files 89
 - Functional Group parameters 97
 - getting started 89
 - importing B2B Hosts 90
 - importing TPs 92
 - Interchange Envelope parameters 96
 - running ePM 89
 - Schedules 91
 - using configuration parameters 94
- example of EDI usage 23
- eXchange Integrator, general information 31
- exporting sample data files 71
- exporting sample files 58
- external systems, configuring
 - brief overview 60

F

- Functional Acknowledgment 17, 19
- Functional Acknowledgments (997) 26
- Functional Group (GS/GE) 17
- functional group parameters, ePM 50
- funds transfers 25

G

- GS01 Functional ID Code 50
- GS02 Application Sender Code 51
- GS03 Application Rcvr Code 51
- GS04 Date Format 51
- GS05 Time Format 51
- GS06 Group Control Num 52
- GS07 Resp Agency Code 52
- GS08 Vers/Rel/Indust ID Code 52

H

- HTTPS (HTTP on SSL), configuring 66

I

- importing ePM files
 - brief overview 62

- initializing Enterprise Designer 70
- installation 35
 - after 38
 - basic procedures 36
 - before 35
 - installing prerequisite files 35
 - Repository and .sar files 36
- intended audience 11
- Interchange Acknowledgment (TA1) 26
- Interchange Envelope (ISA/IEA) 18
- interchange envelope parameters, ePM 45
- ISA01 Author Info Qual 45
- ISA02 Author Information 45
- ISA03 Sec Info Qual 46
- ISA04 Security Information 46
- ISA05 IC Sender ID Qual 46
- ISA06 Interchange Sender ID 47
- ISA07 IC Rcvr ID Qual 47
- ISA08 Interchange Rcvr ID 47
- ISA11 IC Control Standard Identifier 47, 48
- ISA13 IC Control Number 48
- ISA14 Acknowledgment Requested 48
- ISA15 Usage Indicator 49
- ISA16 Comp Elem Sep 49

J

- Java CAPS Readme file 35

L

- LDAP 77
- LDAP, configuring 66
- loops 21

M

- Message Tracking, adding new user 67
- Message Tracking, using
 - accessing Message Tracking 99
 - before beginning 99
 - Message Tracking window 100
- messages, configuring eGate for large 35

O

- operating systems, supported 34
- Oracle configuration 76
- Oracle eWay, configuring 66
- organization of information, guide 11
- overview
 - of EDI payments processing 23

P

parameters, ePM

- Element Separator 50
- GS01 Functional ID Code 50
- GS02 Application Sender Code 51
- GS03 Application Rcvr Code 51
- GS04 Date Format 51
- GS05 Time Format 51
- GS06 Group Control Num 52
- GS07 Resp Agency Code 52
- GS08 Vers/Rel/Indust ID Code 52
- ISA01 Author Info Qual 45
- ISA02 Author Information 45
- ISA03 Sec Info Qual 46
- ISA04 Security Information 46
- ISA05 IC Sender ID Qual 46
- ISA06 Interchange Sender ID 47
- ISA07 IC Rcvr ID Qual 47
- ISA08 Interchange Rcvr ID 47
- ISA11 IC Control Standard Identifier 47, 48
- ISA13 IC Control Number 48
- ISA14 Acknowledgment Requested 48
- ISA15 Usage Indicator 49
- ISA16 Comp Elem Sep 49
- Segment Terminator 49

payment-related EDI transactions 26

physical hierarchy, message 15

Project, see "setting up sample Projects"

properties, message 15

Q

quick start or tutorial? 54

quick start procedures, using 54

R

Readme file 35, 76

repetition separator 22

response transactions 27

running Enterprise Designer 70

running the sample scenario 62

S

sample data files

editing 71

exporting 71

sample data files, editing 59

sample files, exporting 58

sample scenario implementations, introduction 65

sample scenario overview

before you start 57

business description 56

getting started 57

introduction 55

Projects 56

sample scenario, operational diagram 65

scope of guide 11

screenshots 12

SEF file 29

SEF OTD Wizard 29

SEF wizard, using 30

Segment Terminator 49

segment terminator 22

segments 21

server configurations 66

setting up Environments

creating basic components 74

setting up sample Environments

additional external systems, creating and configuring 79

B2B Configurator Service, creating and configuring 77

creating basic components 74

File eWays, creating and configuring 78

final result 80

LDAP system, creating and configuring 77

Oracle system, creating and configuring 76

when finished 80

setting up sample Projects

B2B Host 81

constructing 271_FromInt_270 Project 85

creating validation Connectivity Map 83

deploying additional Deployment Profiles 87

deploying Deployment Profiles 85

deploying Deployment Project Deployment Profile 87

eXchange Deployment Project 83

summary list 88

updating bp271 BP 86

setup steps, basic overview 55

solving business problems, references 65

SSL (Secure Sockets Layer), configuring 66

structures

as part of EDI logic 29

subelement (component) separator 22

Sun Java Composite Application Platform Suite (Java CAPS) 10

syntax 21

system prerequisites 34

T

TA1 (Interchange Acknowledgment) 26

text conventions 12

TP

Index

- Environments 44
- TP agreements 30
- TP configuration
 - overview 39
- Transaction Set (ST/SE) 19
- translations
 - as part of EDI logic 29
- transports, configuring 98
- tutorial, using 64

U

- using Environment Explorer 74

V

- validation Connectivity Map, creating 83
- validations
 - as part of EDI logic 29

X

- X12, see "ASC X12"