

SeeBeyond ICAN Suite

AS2 Manager Composite Application User's Guide

Release 5.0.5



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

SeeBeyond, e*Gate, e*Way, and e*Xchange are the registered trademarks of SeeBeyond Technology Corporation in the United States and/or select foreign countries. The SeeBeyond logo, SeeBeyond Integrated Composite Application Network Suite, eGate, eWay, eInsight, eVision, eXchange, eView, eIndex, eTL, ePortal, eBAM, and e*Insight are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2004 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

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

Version 20041119085254.

Contents

List of Figures	6
-----------------	---

List of Tables	7
----------------	---

Chapter 1

Introduction	8
About This Document	8
Organization of Information	8
Scope, Audience, and Purpose	9
Document Conventions	9
Screenshots	9
Related Documents	10
SeeBeyond Web Site	10
SeeBeyond Documentation Feedback	10

Chapter 2

Introducing AS2 Manager	11
About AS2	11
General History and Definitions	11
Definition of EDI-INT	12
Definition of AS1	12
Definition of AS2	12
AS2 Message Format	12
Packaging Layer References	13
Additional References	14
About eXchange Integrator	14
eXchange and the ICAN Suite	15
eXchange Architectural Overview	15
Process Overview	16

Chapter 3

Installing AS2 Manager	18
System Requirements	18
Supported Operating Systems	18
Database Support	19
Database for eXchange Partner Management and Message Tracking	19
Database for Persistence and Monitoring via eInsight Engine	19
Before You Install	19
Installing the Product Files	19
Uploading AS2 Manager to the Repository	20
Refreshing Enterprise Designer	22
Database Scripts	24
Additional Policy JAR Files Required to Run SME	24

Chapter 4

Implementation Scenario	25
Overview of the Sample Implementation	25
Initial Setup Steps	26
Installing the Sample Files for AS2	26
Importing the Sample Projects	27
Design Steps in Enterprise Designer	28
Setting Up the Sample Environments	28
Viewing and Activating the B2B Host Project	31
Configuring the eXchange Services with Crypto Information	33
Creating and Activating the Project Deployment Profiles	34
Design Steps in eXchange Partner Manager	37
Importing the Trading Partners into the Environments	38
Configuring Trading Partner Parameters for the “Berlin” TP	38
TP “Berlin”: Parameters for the Delivery Channel	39
TP “Berlin”: Parameters for the Internal Delivery Channels	41
Activating the Trading Partner	43
Configuring Trading Partner Parameters for the “Atlanta” TP	43
TP “Atlanta”: Parameters for the Delivery Channel	44
TP “Atlanta”: Parameters for the Internal Delivery Channels	46
Activating the Trading Partner	47
Runtime Steps	48
Starting the Logical Hosts	48
Supplying the Input Data	49
Using the Message Tracking Facility	50

Glossary

Glossary of Acronyms 55

57

Index 58

List of Figures

Figure 1	AS2 Message Structure	13
Figure 2	eXchange and the ICAN Suite	15
Figure 3	eXchange Architecture	17
Figure 4	Enterprise Manager ADMIN Page: Uploading the ProductsManifest.xml File	21
Figure 5	Update Center Wizard: Select Modules to Install	22
Figure 6	Update Center Wizard: Progress Bars	23
Figure 7	Update Center Wizard: Restart Enterprise Designer	23
Figure 8	Project Explorer Tree, After Importing All Sample Projects for AS2	28
Figure 9	Sample Environment, Before Configuration	30
Figure 10	Auto-Mapping Services and eWays	32
Figure 11	Deployment Profile for B2B Host, Before Activation	32
Figure 12	Configuring Private Keys and Trust Stores	34
Figure 13	Deployment Profile for Error Processing	34
Figure 14	Deployment Profile for AS2_Protocol	36
Figure 15	Initial ePM Screen	38
Figure 16	Delivery Channel for "Berlin": General Properties for AS2_Berlin_http1	40
Figure 17	Trading Partner "Atlanta" (in Berlin_Env)	44
Figure 18	Message Tracking, on Startup	51
Figure 19	Message Tracking, Showing Initial Search Results	52
Figure 20	Message Tracking, Showing Filtered Results	53
Figure 21	Message Tracking, Showing Package Details and Message Content	54

List of Tables

Table 1	Document Conventions	9
Table 2	JRE Versions Listed by Operating System	24
Table 3	B2B Protocol Processes Used in the AS2 Sample Implementation	25
Table 4	Delivery Channel Parameters for Trading Partner “Berlin”	39
Table 5	ToPartnerPackaging Parameters for Trading Partner “Berlin”	41
Table 6	FromPartnerPackaging Parameters for Trading Partner “Berlin”	41
Table 7	Trading Partner “Berlin”, Internal Delivery Channel idc_SendToInternal	42
Table 8	Trading Partner “Berlin”, Internal Delivery Channel idc_RecvFromInternal	42
Table 9	Sender Transport Parameters for Trading Partner “Berlin”	42
Table 10	Sender Transport Parameters for Trading Partner “Berlin”	42
Table 11	Delivery Channel Parameters for Trading Partner “Atlanta”	44
Table 12	ToPartnerPackaging Parameters for Trading Partner “Atlanta”	45
Table 13	FromPartnerPackaging Parameters for Trading Partner “Atlanta”	45
Table 14	Trading Partner “Atlanta”, Internal Delivery Channel idc_SendToInternal	46
Table 15	Trading Partner “Atlanta”, Internal Delivery Channel idc_RecvFromInternal	46
Table 16	Sender Transport Parameters for Trading Partner “Atlanta”	47
Table 17	Sender Transport Parameters for Trading Partner “Berlin”	47

Introduction

This chapter introduces you to this guide, its general purpose and scope, and its organization. It also provides sources of related documentation and information.

What's in This Chapter

- [About This Document](#) on page 8
- [Related Documents](#) on page 10
- [SeeBeyond Web Site](#) on page 10
- [SeeBeyond Documentation Feedback](#) on page 10

1.1 About This Document

This section provides a brief outline of this user's guide.

1.1.1. Organization of Information

This guide includes the following chapters:

- [Chapter 1, "Introduction"](#) provides an overview of this document's contents, purpose, conventions, and related documents.
- [Chapter 2, "Introducing AS2 Manager"](#) discusses general features and architecture of eXchange and AS2 Manager.
- [Chapter 3, "Installing AS2 Manager"](#) lists prerequisite that must be met before installation, and provides step-by-step instructions for installing the product files for AS2 Manager and setting it up for use.
- [Chapter 4, "Implementation Scenario"](#) provides step-by-step procedures for creating implementation scenarios that provide a sample of how eXchange can be used to achieve B2B solutions.

In addition, there is a ["Glossary of Acronyms" on page 55](#) that lists and explains the acronyms used in this guide.

1.1.2. Scope, Audience, and Purpose

This user’s guide provides instructions and background information for all users of the AS2 Manager Composite Application. This guide is designed for managers, system administrators, and others who use AS2 Manager.

The purpose of this guide is to help you do the following:

- Understand the nature and function of AS2 Manager.
- Understand the relationship of AS2 Manager to other components of the SeeBeyond Integrated Composite Application Network (ICAN) Suite.
- Learn about the AS2 Manager components and how to use them in your environment.

1.1.3. Document Conventions

The following conventions are observed throughout this document.

Table 1 Document Conventions

Text	Convention	Example
Names of buttons, files, icons, parameters, variables, methods, menus, and objects	Bold text	<ul style="list-style-type: none"> ▪ Click OK to save and close. ▪ From the File menu, select Exit. ▪ Select the logicalhost.exe file. ▪ Enter the timeout value. ▪ Use the getClassname() method. ▪ Configure the Inbound File eWay.
Command line arguments, code samples	Fixed font. Variables are shown in <i>bold italic</i> .	<code>bootstrap -p <i>password</i></code>
Hypertext links	Blue text	See “Document Conventions” on page 9
Hypertext links for Web addresses (URLs) or email addresses	Blue underlined text	http://www.seebeyond.com docfeedback@seebeyond.com

1.1.4. 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.

1.2 Related Documents

For more information about eXchange and the ICAN Suite, refer to the following:

Title	Filename
<i>SeeBeyond ICAN Suite Installation Guide</i>	ICAN_Install_Guide.pdf
<i>eGate Integrator User's Guide</i>	eGate_User_Guide.pdf
<i>Oracle eWay Intelligent Adapter User's Guide</i>	Oracle_eWay.pdf
<i>HTTP(S) eWay Intelligent Adapter User's Guide</i>	HTTPS_eWay.pdf
<i>eXchange Integrator User's Guide</i>	eXchange_User_Guide.pdf
Readme for ICAN 5.0.5	Readme.txt

Also see the URLs provided in **"Additional References" on page 14**.

1.3 SeeBeyond Web Site

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

<http://www.seebeyond.com>

1.4 SeeBeyond Documentation Feedback

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

docfeedback@seebeyond.com

Introducing AS2 Manager

This chapter provides a general overview of AS2 Manager and its place in the ICAN Suite, including system descriptions, general operation, and basic features.

What's in This Chapter

- [About AS2](#) on page 11
- [About eXchange Integrator](#) on page 14

For more information about eGate and eXchange, see the corresponding user's guides.

2.1 About 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.

The AS2 specification describes how applications communicate Electronic Data Interchange (EDI) transaction data over the Internet using the HyperText Transfer Protocol (HTTP), in a secure and interoperable manner. AS2 emphasizes these key aspects of data security:

- Privacy
- Data integrity
- Authenticity
- Non-repudiation of origin and receipt

AS2 specifies the means to connect and to deliver/validate/reply to data, securely and reliably. The purpose of this guide is to assist you in developing an AS2-compliant eXchange system deployment that is interoperable with other implementations used by your trading partners.

2.1.1 General History and Definitions

AS2 is an extension to Applicability Statement 1 (AS1), the standard for secure message transport based on the Simple Mail Transfer Protocol (SMTP). The extension from AS1 to AS2 consists mainly of compatibility with HTTPS and S/MIME.

Definition of EDI-INT

EDI-INT is an Internet Engineering Task Force (IETF) Working Group that exists to document the requirements and best practices for secure, interoperable EDI. The EDI-INT Requirements document contains sufficient background material to give the EDI community an explanation of the Internet-related issues.

The EDI-INT Requirements and Applicability Statements are general in nature, so they can be applied to all types of eBusiness transfers across non-secure networks. The message payload itself does not have to be EDI. The data being transferred can be in the form of Extensible Markup Language (XML) business documents or any other data format.

Definition of AS1

AS1 is an Applicability Statement that described how then-current Internet standards could be leveraged to achieve EDI-INT using SMTP transport technologies. AS1 was published by the IETF EDI-INT Working Group.

Definition of AS2

AS2 is also an IETF EDI-INT Working Group specification. It extends AS1 to include real-time EDI based on S/MIME and HTTPS—that is, HTTP with the Secure Sockets Layer (SSL). AS2 security constructs are the same as AS1, with the addition of session-based encryption and authentication.

2.1.2 AS2 Message Format

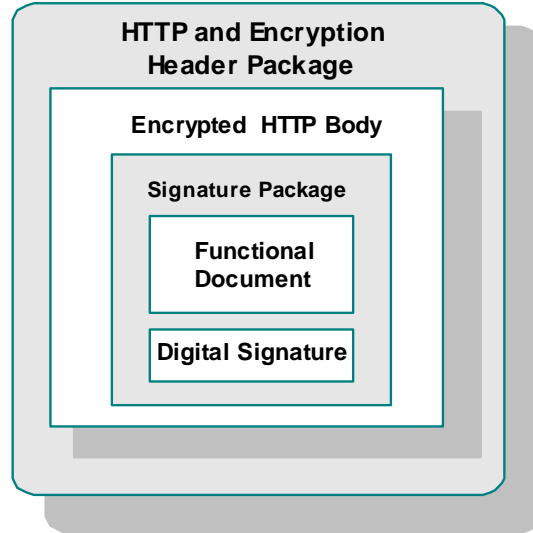
An AS2 message conforms with the following structure:

- HTTP header package: RFC2616/RFC2045
- Encryption package: RFC2633 (application/pkcs7-mime)
 - ♦ Digitally signed package: RFC1847 (multipart/signed) (encrypted)
 - ♦ Message payload: RFC2376 (application/xml) (encrypted)
 - ♦ Digital signature: RFC2633 (application/pkcs7-signature) (encrypted)

The HTTP header is the outermost package, which is supplemented by the headers of the encryption package, which envelopes the signed multipart, which in turn binds the payload and signature parts.

Figure 1 shows a diagram of the basic message format.

Figure 1 AS2 Message Structure



2.1.3. Packaging Layer References

The following list details references for the packaging layers for S/MIME signed, encrypted messages:

- RFC2616
 - ♦ Hypertext Transfer Protocol: HTTP/1.1
<http://www.ietf.org/rfc/rfc2616.txt>
- RFC2045
 - ♦ Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies
<http://www.ietf.org/rfc/rfc2045.txt>
- RFC2633
 - ♦ S/MIME Version 3 Message Specification
<http://www.ietf.org/rfc/rfc2633.txt>
- RFC1847
 - ♦ Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted
<http://www.ietf.org/rfc/rfc1847.txt>

- RFC2376
 - ♦ XML Media Types
<http://www.ietf.org/rfc/rfc2376.txt>
- RFC2298
 - ♦ An Extensible Message Format for Message Disposition Notifications
<http://www.ietf.org/rfc/rfc2298.txt>

2.1.4. Additional References

You can find the current standards-track drafts of the EDI-INT Requirements document, as well as AS2 specifications, at the following Web sites:

- For MIME-based Secure Peer-to-peer Business Data Interchange over the Internet:
<http://www.ietf.org/internet-drafts/draft-ietf-ediint-as1-17.txt>
- For HTTP Transport for Secure Peer-to-peer Business Data Interchange over the Internet:
<http://www.ietf.org/proceedings/02jul/I-D/draft-ietf-ediint-as2-11.txt>
<http://www.ietf.org/proceedings/02mar/I-D/draft-ietf-ediint-as2-10.txt>
- As these documents are updated, links to the most current versions are updated at:
<http://www.ietf.org/html.charters/ediint-charter.html>

Or, if any of the links above are no longer current, go to

<http://www.ietf.org>

and search for documents about AS2.

2.2 About eXchange Integrator

eXchange provides an open B2B protocol framework to support standard EDI and B2B business protocols and enveloping protocols. Not only does it support existing standard protocols, with an extensive set of prebuilt B2B protocol processes, 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 business rules can be added and tailored to address the particular needs of each eBusiness challenge. The tight integration with the rest of the ICAN Suite provides validation, logging, and reporting capabilities, and because each logical step within any business rule is accessible anywhere along the entire BP, the design tools provide complete end-to-end visibility.

The trading partner management facility, eXchange Partner Manager (ePM), is provided via a Web interface. For easy interoperability, trading partner profiles can be exported and imported, and/or configured manually. Each trading partner profile is identified by a unique ID determined by the enterprise, and delivery channels can be

configured for acknowledgments, compression, industry-standard encryption and decryption, and nonrepudiation.

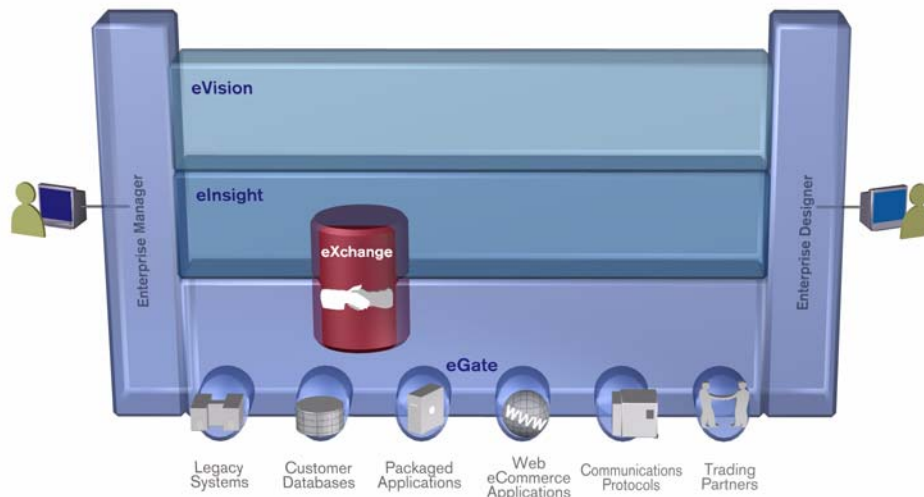
At run time, all steps in the business process, from initial receipt of the message to final delivery to the trading partner, are tracked in real time and also stored in the eXchange database. The Web-based message/package tracker provides tools for retrieving and filtering tracked message and envelope information. Used in conjunction with the other monitoring tools of the ICAN suite, this provides the enterprise with a complete solution for troubleshooting and managing all eBusiness activities.

2.2.1. eXchange and the ICAN Suite

eXchange is part of the SeeBeyond ICAN Suite of products. eXchange provides a Web-based trading partner configuration and management solution for automating and securely managing business partner relationships for real-time interaction between the enterprise and its partners, suppliers, and customers.

eXchange is tightly integrated with the ICAN Suite and runs as a component within the ICAN Suite environment. Figure 2 illustrates how eXchange and other ICAN Suite components work together.

Figure 2 eXchange and the ICAN Suite



2.2.2. eXchange Architectural Overview

eXchange centers around the concept of a *Delivery Channel Profile* for each trading partner relationship. Delivery Channel Profiles are configured within eXchange for use by runtime components. Each profile specifies which B2B protocol(s) to use, where and how to receive inbound messages from trading partners, how to configure and secure messages in this channel, and how and where to deliver outbound messages to trading partners.

eXchange uses the following key components:

- **B2B Host Designer** — Using the **Enterprise Designer** GUI framework, eXchange provides an editor for setting up B2B environments, called the *B2B Host Designer*. Each B2B Host provides one or more protocol-specific delivery channels that are exposed to the eXchange database via the Repository. Delivery channels provided by the B2B Host can then be accessed by specific trading partners and reused.
- **B2B Services and Protocols** — eXchange provides two other special editors: The *eXchange Service Designer*, for modeling the choreography of B2B interactions between the internal system and an external trading partner, as mediated by the B2B Host; and the *eXchange Protocol Designer*, for setting up the message flow logic (*B2B protocol processes*) required to address a specific business challenge, using such activity elements as branching activities, timers, and exception handling.

Prebuilt B2B protocol processes for such industry-standard B2B protocols as AS2 and ebXML are available as separately installable add-on products. eXchange also provides the flexibility of allowing the enterprise to create and configure custom protocol processes.

eXchange also supplies **Channel Manager**, a collection of eXchange Services that provides trading partner-specific integration to the enterprise. Industry-standard transport protocols (FTP, HTTP, HTTPS, SMTP) are supported by Channel Manager and other eWays.

- **Trading Partner Configuration** — eXchange provides a Web-based GUI, eXchange Partner Manager (ePM), for configuring and managing B2B trading partners. Each trading partner has one or more delivery channels that specify the protocols to be used, with corresponding transport mechanisms—encryption parameters such as certificate, signature, and keystore information, acknowledgment-handling preferences, and so forth.
- **eXchange Database** — eXchange uses an Oracle database to mediate retrieval of trading partner information and to store run-time information on message tracking.
- **Message Tracking**— eXchange provides a specific MessageTracker application that can be combined with other processes in a project just by dragging it into the Connectivity Map, as well as a Web-based message tracking GUI with powerful filtering and searching capabilities.

2.2.3. Process Overview

Using eXchange to create a business solution consists of three phases:

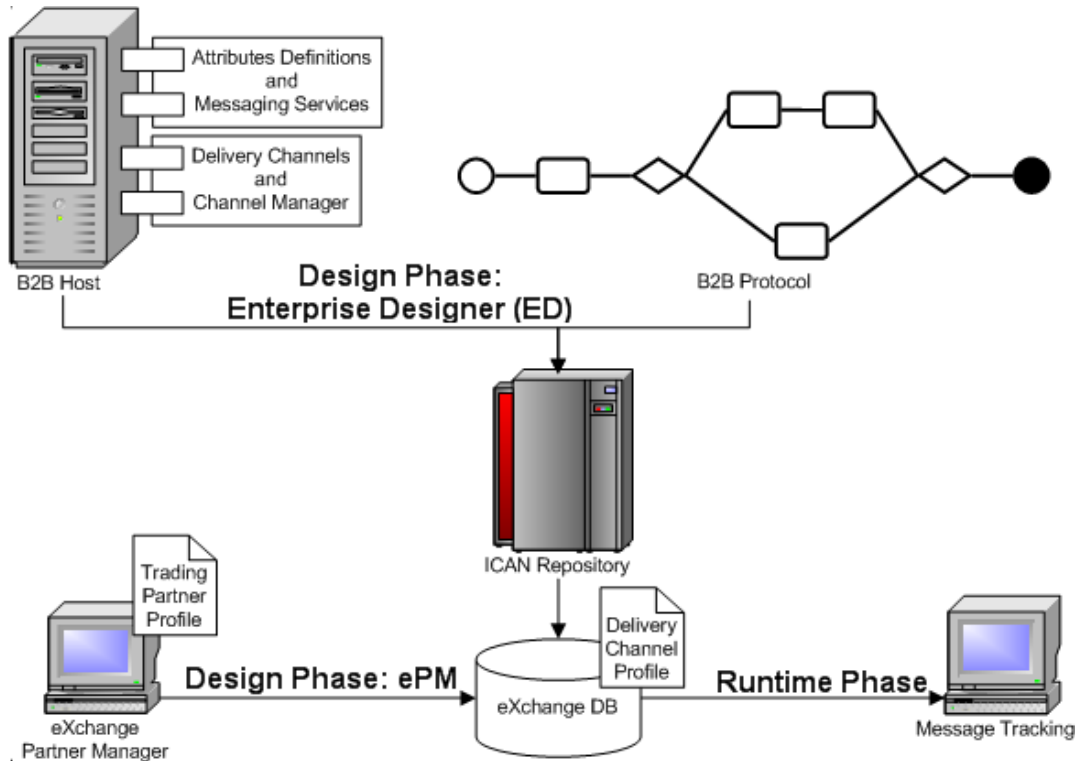
- Design phase within Enterprise Designer
- Design phase within eXchange Partner Manager
- Runtime phase

The purpose of the design phases is to: Create metadata for Delivery Channel Profiles; set up business logic for B2B protocol processes; configure connections with external systems; create and configure trading partners; and associate each trading partner relationship with a Delivery Channel Profile (DCP) configuration. Activating a trading partner exposes its DCP configuration settings to the eXchange database.

At run time, the Logical Host reads the DCP configuration from the database to determine: How to receive and process inbound messages; which business logic to run; and how to process and deliver outbound messages. Results are written to the database, where they can be filtered and viewed by the Message Tracker facility.

These phases are illustrated in Figure 3.

Figure 3 eXchange Architecture



Installing AS2 Manager

This chapter provides the prerequisites and steps for installing AS2 Manager Composite Application.

What's in This Chapter

- [System Requirements](#) on page 18
- [Before You Install](#) on page 19
- [Installing the Product Files](#) on page 19
- [Database Scripts](#) on page 24
- [Additional Policy JAR Files Required to Run SME](#) on page 24

3.1 System Requirements

This section lists system requirements and database requirements. The *SeeBeyond ICAN Suite Installation Guide* and the **Readme.txt** file, available on the product media and via Enterprise Manager (Documentation tab), contain up-to-date operating system requirements for each supported platform.

3.1.1. Supported Operating Systems

AS2 Manager is available on the following operating systems:

- Microsoft Windows 2000 SP3 or SP4, Windows XP SP1a, and Windows Server 2003
- Sun Solaris 8 and Solaris 9, with required patches
- HP Tru64 V5.1A, with required patches
- HP-UX 11.0, 11i (PA-RISC), and 11i v2.0 (11.23) with required patches and parameter changes
- IBM AIX 5.1L and AIX 5.2 (either 64-bit kernel or 32-bit kernel with 64-bit extension), with required maintenance level patches
- Red Hat Linux 8 (Intel x86) and Linux Advanced Server 2.1 (Intel x86)

3.1.2. Database Support

Database for eXchange Partner Management and Message Tracking

The eXchange database is required. It provides a run-time persistent store for trading partner management and message tracking. For eXchange, the following databases are supported:

- Oracle 8.1.7
- Oracle 9.0.1
- Oracle 9.2

Database for Persistence and Monitoring via eInsight Engine

In addition, eXchange can optionally use the eInsight engine (supplied with eXchange) to collect and persist data from your B2B protocol processes. This provides for recovery, and also enables some monitoring and reporting capabilities in Enterprise Manager.

The eInsight engine supports the following databases:

- Oracle 8.1.7, 9.0.1, and 9.2
- Sybase 12.5
- Microsoft SQL Server 2000
- IBM DB2 Universal Database 8.1

3.2 Before You Install

Before you begin installing AS2 Manager, make sure of all the following:

- All projects that will be re-used after the eXchange installation have been exported. Installing this release of eXchange.sar introduces changes to system projects (that is, projects such as SeeBeyond > eXchange and eXchange > Deployment) that delete or permanently modify previous contents of these system projects.
- A Repository server is running on the machine where you will be uploading the product files.

3.3 Installing the Product Files

The steps for installing AS2 Manager are the same as for other products in the ICAN Suite. You can find general product installation instructions in the *ICAN Suite Installation Guide*, which is available on the product media and can also be accessed via Enterprise Manager (Documentation tab).

3.3.1. Uploading AS2 Manager to the Repository

Before you begin

- A Repository server must be running on the machine where you will be uploading the product files.
- The following ICAN **.sar** files must have already been uploaded to this Repository:
 - ♦ eGate Enterprise Designer (**eGate.sar**) 5.0.5
 - ♦ Batch eWay adapter (**BatcheWay.sar**)
 - ♦ Oracle eWay adapter (**OracleeWay.sar**)
 - ♦ File eWay adapter (**FileeWay.sar**) — This not an installation requirement, but it is required by the sample implementation
 - ♦ eXchange Integrator (**eXchange.sar**) 5.0.5
 - ♦ HTTP(S) eWay adapter (**HTTPeWay.sar**)
 - ♦ Secure Message Extension (**SMEWebServices.sar**)

Note: *SMEWebServices.sar* is required for such features as encryption/decryption, signature verification, certificate authentication, and compression/decompression.

To upload AS2 Manager product files to the Repository

- 1 On a Windows machine, start a Web browser and point it at the machine and port where the Repository server is running:

```
http://<hostname>:<port>
```

where

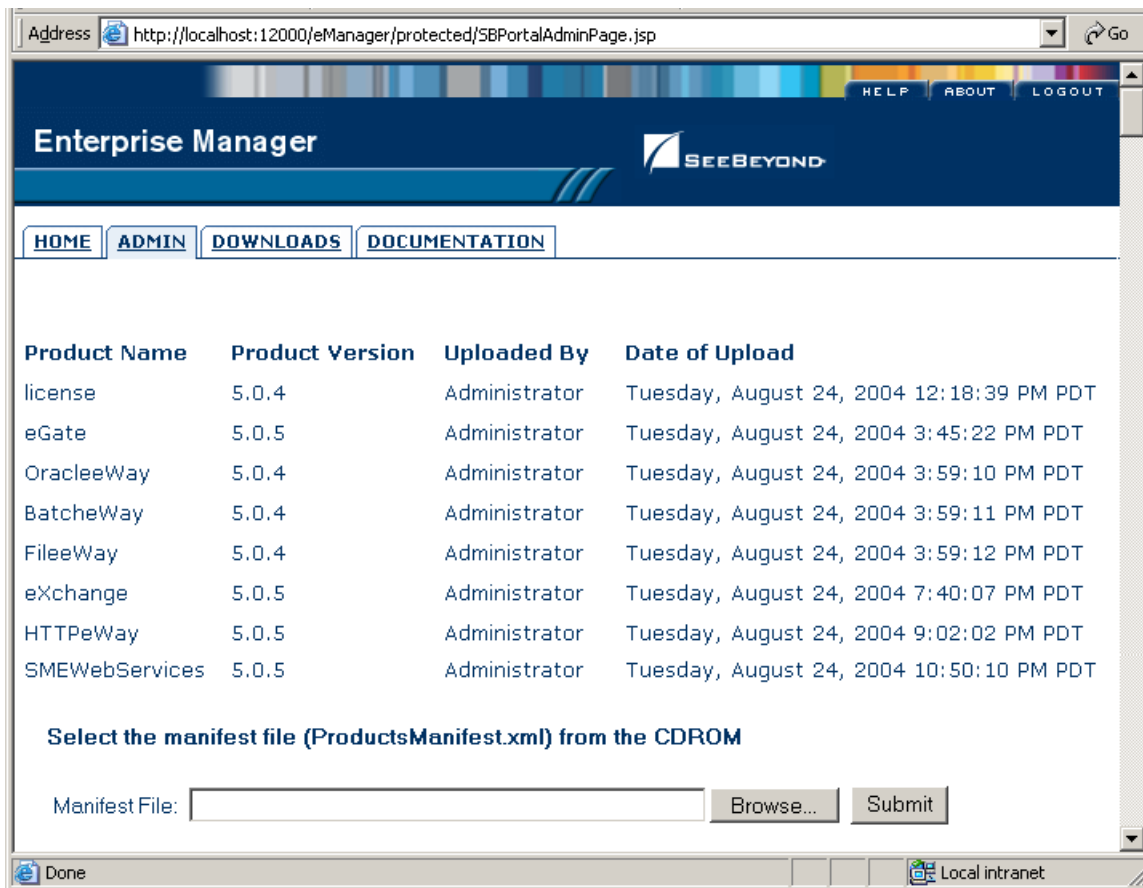
- ♦ *<hostname>* is the name of the machine running the Repository server.
- ♦ *<port>* is the starting port number assigned when the Repository was installed.

For example, the URL you enter might look like either of the following:

```
http://localhost:12001  
http://serv1234.company.com:19876
```

- 2 On the Enterprise Manager **SeeBeyond Customer Login** page, enter your username and password.
- 3 When Enterprise Manager responds, click the **ADMIN** tab. See Figure 4.

Figure 4 Enterprise Manager ADMIN Page: Uploading the ProductsManifest.xml File



- 4 In the ADMIN page, under "Select the manifest file [...]", click **Browse**.
- 5 In the **Choose file** dialog, navigate to the correct directory for the products you want to upload (for example: **CoreProducts** for eXchange, or **Add-ons** for eWays), click the **ProductsManifest.xml** filename, and then click **Open**.
- 6 In the ADMIN page, click **Submit**.
The lower half of the ADMIN page lists the product files you are licensed to upload.
- 7 In the Products column, find **AS2 Manager Composite Application** and then click the **Browse** button for it.
- 8 In the **Choose file** dialog, click **AS2_Manager.sar** and then click **Open**.
- 9 Repeat the previous two steps for other eXchange-related product **.sar** files you are licensed to upload, such as other protocol manager composite applications, or OTD libraries for ASC X12, HIPAA, or UN/EDIFACT.
- 10 For documentation and samples, also upload the corresponding [...] **Docs.sar** files, such as **AS2_ManagerDocs.sar** (and X12_OTD_Docs.sar, and so forth).

3.3.2. Refreshing Enterprise Designer

The following steps are needed only if you have newly uploaded (or re-uploaded) eGate.sar or any other .sar file that affects the Enterprise Designer GUI framework.

Tip: How can you determine whether to use the Update Center? Start Enterprise Designer and, on the **Tools** menu, click **Update Center**; if there are any items under “SeeBeyond 5.0” besides “**Base ESR**”, you need to follow these steps.

Before you begin

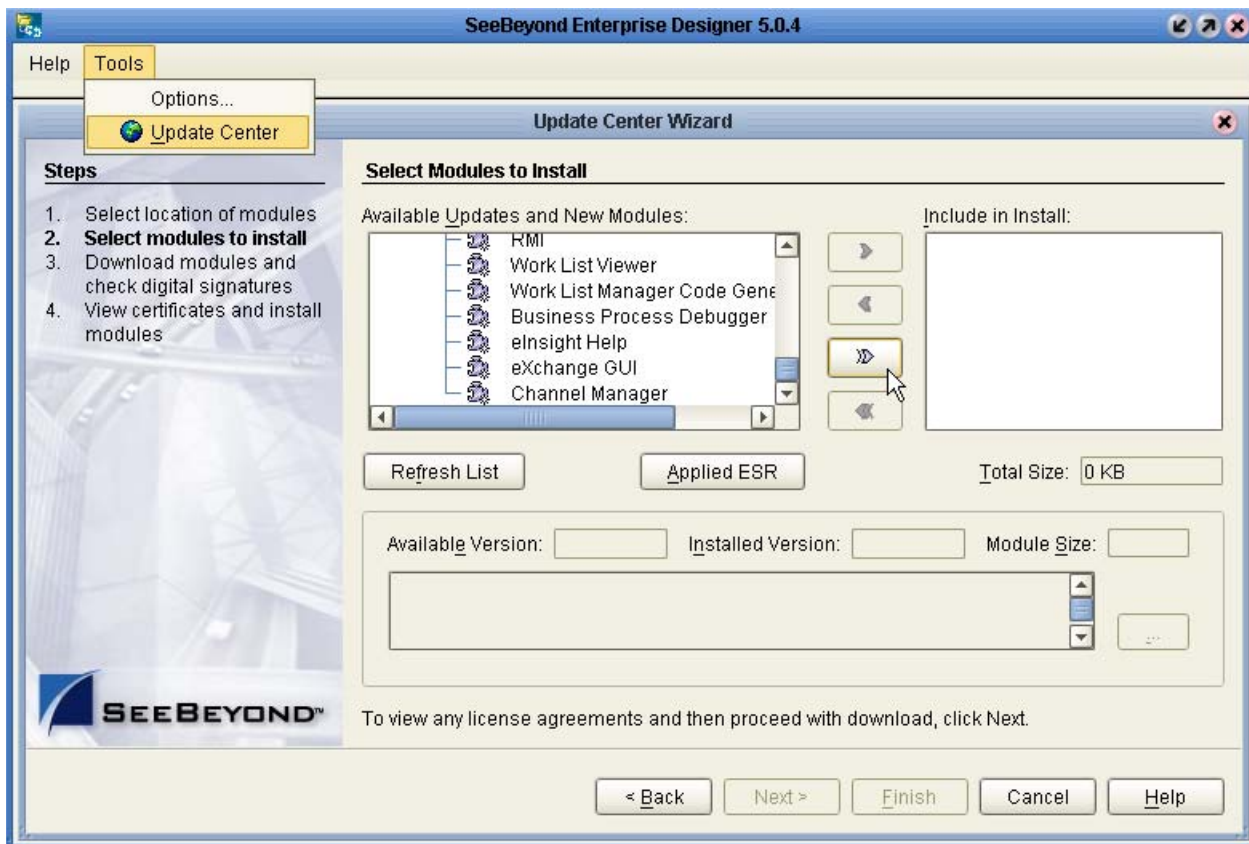
- You must have already downloaded and installed Enterprise Designer.
- A Repository server must be running on the machine where you uploaded the AS2 Manager product files.

To refresh an existing installation of Enterprise Designer

- 1 Start Enterprise Designer.
- 2 On the **Tools** menu, click **Update Center**.

The Update Center shows a list of components ready for updating. See Figure 5.

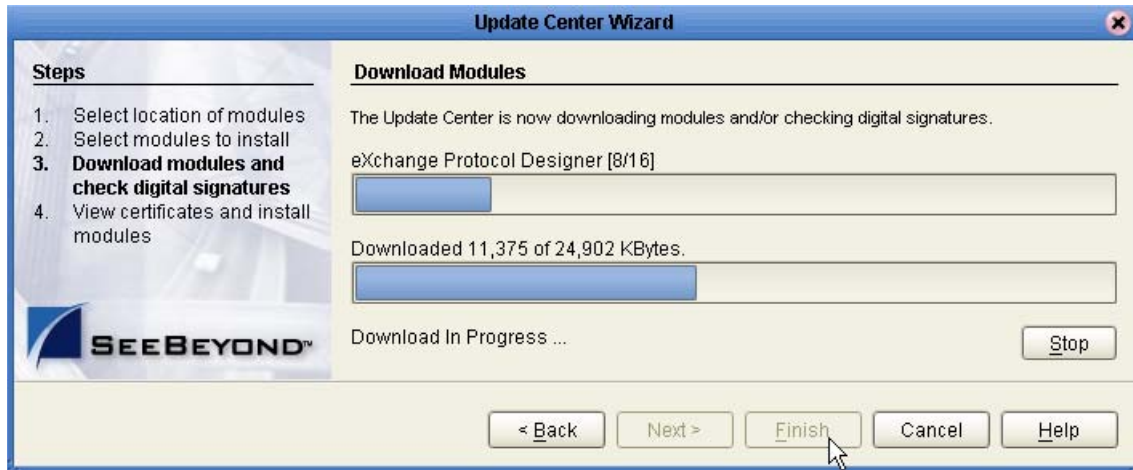
Figure 5 Update Center Wizard: Select Modules to Install



Note: Depending on what products you have installed, and how they are configured, the screenshots pictured may differ from what you see on your system.

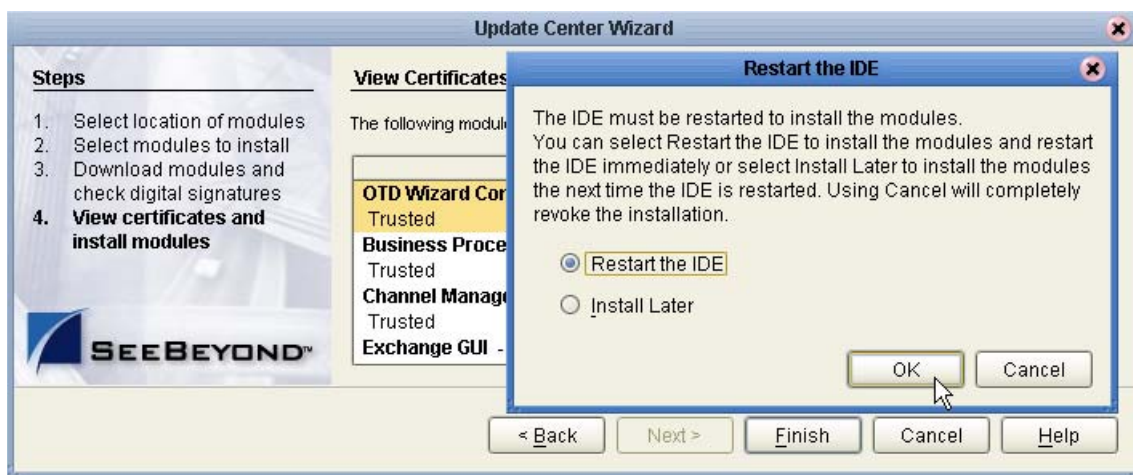
- 3 Click **Add All** (the button with a doubled chevron pointing to the right).
All modules move from the Available/New pane to the **Include in Install** pane.
- 4 Click **Next** and, in the next window, click **Accept** to accept the license agreement.
The wizard shows you the progress of the download. See Figure 6.

Figure 6 Update Center Wizard: Progress Bars



- 5 When the progress bars indicate the download has ended, click **Next**.
- 6 Review the certificates and installed modules, and then click **Finish**.
- 7 When prompted to restart Enterprise Designer, click **OK**. See Figure 7.

Figure 7 Update Center Wizard: Restart Enterprise Designer



When Enterprise Designer restarts, the installation of AS2 Manager Composite Application is complete, and you can use all eXchange tools provided on the Enterprise Designer framework.

3.4 Database Scripts

The eXchange database schema collects and persists data about your trading partner profiles, and also allows you to track message delivery history. eXchange provides database scripts to create and upgrade these database schemas for eXchange. For information, see the *eXchange Integrator User's Guide*.

3.5 Additional Policy JAR Files Required to Run SME

For strong encryption/decryption and signatures/verification, using libraries supplied with Secure Messaging Extension (SME), you must download and apply additional policy .jar files. The type of .jar files required depends on the JVM are using. Refer to your JVM vendor for exact details on the specific policy .jar file requirements.

Use Table 2 to determine which JRE is included in the eGate logical host.

Table 2 JRE Versions Listed by Operating System

Operating System	JRE	URL
Windows, Solaris, HP-UX, Linux, Tru64	1.4.2	http://java.sun.com/j2se/1.4.2/download.html
AIX	1.4.1	http://java.sun.com/products/archive/j2se/1.4.1_07/index.html

To download the required policy.jar files

- 1 Scroll to the bottom of the web page listed in Table 2 for your logical host's JRE.
- 2 Click the [DOWNLOAD](#) link for **Unlimited Strength Jurisdiction Policy Files 1.4.2**. (or, for AIX, **Unlimited Strength Jurisdiction Policy Files 1.4.1**).
- 3 Click the link to download the .zip file containing the required policy .jar files (in other words: for JRE 1.4.2, follow the link [Download jce policy-1 4 2.zip](#); or, for JRE 1.4.1, follow the link [Download jce policy-1 4 1.zip](#))
- 4 Extract the following required policy .jar files:
 - ♦ local_policy.jar
 - ♦ US_export_policy.jar
- 5 Then, for each of your logical hosts, replace the versions of these files in:


```
<logicalhost>\jre\lib\security\
```
- 6 In addition, if you are running a repository on AIX, also replace the versions of these files in:


```
<AIXrepository>/jre/1.4.x/lib/security/
```

For complete information on SME, see the *Secure Messaging Extension User's Guide*.

Implementation Scenario

This chapter provides a sample scenario showing how eXchange can be used to achieve B2B solutions using the AS2 protocol.

What's in This Chapter

The steps for the sample implementation occur in these phases:

Initial Setup Steps	In these steps, you ensure that prerequisites are met, obtain the necessary sample materials, extract sample files, and import the sample projects. See section 4.1 starting on page 26 .
Design Steps in Enterprise Designer	In these steps, you use Enterprise Designer to add and configure externals, view components in the B2B host project, and activate the host, creating an eXchange Service. Then you view the components in the main project and activate it. See section 4.2 starting on page 28 .
Design Steps in eXchange Partner Manager	In these steps, you use the eXchange Partner Management (ePM) facility to import trading partners, view and specify parameter values, and activate them. See section 4.3 starting on page 37 .
Runtime Steps	In these steps, you bootstrap two logical hosts, apply the activated deployments, stage sample input data, and use the Message Tracking facility to follow the progress of messages, as well as acknowledgments and errors, tracking messages as they are transported from one end point, received and handled by inbound B2B protocol processes (de-enveloping, decryption, decompression, signature verification), and outbound B2B protocol processes (enveloping, encryption, compression, digital signing), sent out, and transported to another end point. See section 4.4 starting on page 48 .

Overview of the Sample Implementation

The sample implementation demonstrates AS2 in conjunction with SME (for crypto and compression) for inbound and outbound message processing between two parties: "Atlanta" (=companyA=Company #1) and "Berlin" (=companyB=Company #2). It makes use of the following B2B protocol processes supplied by SeeBeyond:

Table 3 B2B Protocol Processes Used in the AS2 Sample Implementation

For Inbound Messages	For Outbound Messages	Other
AS2InboundBP	AS2OutboundBP	
AS2 Inbound	AS2 Outbound	OutboundHTTPClient
ToInternalDeliverySelector		ErrorHandlerSelector

Table 3 B2B Protocol Processes Used in the AS2 Sample Implementation (Continued)

For Inbound Messages	For Outbound Messages	Other
ToInternalFileDeliveryProtocol		JMSErrorHandler
ToInternalJMSDeliveryProtocol		
AS2 Unpackager	AS2 Packager	
Base64Decode		
Base64Encode		
AS2HTTPHeader	AS2HTTPHeaderPackager	
DecryptMessage*	EncryptMessage*	
VerifyMessageSignature*	SignMessage*	
DecompressMessage*	CompressMessage*	
UnpackagePayload	PackagePayload	
UnpackageMDN		
CalculateMICDigest		
AS2 Outbound MDN		
GenerateMDN		
*—requires Secure Messaging Extension (SME).		

4.1 Initial Setup Steps

In this section

- [Installing the Sample Files for AS2 on page 26](#)
- [Importing the Sample Projects on page 27](#)

4.1.1 Installing the Sample Files for AS2

These steps assume the existence of a temporary eXchange directory for sample files, such as **C:\temp\exchange**. You will extract the sample files to this directory so that you can conveniently access the files in later procedures.

To install the sample files

Before you begin: Your repository must already be running, and you must be logged in to Enterprise Manager. If you have already uploaded the documentation for eXchange, you can skip steps 1 and 2 and start with step 3.

- 1 In the ADMIN tab, if you have not already done so, browse to the [...] \Documentation \ **ProductsManifest.xml** file and submit it.
- 2 In the ADMIN tab, if you have not previously done so, browse to the **AS2_ManagerDocs.sar** file, select it, and click the **upload now** button.

- 3 In the DOCUMENTATION tab, in the Products window, click **AS2 Manager Composite Application**.
- 4 In the window that appears on the right side, click **Download Sample**.
- 5 Preserving file paths, extract the files to your **C:\temp\exChange** directory.

Result: The following directories and files are created:

```
C:\temp\exChange\Sample\Projects\AS2_AllProjects.zip
C:\temp\exChange\Sample\Projects\AS2_B2B Templates.zip
C:\temp\exChange\Sample\Projects\AS2_Host.zip
C:\temp\exChange\Sample\Projects\AS2_Protocol.zip
C:\temp\exChange\Sample\Projects\JmsErrorToFileReceiver.zip
C:\temp\exChange\Sample\Projects\JmsPayloadToInternalReceiver.zip
C:\temp\exChange\Sample\Projects\Atlanta_Env.zip
C:\temp\exChange\Sample\Projects\Berlin_Env.zip
C:\temp\exChange\Sample\TradingPartners\AS2_TP_for_Atlanta_Env.xml
C:\temp\exChange\Sample\TradingPartners\AS2_TP_for_Berlin_Env.xml
C:\temp\exChange\Sample\Data\AS2_Atlanta_binary-payload.~in
C:\temp\exChange\Sample\Data\AS2_Atlanta_x11-payload.~in
C:\temp\exChange\Sample\Data\AS2_Berlin_binary-payload.~in
C:\temp\exChange\Sample\Data\AS2_Berlin_x11-payload.~in
C:\temp\exChange\Sample\Data\Out\
```

Note: If you extract the data files to a different location, modifications will be required, in Enterprise Designer (the value of the “Data Root Directory” project variables) and in exChange Partner Manager (fields in the ToPartner and FromPartner tabs).

4.1.2 Importing the Sample Projects

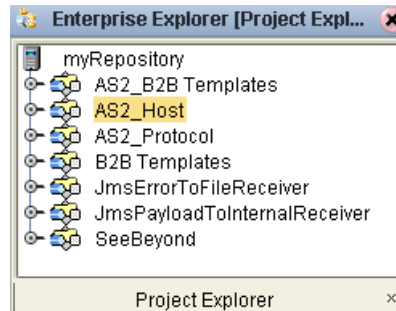
To import the sample projects

Before you begin: Your repository must already be running, and you must be logged in to Enterprise Designer. If your repository already has a project at root level whose name is identical to any of the projects you will be importing, you must delete or rename it before you start; and if you import the environments (rather than re-creating them), ensure the repository does not already contain any environment of the same name.

- 1 In Project Explorer, right-click the repository and, on the popup menu, click: **Import**
- 2 In the **Import Manager** dialog, browse to the folder where you installed the sample files (such as C:\temp\exChange\Sample\Projects) and do one of the following:
 - ♦ Select **AS2_AllProjects.zip**, click **Open**, exclude the B2B Templates project, and then click **Import**. (This is the quickest but least instructive way to get started.)
 - ♦ One by one, in order, open each of the following and import it:
 - ♦ AS2_B2B Templates.zip (required)
 - ♦ AS2_Host.zip (required)
 - ♦ AS2_Protocol.zip (required)
 - ♦ JmsErrorToFileReceiver.zip (required)
 - ♦ JmsPayloadToInternalReceiver.zip (required)
 - ♦ Atlanta_Env.zip and Berlin_Env.zip (optional)
- 3 Close the Import Manager dialog.

Result: The Project Explorer tree displays five new projects: AS2_B2B Templates, AS2_Host, AS2_Protocol, JmsErrorToFileReceiver, and JmsPayloadToInternalReceiver. See Figure 8.

Figure 8 Project Explorer Tree, After Importing All Sample Projects for AS2



4.2 Design Steps in Enterprise Designer

For the AS2 sample implementation, design-time steps in Enterprise Designer consist of the following:

- [Setting Up the Sample Environments on page 28](#)
- [Viewing and Activating the B2B Host Project on page 31](#)
- [Configuring the eXchange Services with Crypto Information on page 33](#)
- [Creating and Activating the Project Deployment Profiles on page 34](#)

4.2.1 Setting Up the Sample Environments


The sample assumes you will use default configurations for all servers where possible, and that you will make any changes where needed. For example:

- **Oracle:** You must create a new outbound Oracle external and configure it, even if you imported the sample environment; sample parameters are for reference only.
- **Ports:** To run anything other than a SeeBeyond Integration Server on ports 18000–18009, make adjustments in steps 3 (ports) and/or 5 (type of Integration Server).
- **HTTPS:** To configure your HTTP server or client to use SSL, see the *HTTP(S) eWay Intelligent Adapter User's Guide* for eWay settings and Integration Server Web Server configuration.

If you imported the sample environments, you can skip past the following procedure (unless you need to adjust base ports and/or configure the HTTP externals to use HTTPS) and continue with the Oracle configuration steps [on page 30](#).

To create the sample environment

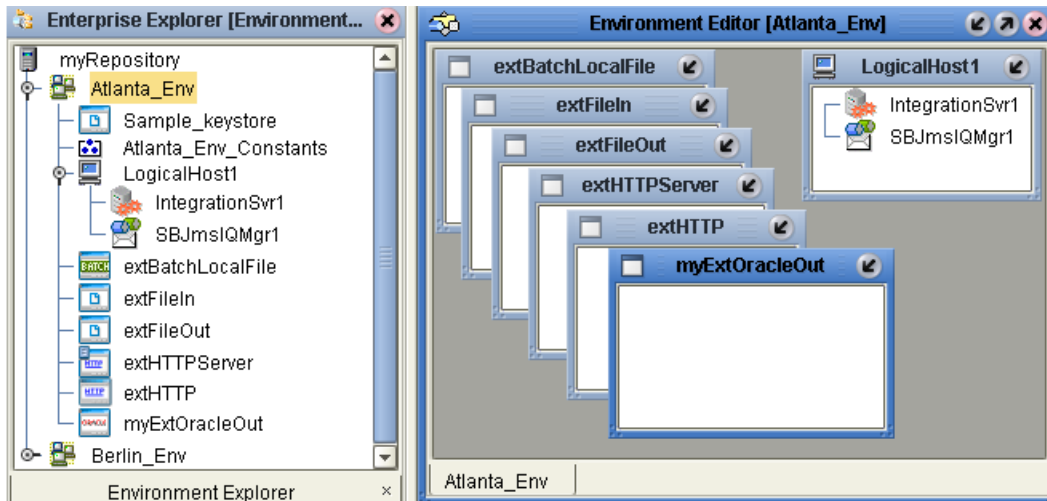
- 1 In Enterprise Designer, near the lower left of the window, click the **Environment Explorer** tab.

- 2 In the Environment Explorer tree, right-click the repository and, on the popup context menu, click **New Environment**
 - ♦ Rename the newly created environment to **Atlanta_Env**.
 - 3 Right-click Atlanta_Env and, on the menu, click: **New Logical Host**
 - ♦ Retain the default name: LogicalHost1
 - 4 For second and subsequent logical hosts—such as Berlin—change the base port:
 - A Right-click LogicalHost1 and open its properties.
 - B Click **Logical Host Configuration** and change the value **Logical Host Base Port** to a larger multiple of 1000. The sample assumes that for Atlanta, you will retain the default (18000); and for Berlin, you will change the base port to **28000**.
 - C Close the properties sheet.
 - 5 Right-click LogicalHost11, click **New SeeBeyond Integration Server**, and retain the default name: IntegrationSvr1
 - 6 Right-click LogicalHost1 and click: **New SeeBeyond JMS IQ Manager**, and retain the default name: SBJmsIQMgr1
 - 7 Right-click Atlanta_Env > **New BatchLocalFile External System**
 - ♦ Name it **extBatchLocalFile** and click OK.
 - 8 Right-click Atlanta_Env > **New File External System**
 - ♦ Name it **extFileIn**, set it to **Inbound File eWay**, and click OK.
 - 9 Right-click Atlanta_Env > **New File External System**
 - ♦ Name it **extFileOut**, set it to **Outbound File eWay**, and click OK.
 - 10 Right-click Atlanta_Env > **New HTTP External System**
 - ♦ Name it **extHTTP** and click OK.
 - 11 Right-click extHTTP > **Properties** to initialize its properties, and then click OK.
 - 12 Right-click Atlanta_Env > **New HTTP Server External System**
 - ♦ Name it **extHTTPServer** and click OK.
- Tip:** To configure for HTTPS, see the HTTP(S) eWay Intelligent Adapter User's Guide.
- 13 Right-click Atlanta_Env > **New Oracle External System**
 - ♦ Name it **myExtOracleOut**, designate it **Outbound Oracle eWay**, and click OK.
 - 14 Right-click Atlanta_Env > **New Constant**
 - ♦ In the **Create a Constant** dialog, name it **Data Root Directory**, give it the value **C:\temp\exChange\Sample\Data\Atlanta** (if you installed sample data files elsewhere than C:\temp\exChange\Sample\Data, substitute appropriately), and click **OK**; then close the **Variables and Constants** dialog box.
 - 15 On the main toolbar, click  **Save All**.
 - 16 For Berlin: Repeat all previous steps, substituting "Berlin_Env" for "Atlanta_Env"; in step 4, be sure to change the base port to 28000; in step 14, use "...\Data\Berlin".

Result: The environments, named **Atlanta_Env** and **Berlin_Env**, have all but two of the externals needed by the projects. Steps for configuring outbound Oracle external are

provided in the following procedure. The final externals will be created by activating the project containing the B2B host and then providing keystore information.

Figure 9 Sample Environment, Before Configuration




To create and configure the Oracle external

Before you begin: Your eXchange 5.0.5 Oracle database must be accessible, and you must know its SID and username/passwords.

Note: This sample assumes that in your Oracle database, you have already created two new username/password combinations: for Atlanta, the sample is preconfigured with username/password = *ex505_A/ex505_A*; for Berlin, the sample is preconfigured with username/password = *ex505_B/ex505_B*. (For details on running the *createuser.sql* script to create username/password combinations and running the *createdb.sql* script to populate tablespaces for each user, see the eXchange User's Guide section on "Running Database Scripts to Set Up the eXchange Database".) For convenience, this setup allows you to run the sample on a single machine where two trading partners ("A" and "B") are simulated using two different database usernames and two different logicalhost port ranges; alternatively (and more realistically), you could create separate databases on separate machines.

- 1 In the Environment Explorer tree, open Atlanta_Env, right-click myExtOracleOut, and configure properties appropriately. For example:
 - ◆ **DatabaseName:** *exch50* (change this to the SID for your eXchange Oracle database)
 - ◆ **DataSourceName:** **local**
 - ◆ **Password:** (replace this with the password for your eXchange 5.0.5 database user "A")
 - ◆ **PortNumber:** **1521** (change this only if your Oracle administrator changed the default)
 - ◆ **ServerName:** *myMachine* (change this to the hostname of the Oracle server machine)
 - ◆ **User:** *ex505_A* (change this to username "A" for your eXchange 5.0.5 database)
- 2 When all properties have been configured correctly for your site, click OK.

- 3 For Berlin: Repeat all previous steps, but instead of “Atlanta”, substitute “Berlin” and instead of “A”, substitute “B”.
- 4 Collapse all environment trees, select Atlanta_Env, click  **Save All**, and close all canvases.

Result: The environments are now well-configured; they contain all externals needed for activation of the B2B Host. You can export the environments at this point if you like; you should not export an environment after it contains an eXchange service.

The final externals—eXchange services for the Atlanta and Berlin environments—will be created by activating the project that contains the B2B host.

4.2.2 Viewing and Activating the B2B Host Project

In the Project Explorer tree, open the B2B host project (named **AS2_Host**) to display its components. Below is a summary of the B2B host’s contents. Activating this project will create an eXchange service that acts as a channel manager and provides a connection to the message tracking application and the eXchange database.

Components of the B2B host project (AS2_Host)

- **myHost_MsgingService** is the only message service used by the B2B host. It contains four messaging actions, alternating inbound and outbound.
- **AS2_Host** is the B2B host itself:
 - ♦ Its Business Protocols window references one service, **myHost_MsgingService**. (To find this, open the Messaging Attributes Definitions (MADs) folder and open the MAD named “AS2”.)
 - ♦ It defines only one external delivery channel (for the **AS2** MAD), named **AS2_DelivCh1_http**. For transport to and from trading partners, this channel references the standard SeeBeyond-supplied HTTP transport attributes definitions (TADs).
 - ♦ It defines two internal delivery channels (IDCs):
 - ♦ **idc_SendToInternal** uses the FILE TAD to communicate in the Sender (toInternal) direction
 - ♦ **idc_RecvFromInternal** uses the ChannelManagerFile TAD to communicate in the Receiver (fromInternal) direction,
- **Host_CMap** is the map whose activation will create the eXchange service.
 - ♦ Its only input is an instance of AS2_Host, with two outbound connections.
 - ♦ Its only output is an instance of Oracle, with two inbound connections.
 - ♦ Connecting to both is an instance of a SeeBeyond-supplied tracking application.

To activate the B2B host, creating the eXchange service

Before you begin: Your environment must contain a well-configured Oracle external (see the preceding procedure), and your Oracle database services must be running.

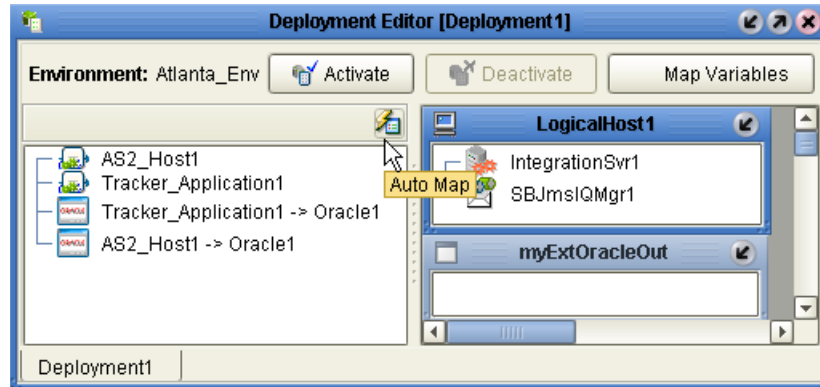
- 1 Right-click the **AS2_Host** project and, on the popup context menu, point at **New** and click **Deployment Profile**

- Keep the default name (**Deployment1**), point it at **Atlanta_Env**, and click OK. (The sample assumes that Deployment1 points at Atlanta_Env and Deployment2 points at Berlin_Env. If you vary from these assumptions, you will need to make corresponding changes in ePM.)

The Deployment Editor opens. Its left pane has two services and two Oracle eWays.

- Click the  **Auto Map** button; see Figure 10.

Figure 10 Auto-Mapping Services and eWays



The services are assigned to the Integration Server, and the eWays to the Oracle external; see Figure 11.

Tip: *If myExtOracleOut refuses to accept eWays, it may be an indication of:*


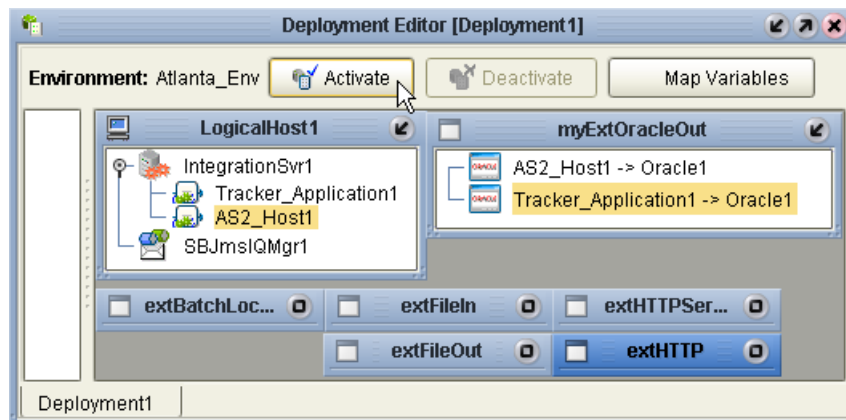
- ♦ *The Oracle database instance it references is inaccessible. Ensure it is running and that the myExtOracleOut properties match its hostname, SID, username, and password. If necessary, see [“To create and configure the Oracle external” on page 30](#).*
- ♦ *It was misdefined as inbound. Delete myExtOracleOut and re-create it as outbound. Then: Click **Save All**, followed by  **Refresh All from Repository**.*

Figure 11 Deployment Profile for B2B Host, Before Activation



- Click  **Save All** and then click **Activate**.


- 5 In response to the dialog box, click **No**; that is, do *not* apply to Logical Host(s).

Result: A new external is created, named **AS2_Host1 eXchange Service**.

- 6 For Berlin: Open Environment Explorer and select **Berlin_Env**. Then return to Project Explorer and repeat all previous steps, but with the following substitutions: make sure the deployment profile is named **Deployment2**, and make sure it references **Berlin_Env**.

Result: Another new external is created, also named **AS2_Host1 eXchange Service**.

Tip: *If either or both of the eXchange Service appear misplaced in the environment tree, it reflects either an uncollapsed tree or a mis-selected environment upon activation. This display error will correct itself when the Repository is refreshed.*

Result: The environments now have all the externals they need. Save all of your work, close all canvases, and click  **Refresh All from Repository**.

4.2.3 Configuring the eXchange Services with Crypto Information

Since the sample assumes you will be using cryptographic features (encryption, decryption, signatures and verifications), additional steps are required for configuring the eXchange services.

To associate the eXchange service with the keystore information

- 1 In the Environment Explorer tree, open **Atlanta_Env**, right-click its **AS2_Host1 eXchange Service** and, on the popup context menu, click **Properties** to configure the public and private keys for the B2B host's delivery channel (**AS2_DelivCh1_http**).

- 2 For Signature Key, click **companya_pvtkey**

Tip: *If companya_pvtkey does not appear in the drop-down list, click the ellipsis [...] and, in the Signature Key dialog, click **Import**. Using alias **companya_pvtkey**, import C:\temp\eXchange\Sample\Crypto\CompanyA-Key.p12 with the following password (all-lowercase): **companya***

- 3 Optionally, for Signature Trust Store, click **mytrust**

Tip: *If mytrust does not appear in the drop-down list, click the ellipsis [...] and, in the Signature Trust Store dialog, click **New**, enter the alias **mytrust**, and click **OK**.*

- 4 For Decryption, click **companya_pvtkey**

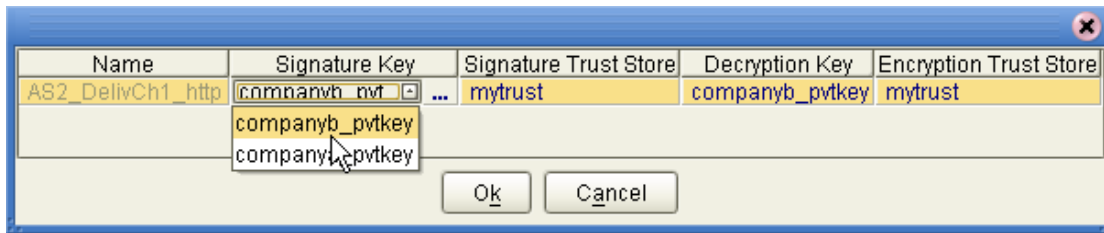
- 5 Optionally, for Encryption Trust Store, click **mytrust**

- 6 Click **OK** to close the dialog box.

- 7 If the logical host were running, you would also need to do the following:
In the environment tree, right-click **LogicalHost1** and, on the menu, click **Apply**

- 8 For Berlin: Repeat all previous steps, but with the following substitutions: open Berlin_Env and populate it with Company "B" information: **companyb_pvtkey** (if necessary, after importing **CompanyB-Key.p12** with password **companyb**). See Figure 12.

Figure 12 Configuring Private Keys and Trust Stores



9 Click **Save All**, and then close all canvases.

Result: Cryptographic information is now associated with the **AS2_DelivCh1_http** delivery channels for each of the two eXchange services.

4.2.4. Creating and Activating the Project Deployment Profiles

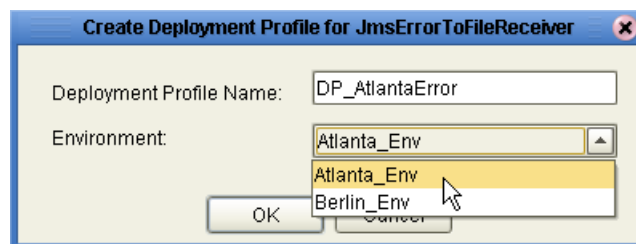
Before you begin: Both environments (Atlanta_Env and Berlin_Env) must already contain eXchange services that have been configured with crypto information; if necessary, see the [procedure on page 31](#) and/or the [procedure on page 33](#).

After you create and activate deployment profiles for the B2B host project (AS2_Host), you must create/activate two deployment profiles for each of the other three projects: JmsErrorToFileReceiver, JmsPayloadToInternalReceiver, and AS2_Protocol. In each case, the first-created deployment profile points to Atlanta_Env and the second-created points to Berlin_Env. Four of the six profiles use mapped variables.

To activate the JmsErrorToFileReceiver project


- 1 In the Project Explorer tree, right-click **JmsErrorToFileReceiver** and, on the popup context menu, point at **New** and click **Deployment Profile**
- 2 Rename it to **DP_AtlantaError**, point it at **Atlanta_Env**, and click OK. See Figure 13.

Figure 13 Deployment Profile for Error Processing



The Deployment Editor's left pane displays: A service (JmsErrorToFileBP1); a topic (HostErrorNotificationTopic->Jms...); and two outbound File eWays (Jms...->File1 and Jms...->PayloadFile).



- 3 For clarity, minimize all externals except LogicalHost1 and extFileOut.
- 4 Click the **Auto Map** button. This assigns all components to their servers:
 - ♦ The JmsErrorToFileBP service is assigned to **IntegrationSvr1**
 - ♦ The HostError....->Jms... topic is assigned to **SbJmsIQMgr1**

- ♦ The outbound File eWays are both assigned to **extFileOut**
- 5 Click **Map Variables**. When the Deployment Profile Mappings dialog box appears: In the row for variable **Data Root Directory**, click in the **Mapped Name** column and, from the drop-down list, choose **DataRootDirectory**. Then click **OK**.
- 6 When the profile is complete—that is, when all runnable components in the project are associated with corresponding external servers and the variables are mapped—click  **Save All**, and then click **Activate**. In response to the dialog box, click **No**; that is, do *not* apply to Logical Host(s).

Tip: *If you receive an error message regarding a JNDI name prefix—CodeGenException: Failed to obtain JNDI name prefix(ERROR_GET_JNDINAMEPREFIX)—ensure your variables have been correctly defined and are correctly mapped.*

- 7 For Berlin: Repeat all previous steps, but with the following substitutions: name the deployment profile **DP_BerlinError**, and point it at **Berlin_Env**.

To activate the **JmsPayloadToInternalReceiver** project

- 1 In the Project Explorer tree, right-click **JmsPayloadToInternalReceiver** and, on the popup context menu, point at **New** and click **Deployment Profile**
- 2 Rename it to **DP_AtlantaPayload**, point it at **Atlanta_Env**, and click **OK**.
- 3 For clarity, minimize all externals except **LogicalHost1** and **ExtFileOut**.
- 4 Click the  **Auto Map** button. This assigns all components to their servers:
 - ♦ The **JmsPayloadTo...** service is assigned to **IntegrationSvr1**
 - ♦ The **ToInternalQueue...** queue is assigned to **SbJmsIQMgr1**
 - ♦ The outbound File eWay is assigned to **extFileOut**
- 5 Click **Map Variables**. When the Deployment Profile Mappings dialog box appears: In the row for variable **Data Root Directory**, click in the **Mapped Name** column and, from the drop-down list, choose **DataRootDirectory**. Then click **OK**.
- 6 When the profile is complete—that is, when all runnable components in the project are associated with corresponding external servers and the variables are mapped—click  **Save All**, and then click **Activate**. In response to the dialog box, click **No**.
- 7 For Berlin: Repeat all previous steps, but with the following substitutions: name the deployment profile **DP_BerlinPayload**, and point it at **Berlin_Env**.


To activate the main **AS2_Protocol** project

Tip: *For a preview of the final result before activation, see [Figure 14 on page 36](#).*

- 1 In the Project Explorer tree, right-click **AS2_Protocol** and, on the popup context menu, point at **New** and click **Deployment Profile**
- 2 Keep the default name (**Deployment1**), point it at **Atlanta_Env**, and click **OK**. (The sample assumes that **Deployment1** points at **Atlanta_Env** and **Deployment2** points at **Berlin_Env**. If you vary from these assumptions, you will need to make corresponding changes in ePM.)

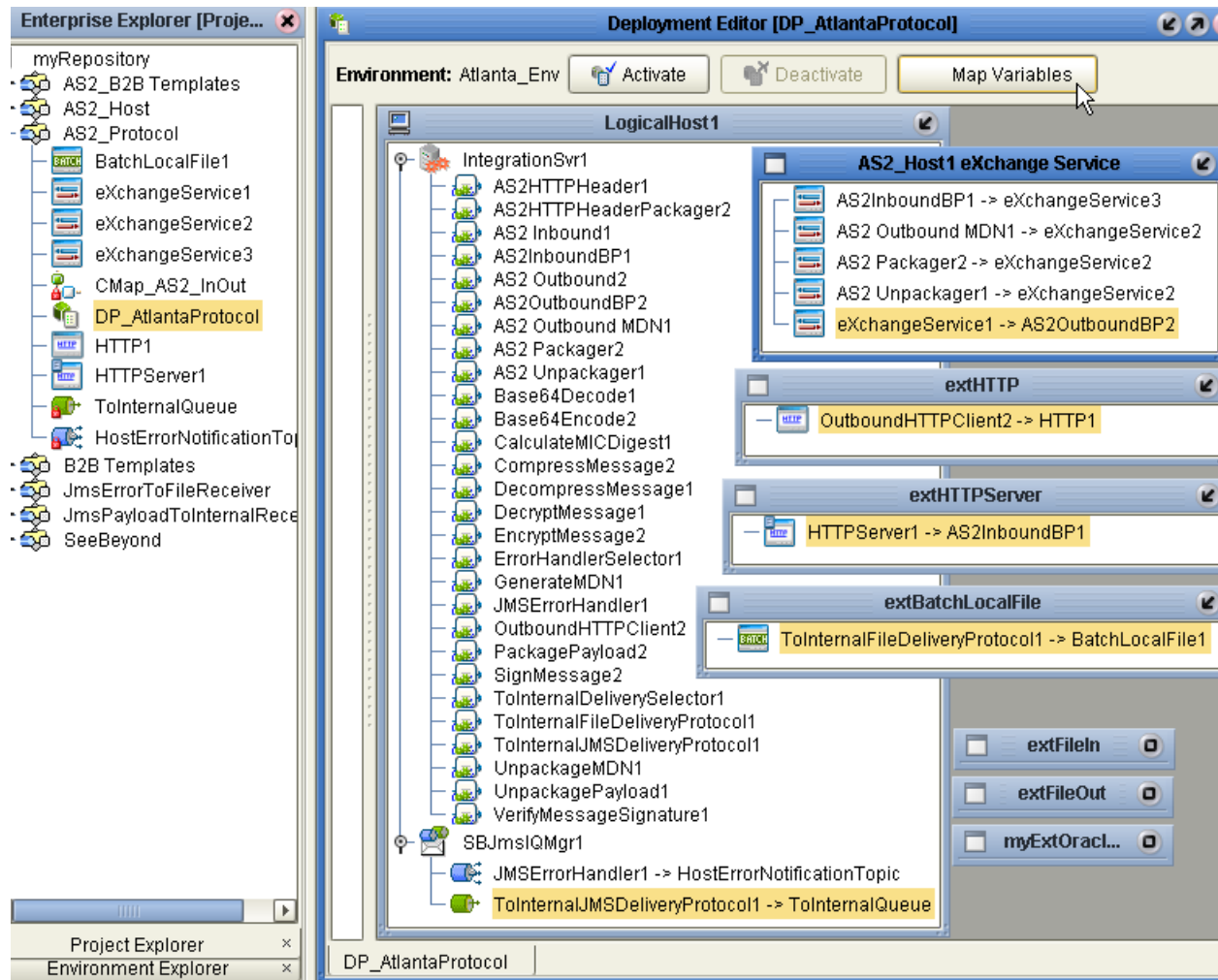
The Deployment Editor's left pane displays: A long list of services; a topic (**JMSErrorHandler1->Host...**) a queue (**ToInternalJMSDeliveryProtocol->...**);




an inbound eWay for HTTPServer; outbound eWays for HTTP and BatchLocalFile; and eWays for one inbound and four outbound eXchange services.

- 3 Click the  **Auto Map** button. This assigns all components to their servers:
 - ◆ The services are assigned to **IntegrationSvr1**
 - ◆ The topic and the queue are assigned to **SbJmsIQMgr1**
 - ◆ The outbound BatchLocalFile eWay is assigned to **extBatchLocalFile**
 - ◆ The inbound HTTPServer eWay is assigned to **extHTTPServer**
 - ◆ The outbound HTTP eWay is assigned to **extHTTP**
 - ◆ The eXchange Service connectors are assigned to **AS2_Host1_eXchange Service**
If necessary, for the service “eXchangeService->AS2OutboundBP2”, manually drag the service into the server and specify **AS2** as the protocol.

Result: See Figure 14.

Figure 14 Deployment Profile for AS2_Protocol



- 4 When the profile is complete—that is, when all runnable components in the project are associated with corresponding external servers—click  **Save All**, and then click **Activate**. In response to the dialog box, click **No**; that is, do *not* apply to Logical Host(s).
 - 5 For Berlin: Repeat all previous steps, but with the following substitutions: name the deployment profile DP_ **Berlin**Protocol, and point it at **Berlin**_Env.
- Tip:** *Successful activation will take some time. If activation fails because of an improper configuration of myExtHTTP, return to the environment, open the properties of the HTTP external, and provide it with dummy values for truststore and password.*
- 6 Click  **Save All**, close all canvases, and click  **Refresh All from Repository**.

Final result: All projects are activated, and will run when the logical hosts are started.

4.3 Design Steps in eXchange Partner Manager

For this sample implementation, design-time steps in eXchange Partner Manager (ePM) consist of the following:

- [Importing the Trading Partners into the Environments on page 38](#)
- [Configuring Trading Partner Parameters for the “Berlin” TP on page 38](#)
- [Activating the Trading Partner on page 43](#)
- [Configuring Trading Partner Parameters for the “Atlanta” TP on page 43](#)
- [Activating the Trading Partner on page 47](#)

Before you begin: Your repository and your eXchange 5.0.5 Oracle database must be running and accessible. Enterprise Designer does *not* need to be running, and you do *not* need to have any logical hosts running.

To start eXchange Partner Manager (ePM)

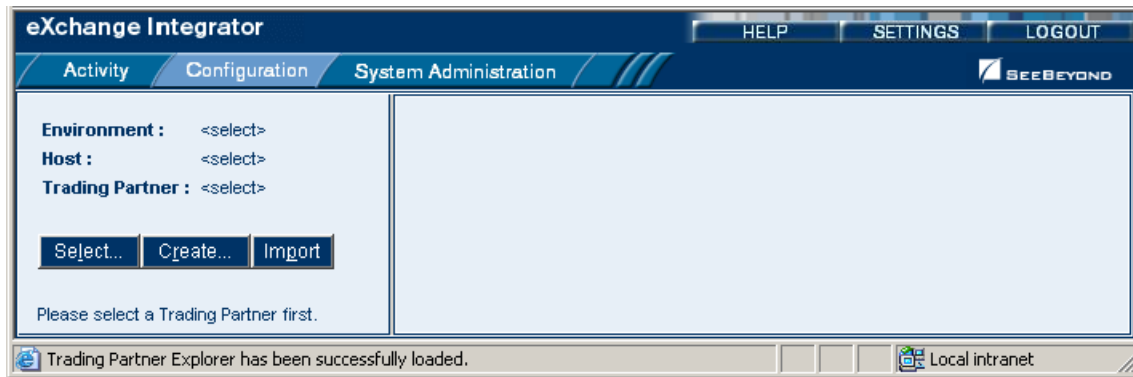
- 1 Start a *new* browser session (that is, do *not* clone a new window of an existing session) pointing it at a repository URL, with **epm** appended. For example:
 - ♦ If your repository were running local on port 12000, the URL would be:
`http://localhost:12000/epm`
 - ♦ For a repository running on machine herMachine on port 33000, it would be:
`http://herMachine:33000/epm`
 - ♦ As usual, IP addresses are also permissible:
`http://10.18.75.85:36271/epm`

The string **epm** is case sensitive. In other words, ePM, Epm, and EPM are all errors.

- 2 When the sign-in screen appears, enter the Enterprise Manager username and password and click **Sign In**.

Result: The status bar (along the lower margin of the window) confirms that Trading Partner Explorer has loaded successfully, and the initial ePM screen appears, with no environment, host, or trading partner. See Figure 15.

Figure 15 Initial ePM Screen



4.3.1 Importing the Trading Partners into the Environments

To import the “Atlanta” trading partner into the Berlin environment

- 1 From the initial ePM screen, in the upper left side, click **Import**.
A new window opens, prompting you to select a B2B host and trading partner.
- 2 Open the B2B Repository and **Berlin_Env** and click **AS2_Host1**.
- 3 For Name, enter **Atlanta**. Browse to C:\temp\exChange\Sample\TradingPartners and open **AS2_TP_for_BerlinEnv.xml**.
- 4 Click the **Import** button and wait for the profile to be modified and imported.

Result: In the explorer tree, under **Berlin_Env**, new trading partner **Atlanta** appears.

To import the “Berlin” trading partner into the Atlanta environment

- 1 In the upper left side of the ePM screen, click **Import**.
A new window opens, prompting you to select a B2B host and trading partner.
- 2 Open the B2B Repository and **Atlanta_Env** and click **AS2_Host1**.
- 3 For Name, enter **Berlin**. Browse to C:\temp\exChange\Sample\TradingPartners and open **AS2_TP_for_AtlantaEnv.xml**.
- 4 Click the **Import** button and wait for the profile to be modified and imported.

Result: In the explorer tree, under **Atlanta_Env**, new trading partner **Berlin** appears.

4.3.2 Configuring Trading Partner Parameters for the “Berlin” TP

When you imported trading partner “Berlin” into the Atlanta environment, parameter settings were valuated in part based on parameters stored in the export file, and in part based on the name of the trading partner. In this section, you will view, set, and/or update the following:

- [TP “Berlin”: Parameters for the Delivery Channel on page 39](#)
- [TP “Berlin”: Parameters for the Internal Delivery Channels on page 41](#)

TP “Berlin”: Parameters for the Delivery Channel

Purpose: To set the parameters governing Atlanta’s message exchange with Berlin.

Keep in mind: You will be configuring a trading partner for the Atlanta environment, and so you will take the viewpoint of the Atlanta B2B host: “ToPartner” means “to Berlin”; “FromPartner” means “from Berlin”.

To configure the delivery channel parameters for trading partner “Berlin”

- 1 In the explorer (lower left) side of the ePM screen, click **Berlin**.

The trading partner’s general properties are displayed

- 2 Click the **Components** tab.

The trading partner’s delivery channel parameters are displayed. See Table 4 .

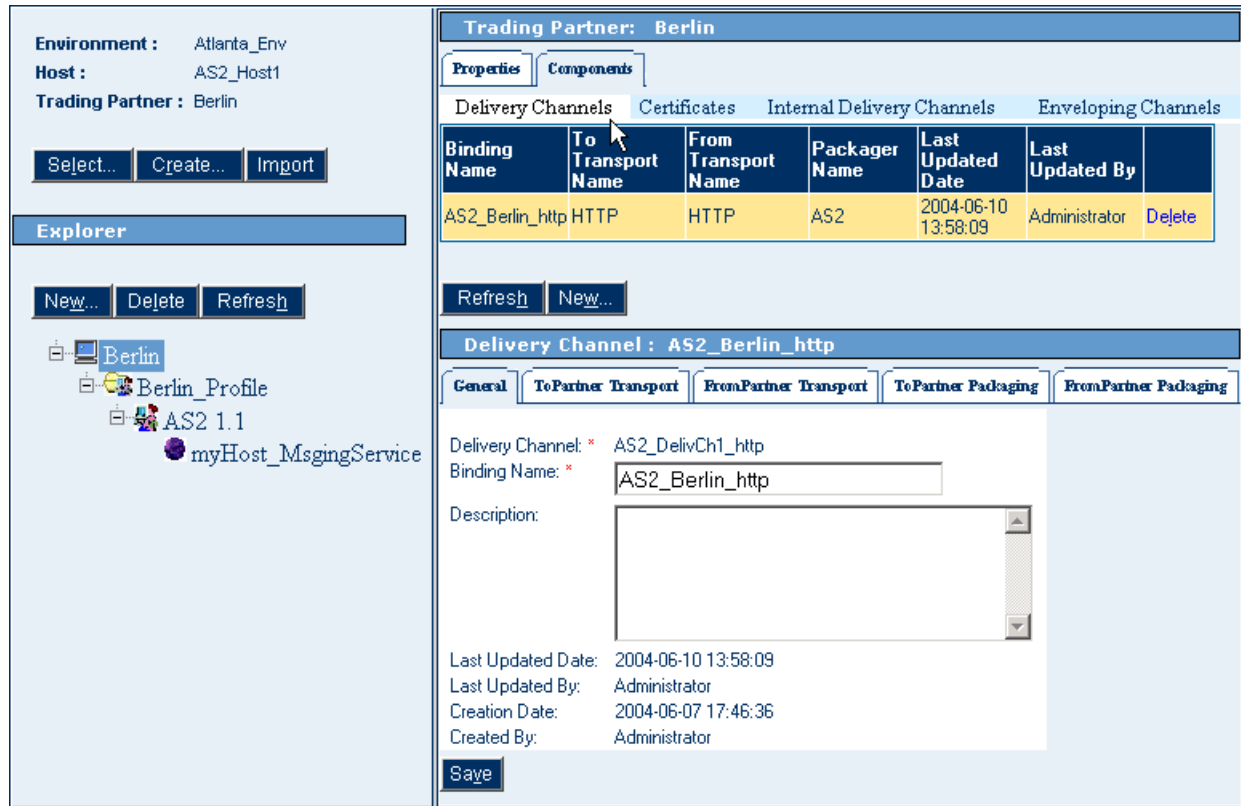
Table 4 Delivery Channel Parameters for Trading Partner “Berlin”

Binding Name	AS2_Berlin_http
ToPartner Transport Name	HTTP
FromPartner Transport Name	HTTP
Packager Name	AS2

- 3 Click the binding name, **AS2_Berlin_http**.

General properties for this binding of the delivery channel (**AS2_DelivCh1_http**) are displayed. See Figure 16.

Figure 16 Delivery Channel for “Berlin”: General Properties for AS2_Berlin_http1



- 4 Click the **ToPartnerTransport** tab and edit the All Purpose End Point so that it contains the correct URL for Berlin. For example:


```
http://localhost:28004/Deployment2_servlet/Inbound
```

 If the logical host running the Berlin host is not named localhost, or its integration server’s web server port is other than 28004, or the deployment profile name is not Deployment2, make appropriate changes. When you are done, click **Save**.
- 5 Click the **FromPartnerTransport** tab and edit the All Purpose End Point so that it contains the correct URL for Berlin. For example:


```
http://localhost:18004/Deployment1_servlet/Inbound
```

 If you have used values other than “localhost” or “18004” or “Deployment1”, make the appropriate changes. When you are done, click **Save**.
- 6 Click the **ToPartnerPackaging** tab and import the “B” encryption key, as follows:
 - A Click **Import** to open the Import Certificate dialog box.
 - B For Certificate Name, enter: **companyb_pubcert**
 - C Click **Browse**, navigate to C:\temp\exchange\Sample\Common\Crypto, and choose the file **CompanyB-Cert.der**
 - D Click **OK** to close the Import Certificate dialog box.

- In the **ToPartnerPackaging** tab, make other changes as needed (the as-shipped values are shown in Table 5 ; be sure the URL for AS2_MDN_DELIVERY_URL matches any changes you may have made), and then click **Save**.

Table 5 ToPartnerPackaging Parameters for Trading Partner “Berlin”

Encryption Key	company b _pubcert
AS2_FROM: *	Atlanta
AS2_TO: *	Berlin
AS2_HOST: *	Atlanta_Host
AS2_VERSION: *	1.1
AS2_HTTP-FROM: *	Atlanta_http
AS2_SIGNATURE_REQ: *	TRUE
AS2_ENCRYPT_REQ: *	TRUE
AS2_MDN_REQ: *	TRUE
AS2_MDN_SIGNATURE_REQ: *	TRUE
AS2_MDN_RESP_TYPE: *	SYNC
AS2_MDN_DELIVERY_URL:	http://localhost:18004/Deployment1_servlet/Inbound
AS2_SUBJECT: *	EDI Message from Atlanta
AS2_COMPRESSED: *	TRUE
AS2_COMPRESSED_BEFORE_SIGNED: *	TRUE
AS2_PAYLOAD_TYPE: *	x12
AS2_MESSAGE_FORMAT: *	SMIME
AS2_ENCODING: *	binary

- Click the **FromPartnerPackaging** tab, make changes as needed (the as-shipped values are shown in Table 6), and then click **Save**.

Table 6 FromPartnerPackaging Parameters for Trading Partner “Berlin”

Signature Certificate	company b _pubcert
AS2_FROM: *	Berlin
AS2_TO: *	Atlanta
AS2_REPORTING_UA: *	Atlanta_CA
AS2_POSITIVE_MDN_DISPOSITION_MSG:	Positive DN msg from Atlanta

Result: For trading partner “Berlin”, the parameters for external delivery channel **AS2_DelivCh1** for the binding named **AS2_Berlin_http** are now set correctly.

TP “Berlin”: Parameters for the Internal Delivery Channels

Purpose: To set the parameters governing Atlanta’s internal message processing when handling messages received from Berlin and preparing messages to be sent to Berlin.

Keep in mind: You will be configuring a trading partner for the Atlanta environment, and so you will take the viewpoint of the Atlanta B2B host: “ToInternal” means “to the

Atlanta internal (having been received from Berlin)”; “FromInternal” means “from the Atlanta internal (and destined for sending to Berlin)”.

Note: Internal Delivery Channel bindings are always required for an AS2 project to run. Although it is possible to activate a trading partner whose messaging actions lack IDCs, the Channel Manager would have no instructions to read any input.

To configure the internal delivery channel parameters for trading partner “Berlin”

- 1 In the explorer (lower left) side of the ePM screen, click **Berlin**.
The trading partner’s general properties are displayed
- 2 Click the **Components** tab.
The trading partner’s delivery channel parameters are displayed.
- 3 Click the **Internal Delivery Channels** tab.
The internal delivery channel parameters are displayed. See Table 7 and Table 8 .

Table 7 Trading Partner “Berlin”, Internal Delivery Channel `idc_SendToInternal`

Binding Name	<code>idc_SendToInternal</code>
Direction	Sender
Transport Name	FILE

Table 8 Trading Partner “Berlin”, Internal Delivery Channel `idc_RecvFromInternal`

Binding Name	<code>idc_RecvFromInternal</code>
Direction	Receiver
Transport Name	ChannelManagerFile

- 4 Click the binding name for the first IDC (`idc_SendToInternal`), click the **Sender Transport** tab, make changes as needed (the as-shipped values are shown in Table 9 ; be sure the directory matches any changes you may have made to the value for project variable **Data Root Directory**), and then click **Save**.

Table 9 Sender Transport Parameters for Trading Partner “Berlin”

FilePattern: *	AS2_Berlin_%d_%H_%m_%s_%S.dat
Directory: *	C:\temp\Exchange\Sample\Data\Atlanta

- 5 Click the binding name for the second IDC (`idc_RecvFromInternal`), click the **Receiver Transport** tab, make changes as needed (the as-shipped values are shown in Table 10 ; be sure the directory matches any changes you may have made to the value for project variable **Data Root Directory**), and then click **Save**.

Table 10 Sender Transport Parameters for Trading Partner “Berlin”

ChannelManagerMode: *	FILE
FilePattern: *	AS2_Berlin*.in
Directory: *	C:\temp\Exchange\Sample\Data\Atlanta
PollMilliseconds: *	5000

Result: For trading partner “Berlin”, the parameters for internal delivery channels **idc_SendToInternal** and **idc_RecvFromInternal** are now set correctly.

4.3.3 Activating the Trading Partner

To activate the “Berlin” trading partner

Purpose: To save all the configuration information to the Oracle database to make it available at run time.

Before you begin: Your eXchange 5.0.5 Oracle database for the corresponding B2B host must be running.

- 1 In the explorer (lower left) side of the ePM screen, click **Berlin**.
- 2 In the bottom lower left of the canvas, click the **Activate** button.
- 3 In response to the confirmation prompt, click **Activate**.

The canvas displays a confirmation: **Trading Partner is successfully activated**.

Result: The **Berlin** trading partner is entirely complete and ready to be run.

4.3.4 Configuring Trading Partner Parameters for the “Atlanta” TP

When you imported trading partner “Atlanta” into the Berlin environment, parameter settings were valuated in part based on parameters stored in the export file, and in part based on the name of the trading partner. In this section, you will view, set, and/or update the following:

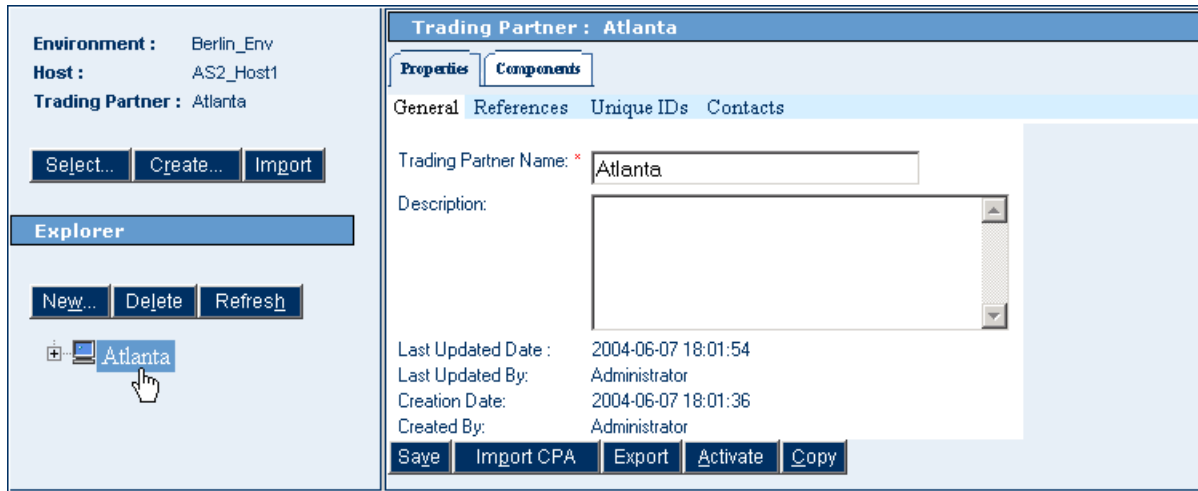
- [TP “Atlanta”: Parameters for the Delivery Channel on page 44](#)
- [TP “Atlanta”: Parameters for the Internal Delivery Channels on page 46](#)

To find the “Atlanta” trading partner (in Berlin_Env)

- 1 In the upper left side of the ePM screen, click **Select**.
A new window opens, prompting you to select a B2B host and trading partner.
- 2 Open the B2B Repository and **Berlin_Env** and click **AS2_Host1**.
- 3 Click **Search**, and then click **OK**.

Result: In the explorer tree, under Berlin_Env, trading partner **Atlanta** reappears. See Figure 17.

Figure 17 Trading Partner “Atlanta” (in Berlin_Env)



TP “Atlanta”: Parameters for the Delivery Channel

Purpose: To set the parameters governing Berlin’s message exchange with Atlanta.

Keep in mind: You will be configuring a trading partner for the Berlin environment, and so you will take the viewpoint of the Berlin B2B host: “ToPartner” means “to Atlanta”; “FromPartner” means “from Atlanta”.

To configure the delivery channel parameters for trading partner “Atlanta”

- 1 In the explorer (lower left) side of the ePM screen, click **Atlanta**. See Figure 17.

The trading partner’s general properties are displayed

- 2 Click the **Components** tab.

The trading partner’s delivery channel parameters are displayed. See Table 11 .

Table 11 Delivery Channel Parameters for Trading Partner “Atlanta”

Binding Name	AS2_Atlanta_http
ToPartner Transport Name	HTTP
FromPartner Transport Name	HTTP
Packager Name	AS2

- 3 Click the binding name, **AS2_Atlanta_http**.

The general properties for this binding of the delivery channel (**AS2_DelivCh1**) are displayed.

- 4 Click the **ToPartnerTransport** tab and edit the All Purpose End Point so that it contains the correct URL for Atlanta. For example:

`http://localhost:18004/Deployment1_servlet/Inbound`

If the logical host running the Atlanta host is not named localhost, or its integration server’s web server port is other than 18004, or the deployment profile name is not Deployment1, make appropriate changes. When you are done, click **Save**.

- 5 Click the **FromPartnerTransport** tab and edit the All Purpose End Point so that it contains the correct URL for Atlanta. For example:


```
http://localhost:28004/Deployment2_servlet/Inbound
```

 If you have used values other than "localhost" or "28004" or "Deployment1", make the appropriate changes. When you are done, click **Save**.
- 6 Click the **ToPartnerPackaging** tab and import the "A" encryption key, as follows:
 - A Click **Import** to open the Import Certificate dialog box.
 - B For Certificate Name, enter: **companya_pubcert**
 - C Click **Browse**, navigate to C:\temp\exchange\Sample\Common\Crypto, and choose the file **CompanyA-Cert.der**
 - D Click OK to close the Import Certificate dialog box.
- 7 In the **ToPartnerPackaging** tab, make other changes as needed (the as-shipped values are shown in Table 12 ; be sure the URL for AS2_MDN_DELIVERY_URL matches any changes you may have made), and then click **Save**.

Table 12 ToPartnerPackaging Parameters for Trading Partner "Atlanta"

Encryption Key	companya_pubcert
AS2_FROM: *	Berlin
AS2_TO: *	Atlanta
AS2_HOST: *	Berlin_Host
AS2_VERSION: *	1.1
AS2_HTTP-FROM: *	Berlin_http
AS2_SIGNATURE_REQ: *	TRUE
AS2_ENCRYPT_REQ: *	TRUE
AS2_MDN_REQ: *	TRUE
AS2_MDN_SIGNATURE_REQ: *	TRUE
AS2_MDN_RESP_TYPE: *	SYNC
AS2_MDN_DELIVERY_URL:	http://localhost:28004/Deployment2_servlet/Inbound
AS2_SUBJECT: *	EDI Message from Berlin
AS2_COMPRESSED: *	TRUE
AS2_COMPRESSED_BEFORE_SIGNED: *	TRUE
AS2_PAYLOAD_TYPE: *	x12
AS2_MESSAGE_FORMAT: *	SMIME
AS2_ENCODING: *	binary

- 8 Click the **FromPartnerPackaging** tab, make changes as needed (the as-shipped values are shown in Table 13), and then click **Save**.

Table 13 FromPartnerPackaging Parameters for Trading Partner "Atlanta"

Signature Certificate	companya_pubcert
-----------------------	------------------

Table 13 FromPartnerPackaging Parameters for Trading Partner “Atlanta” (Continued)

AS2_FROM: *	Atlanta
AS2_TO: *	Berlin
AS2_REPORTING_UA: *	Berlin_CA
AS2_POSITIVE_MDN_DISPOSITION_MSG:	Positive DN msg from Berlin

Result: For trading partner “Atlanta”, the parameters for external delivery channel **AS2_DelivCh1** for the binding named **AS2_Atlanta_http** are now set correctly.

TP “Atlanta”: Parameters for the Internal Delivery Channels

Purpose: To set the parameters governing Berlin’s internal message processing when handling messages received from Atlanta and preparing messages to be sent to Atlanta.

Keep in mind: You will be configuring a trading partner for the Berlin environment, and so you will take the viewpoint of the Berlin B2B host: “ToInternal” means “to the Berlin internal (having been received from Atlanta)”; “FromInternal” means “from the Berlin internal (and destined for sending to Atlanta)”.

Note: Internal Delivery Channel bindings are always required for a project to run. Although it is possible to activate a trading partner whose messaging actions lack IDCs, the Channel Manager would have no instructions to read any input.

To configure the internal delivery channel parameters for trading partner “Atlanta”

- 1 In the explorer (lower left) side of the ePM screen, click **Atlanta**.

The trading partner’s general properties are displayed

- 2 Click the **Components** tab.

The trading partner’s delivery channel parameters are displayed.

- 3 Click the **Internal Delivery Channels** tab.

The internal delivery channel parameters are displayed. See Table 14 and Table 15.

Table 14 Trading Partner “Atlanta”, Internal Delivery Channel `idc_SendToInternal`

Binding Name	<code>idc_SendToInternal</code>
Direction	Sender
Transport Name	FILE

Table 15 Trading Partner “Atlanta”, Internal Delivery Channel `idc_RecvFromInternal`

Binding Name	<code>idc_RecvFromInternal</code>
Direction	Receiver
Transport Name	ChannelManagerFile

- 4 Click the binding name for the first IDC (**idc_SendToInternal**), click the **Sender Transport** tab, make changes as needed (the as-shipped values are shown in Table 16 ; be sure the directory matches any changes you may have made to the value for project variable **Data Root Directory**), and then click **Save**.

Table 16 Sender Transport Parameters for Trading Partner “Atlanta”

FilePattern: *	AS2_Atlanta_%d_%H_%m_%s_%S.dat
Directory: *	C:\temp\Exchange\Sample\Data\Berlin

- 5 Click the binding name for the second IDC (**idc_RecvFromInternal**), click the **Receiver Transport** tab, make changes as needed (the as-shipped values are shown in Table 17 ; be sure the directory matches any changes you may have made to the value for project variable **Data Root Directory**), and then click **Save**.

Table 17 Sender Transport Parameters for Trading Partner “Berlin”

ChannelManagerMode: *	FILE
FilePattern: *	AS2_Atlanta*.in
Directory: *	C:\temp\Exchange\Sample\Data\Berlin
PollMilliseconds: *	5000

Result: For trading partner “Atlanta”, the parameters for internal delivery channels **idc_SendToInternal** and **idc_RecvFromInternal** are now set correctly.

4.3.5 Activating the Trading Partner

To activate the “Atlanta” trading partner

Purpose: To save all the configuration information to the Oracle database to make it available at run time.

Before you begin: Your eXchange 5.0.5 Oracle database for the corresponding B2B host must be running.

- 1 In the explorer (lower left) side of the ePM screen, click **Atlanta**.
- 2 In the bottom lower left of the canvas, click the **Activate** button.
- 3 In response to the confirmation prompt, click **Activate**.

The canvas displays a confirmation: **Trading Partner is successfully activated**.

Result: Both trading partners are entirely complete and ready to be run.

4.4 Runtime Steps

In this section

For the AS2 sample implementation, runtime steps consist of the following:

- [Starting the Logical Hosts on page 48](#)
- [Supplying the Input Data on page 49](#)
- [Using the Message Tracking Facility on page 50](#)

4.4.1 Starting the Logical Hosts

These steps assume you have already installed two or more logical hosts, and that you have already replaced the additional `*_policy.jar` files as noted in [“Additional Policy JAR Files Required to Run SME” on page 24](#). The directory for the logical host that will run Atlanta is called `<AtlantaLH>`, and the directory for the logical host that will run Berlin is called `<BerlinLH>`.

To bootstrap the logical host

- 1 Open a command prompt and change directories to the location of your logical host’s bootstrap executables. For example:

```
cd <AtlantaLH>\bootstrap\bin
```

- 2 Start the bootstrap script using appropriate parameters. For example:

```
bootstrap -r http://myBox:12345/myRepository -i myId -p myPassword  
-e Atlanta_Env -l LogicalHost1 -f
```

- For the `-r` (repository) parameter), supply the correct URL with repository name.
- For the `-i` and `-p` (ID and password) parameters, supply the appropriate values.
- For `-e` (environment) parameter, use: **Atlanta_Env**
- For `-l` (logical host name) parameters, use: **LogicalHost1**
- You need not supply the `-f` (force) flag if all configuration changes have already been applied, or if this is the first time that bootstrap has been run.

Result: After a time, the logical host starts running, and all activated projects that reference `Atlanta_Env` are automatically applied to it.

- 3 Repeat steps 1 and 2 on a different logical host, referencing the same repository but pointing it at the **Berlin_Env** environment. For example:

```
cd <BerlinLH>\bootstrap\bin  
bootstrap -r -f[...] -e Berlin_Env -l LogicalHost1
```

Optional: To apply environment changes when a logical host is running

Purpose: If changes are made to parameters in an environment component while a logical host is running—for example, adding or modifying keystores for an eXchange service—use these steps to apply the changes without having to shutdown and re-

bootstrap the logical host. This is the equivalent of shutting down the logical host and re-bootstrapping it using the **-f** flag.

- 1 In Enterprise Designer, in Environment Explorer, open the environment where the changes have occurred.
- 2 Right-click the logical host that is running and, on the popup menu, click **Apply**
- 3 Repeat the previous step as needed for other logical hosts in the same environment.

Result: In the back end, a “mini-shutdown/mini-rebootstrap” occurs, and the changes are applied to the running logical host.

4.4.2. Supplying the Input Data

Before you begin: Your logical hosts must both be running and the eXchange 5.0.5 Oracle database must be accessible.

The scenarios are set up so that the project variable **Data Root Directory** points at C:\temp\exchange\Sample\Data\Atlanta\ for the Atlanta host, but points at C:\temp\exchange\Sample\Data\Berlin\ for the Berlin host. What this means is:

- The Atlanta project reads files of the form ... \Sample\Data\Atlanta*.in
Output files resulting from successful processing have the form ... \Sample\Data\Berlin\AS2_Atlanta_<timestamp>.dat
Error messages are written to ... \Sample\Data\Atlanta\ErrorTopic_%.dat
Error payload is written to ... \Sample\Data\Atlanta\ErrorTopic_payload_%.dat
Internal messages are received at ... \Sample\Data\Atlanta\toInternal_%.dat
- The Berlin project reads files of the form ... \Sample\Data\Berlin*.in
Output files resulting from successful processing have the form ... \Sample\Data\Atlanta\AS2_Berlin_<timestamp>.dat
Error messages are written to ... \Sample\Data\Berlin\ErrorTopic_%.dat
Error payload is written to ... \Sample\Data\Berlin\ErrorTopic_payload_%.dat
Internal messages are received at ... \Sample\Data\Berlin\toInternal_%.dat

To feed input data to the Atlanta project

- 1 In C:\temp\exchange\Sample\Data\Atlanta\, rename files AS2_Berlin*.~in to AS2_Berlin*.in
- 2 Watch the file extensions change from .in to .~in as the files are picked up.

To feed input data to the Berlin project

- 1 In C:\temp\exchange\Sample\Data\Berlin\, rename files AS2_Atlanta*.~in to AS2_Atlanta*.in
- 2 Watch the file extensions change from .in to .~in as the files are picked up.

4.4.3. Using the Message Tracking Facility

Before you begin: Your logical hosts must both be running and the eXchange 5.0.5 Oracle database must be accessible.

To access Message Tracking

- 1 Open a *new* browser session. (In other words, do *not* clone a new window from an existing session.)
- 2 Point your browser at the URL for message and package tracking. This takes the following form:

http://<loghostname>:<port>/<AppName>/msgTrack/EnterPkgTrack.do

where:

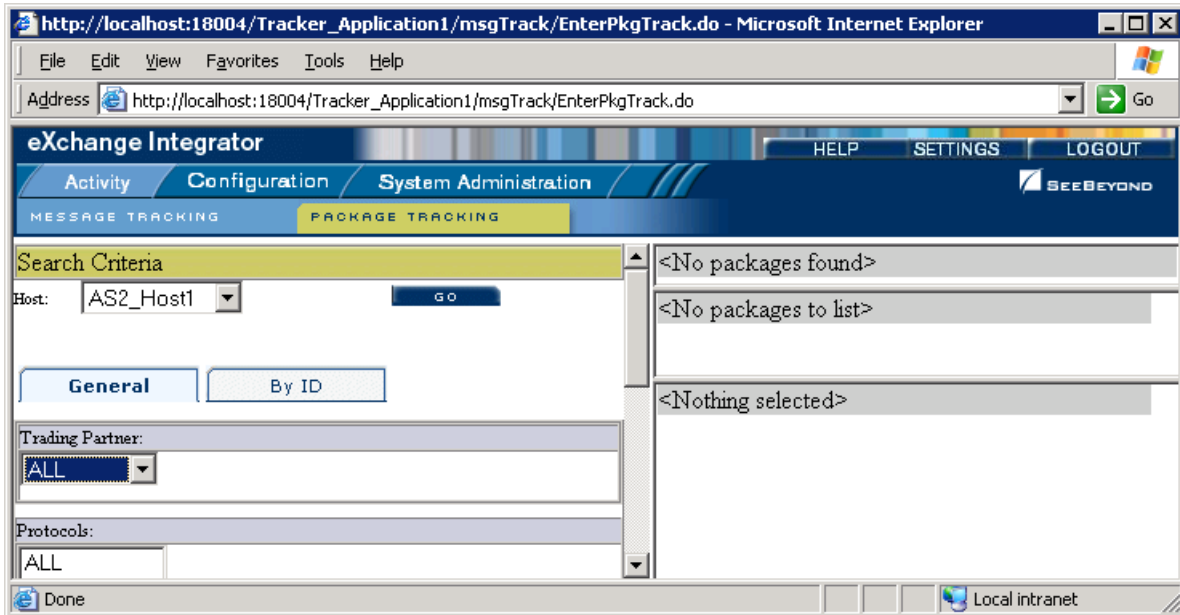
- ♦ <loghostname> is the hostname of the physical machine where you ran bootstrap.exe and pointed it at a particular environment and logical host.
- ♦ <port> is the port number assigned in that environment and logical host (in the configuration for Integration Server .> ... > Web server).
- ♦ <AppName> is the name of the Message Tracking application as it appears in the connectivity map. For example, if there is only one such instance, the default name is **Tracker_Application1**

Examples:

- To access message tracking for Atlanta (IS ports 18000–18009, web port=18004), for a logical host bootstrapped on a physical machine whose hostname is “Dixie”:
http://Dixie:**18004**/Tracker_Application1/msgTrack/EnterPkgTrack.do
- To access message tracking for Berlin (IS ports 19000–19009, web port=19004), if the logical host was bootstrapped locally:
http://localhost:**19004**/Tracker_Application1/msgTrack/EnterPkgTrack.do
- To access message tracking for an Integration Server that was customized to use a special SSL-only web port=21444, if the logical host was bootstrapped on a physical machine whose IP address is 10.18.5.5:
http://10.18.5.5:**21444**/Tracker_Application1/msgTrack/EnterPkgTrack.do

Result: See Figure 18.

Figure 18 Message Tracking, on Startup



To search by B2B host, trading partner, and protocol

- 1 Under Search Criteria, use the Host drop-down list to choose the B2B host whose messages you want to examine (in this case, **AS2_Host1**), and click **GO**.
- 2 Under Trading Partner, either click **ALL** or choose a particular trading partner (such as **Berlin**) from the drop-down list.
- 3 Under Protocols, either click **ALL** or choose **AS2** from the drop-down list.
- 4 At the lower left of the window, click **SEARCH**.

Result: The canvas (right side), under Search Results, displays a page containing the Package IDs of the latest ten tracked messages fitting the criteria you specified. Navigation links (Previous, Next, and Go to Page) allow you to see other pages of ten results each. See Figure 19.

Figure 19 Message Tracking, Showing Initial Search Results

The screenshot shows the 'MESSAGE TRACKING' application window. The 'Search Criteria' pane on the left includes a 'Host' dropdown set to 'AS2_Host1', a 'Trading Partner' dropdown set to 'Berlin', and a 'Protocols' list with 'AS2' selected. The 'Filters' section at the bottom left has 'Error Type' set to 'ALL', 'Direction' set to 'Both', and 'Date Filter' set to 'Process date'. The 'SEARCH' button is visible at the bottom left of the criteria pane.

The 'Search Results' pane on the right displays a table of message tracking data. The table has the following columns: Package ID, Trading Partner, Protocol, Package Conversation Type, ID, Direction, Error Process Data, Date, Response Req, and Ack Date. The results show ten entries with various package IDs, trading partners (Berlin and Atlanta), and directions (Inbound and Outbound).

Package ID	Trading Partner	Protocol	Package Conversation Type	ID	Direction	Error Process Data	Date	Response Req	Ack Date
<20040605004911969@Berlin_Host>	Berlin	AS2			Inbound	No	06/05/2004 00:48:35	No	
<20040605004907723@Berlin_Host>	Berlin	AS2			Inbound	No	06/05/2004 00:48:33	No	
<20040605004822359@Atlanta_Host>	Berlin	AS2			Outbound	No	06/05/2004 00:48:24	Yes	06/05/2004 00:48:36
<20040605004822437@Atlanta_Host>	Berlin	AS2			Outbound	No	06/05/2004 00:48:23	Yes	06/05/2004 00:48:33
<20040605004326232@Berlin_Host>	Berlin	AS2			Inbound	No	06/05/2004 00:43:09	No	
<20040605004119906@Atlanta_Host>	Berlin	AS2			Outbound	No	06/05/2004 00:41:35	Yes	06/05/2004 00:43:20
<20040604191043487@Berlin_Host>	Berlin	AS2			Inbound	No	06/04/2004 19:09:35	No	
<20040604190931328@Atlanta_Host>	Berlin	AS2			Outbound	No	06/04/2004 19:09:31	Yes	
<20040604183039911@Berlin_Host>	Berlin	AS2			Inbound	No	06/04/2004 18:29:32	No	
<20040604182911140@Atlanta_Host>	Berlin	AS2			Outbound	No	06/04/2004 18:29:17	Yes	06/04/2004 18:29:35

To filter results by error type, direction, and/or date

Purpose: After performing a search, or after setting up a search using the previous procedure, you can specify one or more further criteria.

- 1 Near the bottom of the left pane, under **Filters**, specify one or more of the following:
 - ♦ For **Error Type**: If you do not choose ALL, you can restrict your search either to display error messages only, or to display non-error messages only.
 - ♦ For **Direction**: If you do not choose ALL, you can restrict your search either to display inbound messages only, or to display outbound messages only.
 - ♦ For **Date Filter**: You can choose to include only those messages whose *processing* date lies within a range you specify, or only those messages whose *acknowledgment* date lies within the range.
- 2 At the lower left of the window, click **SEARCH**.

Result: The canvas displays a page containing the Package IDs of the latest ten tracked messages fitting the criteria you specified. See Figure 20.

Figure 20 Message Tracking, Showing Filtered Results

Search Criteria

Host: AS2_Host1

General | By ID

Trading Partner: Berlin

Protocols: ALL, ebxml, AS2, EmptyMAD

Filters:
 Error Type: Error
 Direction: Inbound
 Date Filter: Acknowledgement date
 From and To: Process date, Acknowledgement date

Search Results

Print << Previous Page 1 of 1 Next >> Go to Page: 1

Package ID	Trading Partner	Protocol Type	Conversation ID	Direction	Error Data	Process Date	Response Req	Ack Date
<20040603155546140@Berlin_Host>	Berlin	AS2		Inbound	Yes	06/03/2004 15:56:52	No	06/03/2004 15:57:13
<20040603155546125@Berlin_Host>	Berlin	AS2		Inbound	Yes	06/03/2004 15:56:53	No	06/03/2004 15:57:13
<20040603161205390@Berlin_Host>	Berlin	AS2		Inbound	Yes	06/03/2004 16:12:07	No	06/03/2004 16:12:08
<20040603181209546@Berlin_Host>	Berlin	AS2		Inbound	Yes	06/03/2004 18:12:11	No	06/03/2004 18:12:11

<Nothing selected>

To obtain details of a specified package

Purpose: On a package-by-package basis, you can examine the text of either the original message or the acknowledgment.

- 1 After obtaining results from a search using any of the procedures mentioned earlier, click the package ID for any of the returned results.
- 2 In the “Details for package <package-ID>” pane, click **Open** to see the contents (possibly encrypted) of either the original message or the acknowledgment.

Result: See Figure 21. You can use cut, copy, and paste on any text in the window.

Figure 21 Message Tracking, Showing Package Details and Message Content

The screenshot shows a web-based message tracking interface. At the top, there is a 'Search Results' header with navigation options like 'Print', '<< Previous', 'Page 1 of 1', 'Next >>', and 'Go to Page: 1'. Below this is a table with columns: Package ID, Trading Partner, Protocol, Package Type, Conversation ID, Direction, Error Data, Process Date, and Response Req Ack Date. A single row is visible with the Package ID '<20040611165132643@Berlin_Host>' circled in green. A green arrow points from this ID to the text 'Step 1: Select Package ID.' Below the table, the details for the selected package are shown: 'Details for package: <20040611165132643@Berlin_Host>'. There are four tabs: 'Message', 'Attributes', 'Errors', and 'Dialog'. The 'Message' tab is selected. Below the tabs, it says 'Following are the messages available to the selected item:' and 'Original message:'. A green arrow points from the text 'Step 2: Click Open.' to an 'OPEN' button. To the left of the message content window, a green note says 'Result is displayed in a new window.' The 'Message View' window shows the message content, including headers like 'Message for Transaction ID: <20040611165132643@Berlin_Host>' and 'Content-Type: multipart/signed; protocol="application/pkcs7-sig boundary="-----_Part_0_255376287.1086998019233"', followed by EDI data.

Package ID	Trading Partner	Protocol	Package Type	Conversation ID	Direction	Error Data	Process Date	Response Req Ack Date
<20040611165132643@Berlin_Host>	Berlin_hliuxp	AS2			Outbound No		06/11/2004 17:01:31	Yes

Step 1: Select Package ID.

Details for package: <20040611165132643@Berlin_Host>

Message | Attributes | Errors | Dialog

Following are the messages available to the selected item:

Original message:

Step 2: Click Open.

Result is displayed in a new window.

Message View - Microsoft Internet Explorer provided by SeeBeyond

Message for Transaction ID:
<20040611165132643@Berlin_Host>

Content-Type: multipart/signed; protocol="application/pkcs7-sig boundary="-----_Part_0_255376287.1086998019233"

-----_Part_0_255376287.1086998019233

Content-Type: Application/EDI-X12

ISA*00*TEST060804*00*TEST060804*ZZ*Atlanta *ZZ*Berlin
*961007*2013*U*00200*000000001*0*T*~GS*PO*S1S1S1S1S1S1S1S1S1S1R1R
Number~PER*AA*Hans Gutten*CP*1.322.323.4444*****rgg4egv4t4~TA
aaa*1000000*AS*90.00*BD*AK*234235v3534q6f3534v4353453vq3q32***
aaa*1000000*AS*90.00*BD*AK*234235v3534q6f3534v4353453vq3q32***
aaa*1000000*AS*90.00*BD*AK*234235v3534q6f3534v4353453vq3q32***
aaa*1000000*AS*90.00*BD*AK*234235v3534q6f3534v4353453vq3q32***

----- Part 0 255376287.1086998019233

Glossary of Acronyms

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.

BAD

In eXchange, *Business Attribute Definitions (BADs)* define the metadata attributes of parameters used in business protocols such as X12, HIPAA, or EDIFACT.

B2B

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

DCP

In eXchange, a *Delivery Channel Profile (DCP)* is an association between a particular messaging service and a particular transport attributes definition (TAD). Also see “IDC” and “XDC”.

EAD

In eXchange, *Enveloping Attribute Definitions (EADs)* define the metadata attributes of parameters used in enveloping protocols such as X12, HIPAA, or EDIFACT.

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

See Beyond's *Integrated Composite Application Network (ICAN) Suite* includes eGate Integrator, eXchange Integrator, various eWay Intelligent Adapters, OTD Libraries, and Protocol Manager Composite Applications, as well as many other products.

IDC

In eXchange, an *Internal Delivery Channel (IDC)* is an association between a particular transport attributes definition (TAD) and a direction (either Sender or Receiver). Compare with "XDC".

MAD

In eXchange, *Messaging Attribute Definitions (MADs)* define the metadata attributes of parameters used in messaging protocols such as AS2 or ebXML.

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 ICAN, 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.

SME

In ICAN, *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.

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.

XDC

In eXchange, an *External Delivery Channel* (XDC) is an association between three items:

- (1) either a messaging service (passthrough) or a business service (dialog);
- (2) a transport attributes definition (TAD) for the ToPartner (Sender) direction; and
- (3) a TAD for the FromPartner (Receiver) direction.

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

- AS2
 - (defined) 12
 - compared to AS1 11
 - specification for 11, 14
- AS2 message structure 12
 - (illustrated) 13
- AS2 sample implementation 25–54

C

- CodeGenException (JNDI name prefix) 35
- configuring
 - HTTPS 28
 - Oracle eWay 28, 30
 - ports 28, 29, 40, 44
- conventions, document 9

D

- database for eInsight engine
 - supported levels 19
- database for eXchange
 - supported levels 19
- document conventions 9

E

- EDI-INT Requirements document 14
- EDI-INT, defined 12
- eInsight engine
 - supported databases 19
- encryption .jar files
 - updating 24
 - where to download 24
- error "Failed to obtain JNDI name prefix" 35
- eXchange database
 - system requirements 19
- eXchange support
 - for Oracle databases 19
 - for platforms 18

H

- HTTPS (HTTP on SSL), configuring 28

I

- IETF (URLs for) 14

J

- JNDI name prefix (error message) 35

L

- local_policy.jar, updating 24

M

- message structure, AS2 12

O

- operating systems supported by eXchange 18
- Oracle database for eXchange
 - supported levels 19
- Oracle eWay, configuring 28, 30

P

- platforms supported by eXchange 18
- policy .jar files
 - updating 24
 - where to download 24
- ports, configuring 28, 29, 40, 44

S

- sample implementation 25–54
- Screenshots 9
- Secure Messaging Extension (SME)
 - updating policy .jar files for 24
- SSL (Secure Sockets Layer), configuring 28
- structure of an AS2 message 12

U

- URLs
 - for IETF 14
 - for SeeBeyond 10
- US_export_policy.jar
 - updating 24
 - where to download 24