

***SeeBeyond ICAN Suite***

# **AS2 Manager Composite Application User's Guide**

***Release 5.0.4***



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 This work is protected as an unpublished work under the copyright laws.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 20040611194157.

# Contents

List of Figures	5
-----------------	---

List of Tables	6
----------------	---

---

## Chapter 1

<b>Introduction</b>	<b>7</b>
Overview	7
Contents of This Guide	8
Writing Conventions	8
Additional Conventions	8
Supporting Documents	9
Online Documents	9
The SeeBeyond Web Site	9

---

## Chapter 2

<b>Overview</b>	<b>10</b>
<b>About eXchange Integrator</b>	<b>10</b>
eXchange and the ICAN Suite	11
eXchange Architectural Overview	11
Process Overview	12
<b>About AS2</b>	<b>13</b>
General History and Definitions	14
Definition of EDI-INT	14
Definition of AS1	14
Definition of AS2	14
AS2 Message Format	14
Packaging Layer References	15
Additional References	16

---

Chapter 3

<b>Installing AS2 Manager</b>	<b>17</b>
<b>System Requirements</b>	<b>17</b>
Platform Support	17
Database Support	17
Database for eXchange Partner Management and Message Tracking	17
<b>Installation Steps</b>	<b>18</b>
Uploading AS2 Manager to the Repository	18
Refreshing Enterprise Designer with AS2 Manager	20
<b>Database Scripts</b>	<b>21</b>
<b>Additional Policy JAR Files Required to Run SME</b>	<b>22</b>

---

Chapter 4

<b>Implementation Scenario</b>	<b>23</b>
Overview of the Sample Implementation	23
<b>Initial Setup Steps</b>	<b>24</b>
Installing the Sample Files for AS2	24
Importing the Sample Projects	25
<b>Design Steps in Enterprise Designer</b>	<b>26</b>
Setting Up the Sample Environments	26
Viewing and Activating the B2B Host Project	29
Configuring the eXchange Services with Crypto Information	31
Creating and Activating the Project Deployment Profiles	32
<b>Design Steps in ePM</b>	<b>35</b>
Importing the Trading Partners into the Environments	36
Configuring Trading Partner Parameters for the “Berlin” TP	36
TP “Berlin”: Parameters for the Delivery Channel	36
TP “Berlin”: Parameters for the Internal Delivery Channels	39
Activating the Trading Partner	40
Configuring Trading Partner Parameters for the “Atlanta” TP	40
TP “Atlanta”: Parameters for the Delivery Channel	41
TP “Atlanta”: Parameters for the Internal Delivery Channels	43
Activating the Trading Partner	44
<b>Runtime Steps</b>	<b>44</b>
Starting the Logical Hosts	45
Supplying the Input Data	46
Using the Message Tracking Facility	46

<b>Index</b>	<b>51</b>
--------------	-----------

# List of Figures

Figure 1	eXchange and the ICAN Suite	11
Figure 2	eXchange Architecture	13
Figure 3	AS2 Message Structure	15
Figure 4	Enterprise Manager ADMIN page	19
Figure 5	Update Center Wizard: Select Modules to Install	20
Figure 6	Update Center Wizard: Progress Bars	21
Figure 7	Update Center Wizard: Restart Enterprise Designer	21
Figure 8	Project Explorer Tree, After Importing All Sample Projects for AS2	26
Figure 9	Sample Environment, Before Configuration	28
Figure 10	Deployment Profile for B2B Host, Before Activation	30
Figure 11	Configuring Private Keys and Trust Stores	31
Figure 12	Deployment Profile for Error Processing	32
Figure 13	Deployment Profile for AS2_Protocol	34
Figure 14	Initial ePM Screen	35
Figure 15	Delivery Channel for “Berlin”: General Properties for AS2_Berlin_http1	37
Figure 16	Trading Partner “Atlanta” (in Berlin_Env)	41
Figure 17	Message Tracking, on Startup	47
Figure 18	Message Tracking, Showing Initial Search Results	48
Figure 19	Message Tracking, Showing Filtered Results	49
Figure 20	Message Tracking, Showing Package Details and Message Content	50

# List of Tables

Table 1	Writing Conventions	8
Table 2	JRE Versions Listed by Operating System	22
Table 3	B2B Protocol Processes Used in the AS2 Sample Implementation	23
Table 4	Delivery Channel Parameters for Trading Partner “Berlin”	37
Table 5	ToPartnerPackaging Parameters for Trading Partner “Berlin”	38
Table 6	FromPartnerPackaging Parameters for Trading Partner “Berlin”	38
Table 7	Trading Partner “Berlin”, Internal Delivery Channel idc_SendToInternal	39
Table 8	Trading Partner “Berlin”, Internal Delivery Channel idc_RecvFromInternal	39
Table 9	Sender Transport Parameters for Trading Partner “Berlin”	39
Table 10	Sender Transport Parameters for Trading Partner “Berlin”	40
Table 11	Delivery Channel Parameters for Trading Partner “Atlanta”	41
Table 12	ToPartnerPackaging Parameters for Trading Partner “Atlanta”	42
Table 13	FromPartnerPackaging Parameters for Trading Partner “Atlanta”	42
Table 14	Trading Partner “Atlanta”, Internal Delivery Channel idc_SendToInternal	43
Table 15	Trading Partner “Atlanta”, Internal Delivery Channel idc_RecvFromInternal	43
Table 16	Sender Transport Parameters for Trading Partner “Atlanta”	43
Table 17	Sender Transport Parameters for Trading Partner “Berlin”	44

# 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.

### In this chapter

- [Overview](#) on page 7
- [Contents of This Guide](#) on page 8
- [Writing Conventions](#) on page 8
- [Supporting Documents](#) on page 9
- [Online Documents](#) on page 9
- [The SeeBeyond Web Site](#) on page 9

---

## 1.1 Overview

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 of AS2 Manager.
- Understand the 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 editors and how to use them in your environment.

## 1.2 Contents of This Guide

This guide is arranged as follows:

- **Chapter 1, “Introduction”** provides an overview of this document’s purpose, contents, writing conventions, and supported documents.
- **Chapter 2, “Overview”** discusses general features and architecture of eXchange.
- **Chapter 3, “Installing AS2 Manager”** provides step-by-step instructions for installing the eXchange product 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.

## 1.3 Writing Conventions

The following writing conventions are observed throughout this document.

**Table 1** Writing Conventions

Text	Convention	Example
Names of buttons, files, icons, parameters, variables, methods, menus, and objects	<b>Bold</b> text	<ul style="list-style-type: none"> <li>▪ Click <b>OK</b> to save and close.</li> <li>▪ From the <b>File</b> menu, select <b>Exit</b>.</li> <li>▪ Select the <b>logicalhost.exe</b> file.</li> <li>▪ Enter the <b>timeout</b> value.</li> <li>▪ Use the <b>getClassName()</b> method.</li> <li>▪ Configure the <b>Inbound</b> File eWay.</li> </ul>
Command-line arguments, code samples	Fixed font. Variables are shown in <i><b>bold italic</b></i> .	<code>bootstrap -p <i><b>password</b></i></code>
Hypertext links	<b>Blue</b> text	For more information, see <a href="#">“Writing Conventions” on page 8.</a>

### Additional Conventions

#### Windows Systems

For the purposes of this guide, references to “Windows” will apply to Microsoft Windows Server 2003, Windows XP, and Windows 2000.

#### Path Name Separator

This guide uses the backslash (“\”) as the separator within path names. If you are working on a UNIX system, please make the appropriate substitutions.



---

## 1.4 Supporting Documents

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

<b>Title</b>	<b>Filename</b>
<i>SeeBeyond ICAN Site Installation Guide</i>	ICAN_Install_Guide.pdf
<i>SeeBeyond ICAN Suite Primer</i>	Primer.pdf
<i>SeeBeyond ICAN Suite Deployment Guide</i>	Deployment_Guide.pdf
<i>eGate Integrator Release Notes</i>	eGate_Release_Notes.pdf
<i>eGate Integrator Tutorial</i>	eGate_Tutorial.pdf
<i>eGate Integrator User's Guide</i>	eGate_User_Guide.pdf
<i>eGate Integrator System Administration Guide</i>	Sys_Admin_Guide.pdf
<i>eGate Integrator JMS Reference Guide</i>	eGate_JMS_Reference.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>eGate Integrator User's Guide</i>	eXchange_User_Guide.pdf
<i>ebXML Manager Composite Application User's Guide</i>	ebXML_Manager_Guide.pdf
<i>X12 OTD Library User's Guide</i>	X12_OTD_Library.pdf
<i>HIPAA OTD Library User's Guide</i>	HIPAA_OTD_Library.pdf
<i>UN/EDIFACT OTD Library User's Guide</i>	EDIFACT_OTD_Library.pdf
Readme for ICAN 5.0.4	Readme.txt

Also see the URLs provided in **"Additional References"** on page 16.

---

## 1.5 Online Documents

The documentation for the SeeBeyond ICAN Suite is distributed as a collection of online documents. These documents are viewable with the Acrobat Reader application from Adobe Systems. Acrobat Reader can be downloaded from:

<http://www.adobe.com>

---

## 1.6 The 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>

# Overview

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

## In this chapter

- [About eXchange Integrator](#) on page 10
- [About AS2](#) on page 13

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

---

## 2.1 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 business protocol pipelines, it also provides the tools and framework to create and adopt new protocols and to build custom pipelines.

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 business pipeline, 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 partners can be configured by importing Collaboration Protocol Agreements (CPAs); or trading partner profiles can be 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.

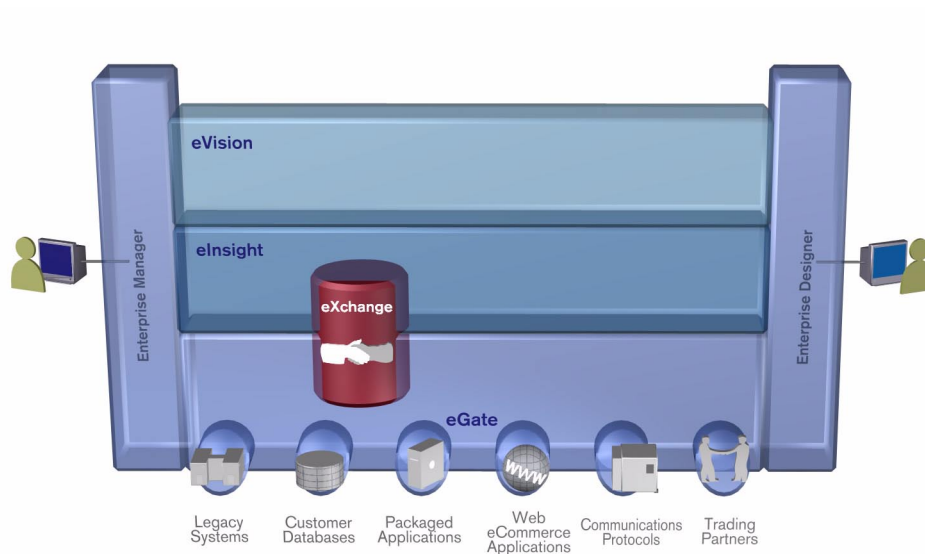
At run time, all steps in the business process, from initial receipt of the message to final delivery to the trading partner, are tracked in real time and also stored in the eXchange database. The Web-based message/package tracker provides tools for retrieving and filtering tracked message and envelope 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.

### 2.1.1. 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 1 illustrates how eXchange and other ICAN Suite components work together.

**Figure 1** eXchange and the ICAN Suite



### 2.1.2. 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.

### 2.1.3. Process Overview

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

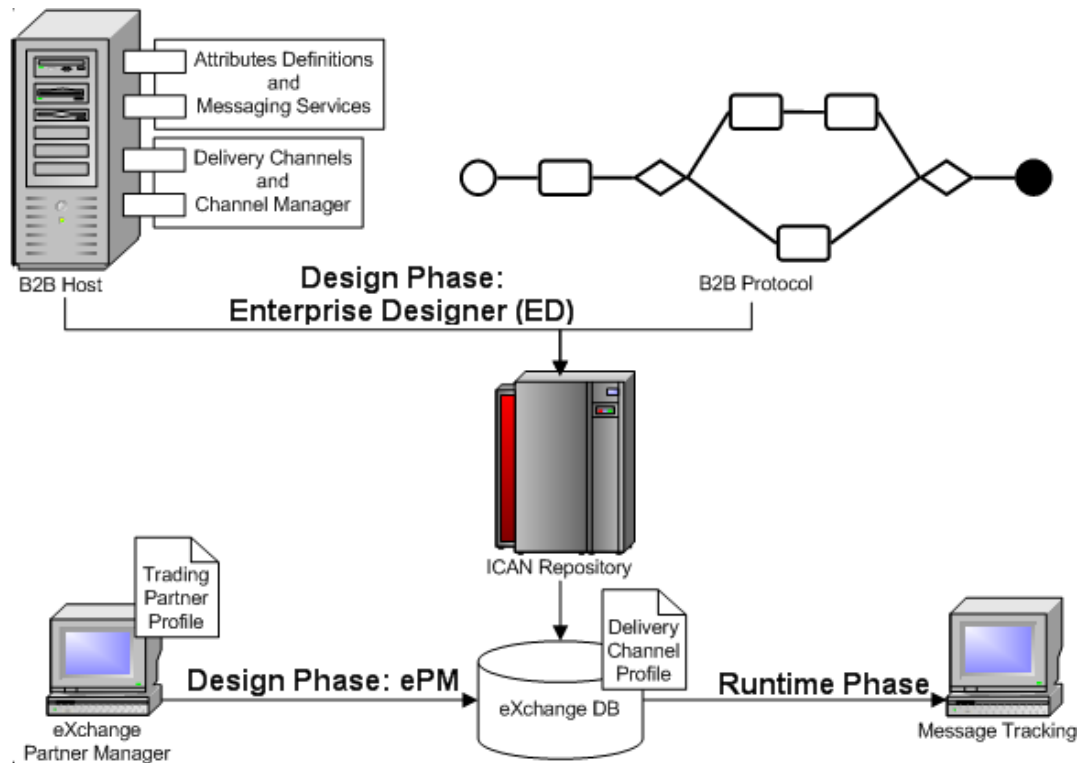
- Design phase within Enterprise Designer
- Design phase within eXchange Partner Manager
- Runtime phase

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 2.

**Figure 2** eXchange Architecture



## 2.2 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.

## 2.2.1 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.

## 2.2.2 AS2 Message Format

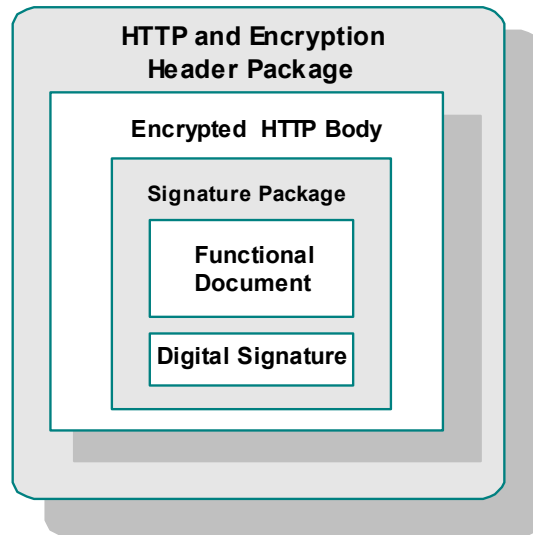
An AS2 message conforms with the following structure:

- 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 3 shows a diagram of the basic message format.

**Figure 3** AS2 Message Structure



### 2.2.3. 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>

## 2.2.4. 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/internet-drafts/draft-ietf-ediint-as2-15.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.



## Chapter 3

# Installing AS2 Manager

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

---

## 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. Platform Support

eXchange supports the following operating systems:

- Microsoft Windows Server 2003, Windows XP SP1a, and Windows 2000 SP3 or SP4
- Sun Solaris 8 and 9, with required patches
- HP Tru64 V5.1A with required patches
- HP-UX 11.0 and 11i (PA-RISC), with required patches and parameter changes
- IBM AIX 5.1L and 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 provides a run-time persistent store for trading partner management and message tracking. eXchange supports the following databases:

- Oracle 8.1.7
- Oracle 9.01
- Oracle 9.2

## 3.2 Installation Steps

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).

### 3.2.1. 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**) 5.0.4
  - ♦ Batch eWay adapter (**BatcheWay.sar**) 5.0.4
  - ♦ Oracle eWay adapter (**OracleeWay.sar**) 5.0.4
  - ♦ File eWay adapter (**FileeWay.sar**) — This not an installation requirement, but it is required by the sample implementation
  - ♦ eXchange Integrator (**eXchange.sar**) 5.0.4
  - ♦ HTTP(S) eWay adapter (**HTTPeWay.sar**) 5.0.4
  - ♦ Secure Message Extension (**SMEWebServices.sar**) 5.0.4

**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

Product Name	Product Version	Uploaded By	Date of Upload
license	5.0.4	Administrator	Sunday, May 30, 2004 2:58:23 PM PDT
eGate	5.0.4	Administrator	Sunday, May 30, 2004 3:28:04 PM PDT
BatcheWay	5.0.4	Administrator	Sunday, May 30, 2004 3:45:16 PM PDT
HTTPeWay	5.0.4	Administrator	Sunday, May 30, 2004 3:45:17 PM PDT
SMEWebServices	5.0.4	Administrator	Sunday, May 30, 2004 3:45:19 PM PDT
OracleeWay	5.0.4	Administrator	Sunday, May 30, 2004 3:45:20 PM PDT
FileeWay	5.0.4	Administrator	Sunday, May 30, 2004 3:45:21 PM PDT

Select the manifest file (ProductsManifest.xml) from the CDROM

Manifest File:

- 4 In the ADMIN page, click **Browse**.
- 5 In the **Choose file** dialog, click **ProductsManifest.xml**, 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 OTD libraries for ASC/X12 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).
- 11 In the ADMIN page, click the | **upload now** | button.

### 3.2.2. Refreshing Enterprise Designer with AS2 Manager

#### Before you begin

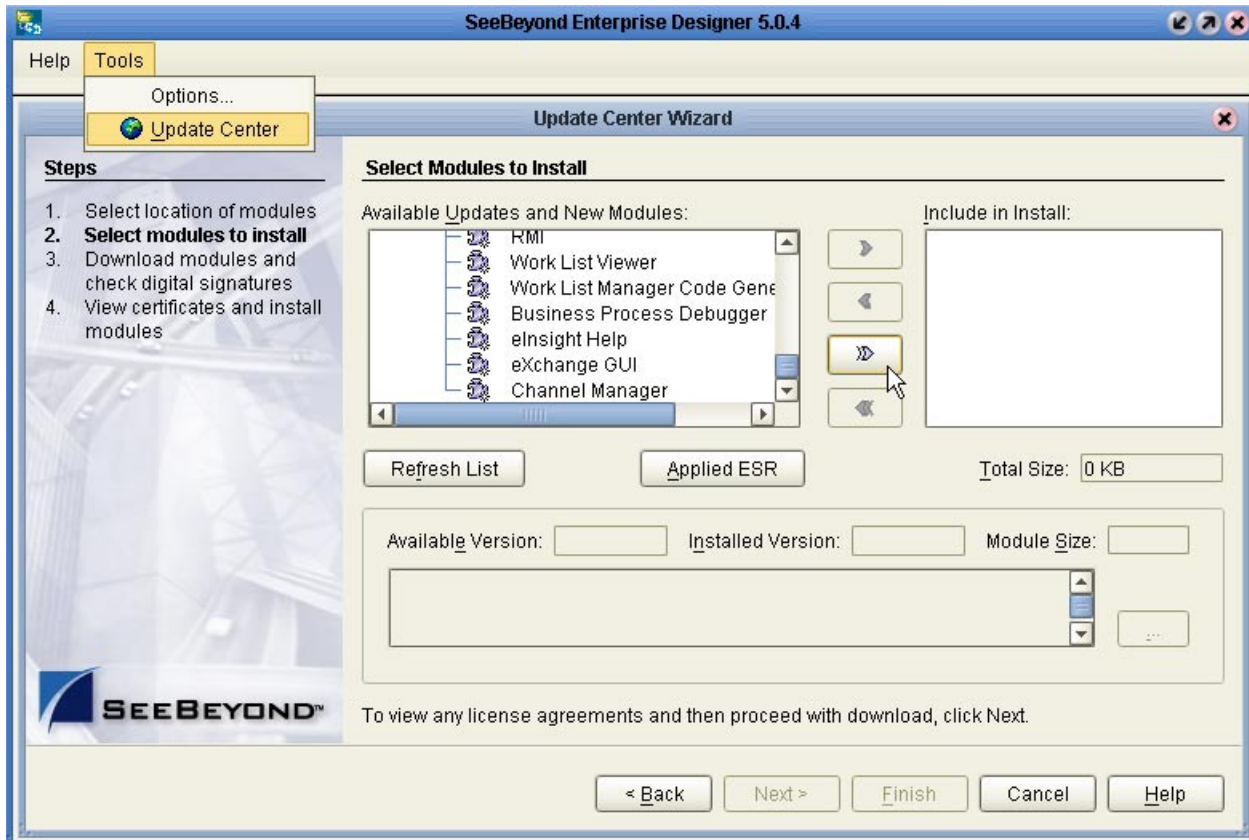
- You must have already downloaded and installed Enterprise Designer.
- A Repository server must be running on the machine where you uploaded the AS2 Manager product files.

#### To refresh an existing installation of Enterprise Designer

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

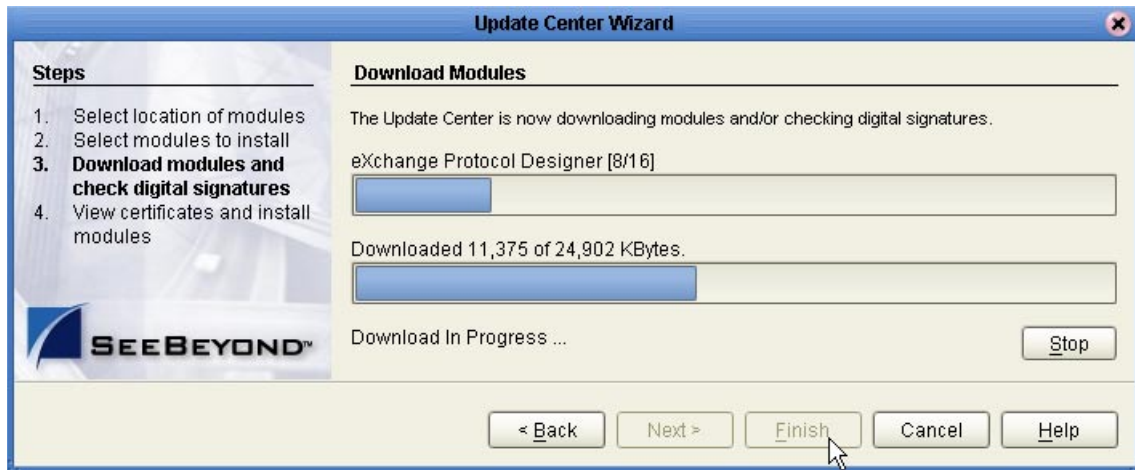
The Update Center shows a list of components ready for updating. See Figure 5.

**Figure 5** Update Center Wizard: Select Modules to Install



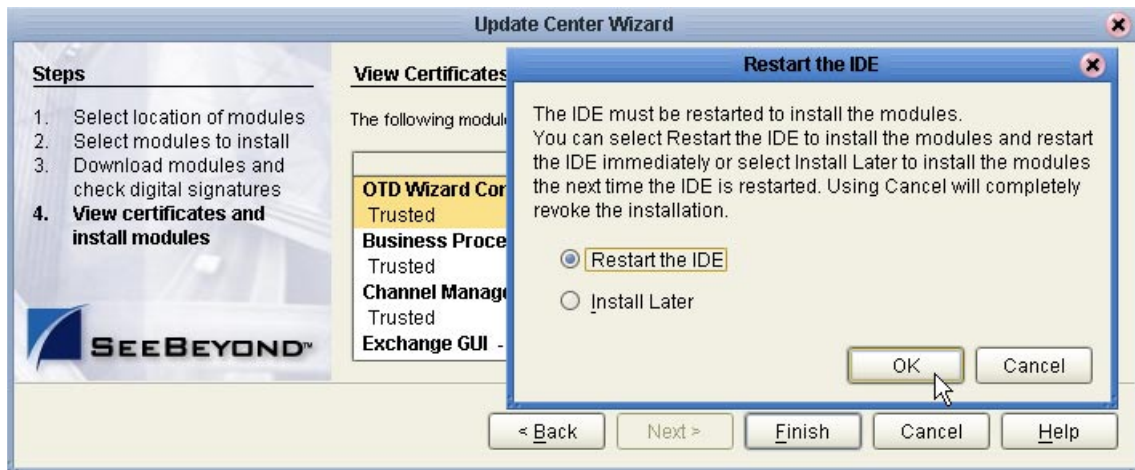
- 3 Click **Add All** (the button with a doubled chevron pointing to the right).  
All modules move from the Available/New pane to the **Include in Install** pane.
- 4 Click **Next** and, in the next window, click **Accept** to accept the license agreement.  
The wizard shows you the progress of the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

---

## 3.3 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*.

## 3.4 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	<a href="http://java.sun.com/j2se/1.4.2/download.html">http://java.sun.com/j2se/1.4.2/download.html</a>
AIX	1.4.1	<a href="http://java.sun.com/products/archive/j2se/1.4.1_07/index.html">http://java.sun.com/products/archive/j2se/1.4.1_07/index.html</a>

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, the link [Download jce\\_policy-1\\_4\\_2.zip](#); or, for JRE 1.4.1, 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.

The steps for the sample implementation occur in these phases:

<b>Initial Setup Steps</b>	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 <a href="#">on page 24</a> .
<b>Design Steps in Enterprise Designer</b>	In these steps, you use Enterprise Designer to add and configure externals, view components in the B2B host project, and it, creating an eXchange Service for the B2B host. Then you view the components in the main project and activate it. See section 4.2 starting <a href="#">on page 26</a> .
<b>Design Steps in ePM</b>	In these steps, you use the eXchange Partner Management (ePM) facility to create a trading partner, view and specify parameter values for it, and activate it. See section 4.3 starting <a href="#">on page 35</a> .
<b>Runtime Steps</b>	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 <a href="#">on page 44</a> .

## 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
ToInternalFileDeliveryProtocol		JMSErrorHandler
ToInternalJMSDeliveryProtocol		

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

For Inbound Messages	For Outbound Messages	Other
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 24](#)
- [Setting Up the Sample Environments on page 26](#)
- [Importing the Sample Projects on page 25](#)

### 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

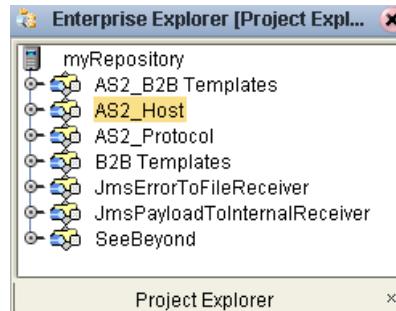
### To import the sample projects

*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 the root level whose name is identical to one of the five 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 environments 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 26](#)
- [Viewing and Activating the B2B Host Project on page 29](#)
- [Configuring the eXchange Services with Crypto Information on page 31](#)
- [Creating and Activating the Project Deployment Profiles on page 32](#)

### 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 28](#).

To create the sample environment

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

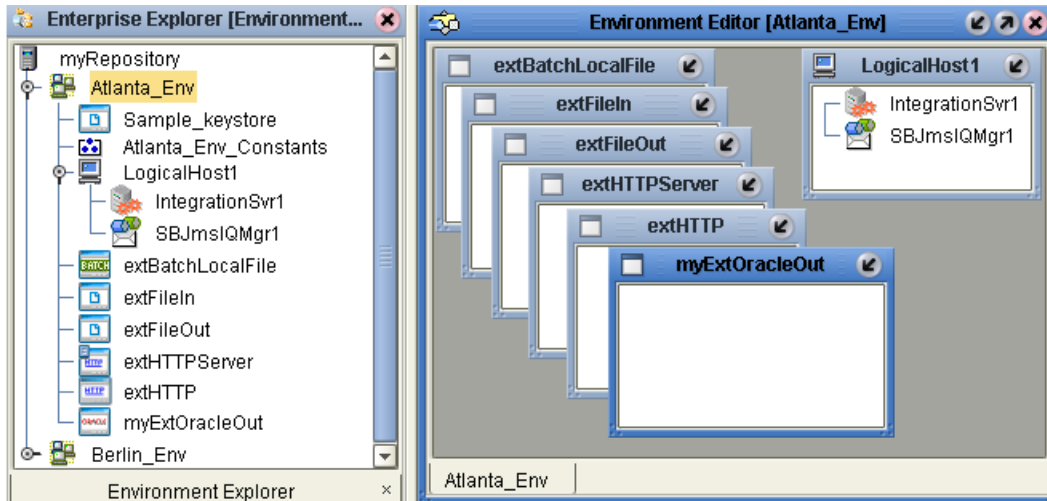
- 2 In the Environment Explorer tree, right-click the repository and, on the popup context menu, click **New Environment**
  - ♦ Rename the newly created environment to **Atlanta\_Env**.
- 3 Right-click Atlanta\_Env and, on the menu, click: **New Logical Host**
  - ♦ Retain the default name: LogicalHost1
- 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 Server External System**
  - ♦ Name it **extHTTPServer** and click OK.
- 11 Right-click Atlanta\_Env > **New HTTP External System**
  - ♦ Name it **extHTTP** and click OK.

**Tip:** To configure for HTTPS, see the *HTTP(S) eWay Intelligent Adapter User's Guide*.

- 12 Right-click Atlanta\_Env > **New Oracle External System**
  - ♦ Name it **myExtOracleOut**, designate it **Outbound Oracle eWay**, and click OK.
- 13 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.
- 14 On the main toolbar, click  **Save All**.
- 15 For Berlin: Repeat all previous steps, but instead of “Atlanta”, substitute “Berlin”. (Also, in step 4, be sure to change the base port to 28000).

*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 5.0.4 Oracle database must be accessible, and you must know its SID and username/passwords. This sample assumes you have already set up distinct username/password combinations for two trading partners, “A” and “B”: for Atlanta, the sample is configured with username/password = **ex504\_A/ex504\_A**; for Berlin, the sample is configured with username/password = **ex504\_B/ex504\_B**

- 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 5.0.4 database user “B”)
  - ◆ **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:** *ex504\_B* (change this to username “B” for your eXchange 5.0.4 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 four 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 eight messaging actions, alternating inbound and outbound.
- **AS2\_Host** is the B2B host itself:
  - ♦ Its Business Protocols window references one service, under the **AS2** MAD: **myHost\_MsgingService** (from the current project)
  - ♦ 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.

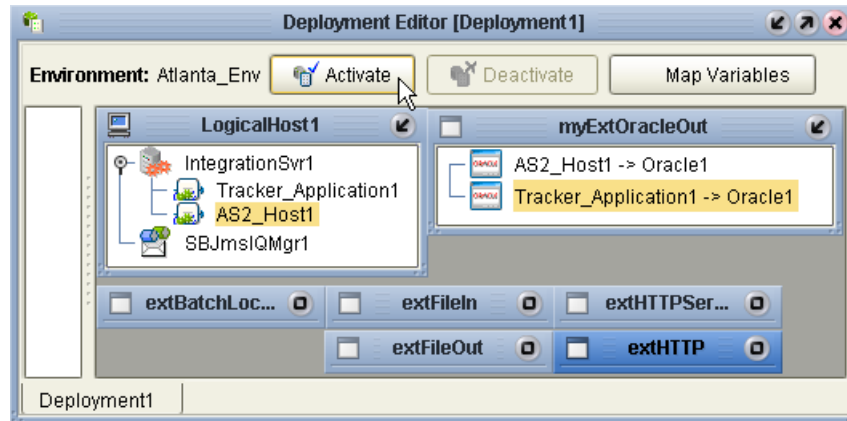
- 1 Right-click the **AS2\_Host** project and, on the popup context menu, point at **New** and click **Deployment Profile**
- 2 Keep the default name, point it at **Atlanta\_Env**, and click OK. (The sample assumes that Deployment1 points at Atlanta\_Env and Deployment2 points at Berlin\_Env. If you 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.

- 3 On the right side, minimize all windows except LogicalHost1 and myExtOracleOut.
- 4 One by one, drag the two services into LogicalHost1 and under **IntegrationSvr1**.
- 5 One by one, drag the two Oracle eWays into **myExtOracleOut**. See Figure 10.

- 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 28.**
  - ♦ It was misdefined as inbound. Delete *myExtOracleOut* and re-create it as outbound. Then: Click **Save All**, followed by  **Refresh All from Repository.**

**Figure 10** Deployment Profile for B2B Host, Before Activation



- 6 Click  **Save All**, and then click **Activate**.
  - 7 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**.
  - 8 For Berlin: Open Environment Explorer, 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 11.

**Figure 11** 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

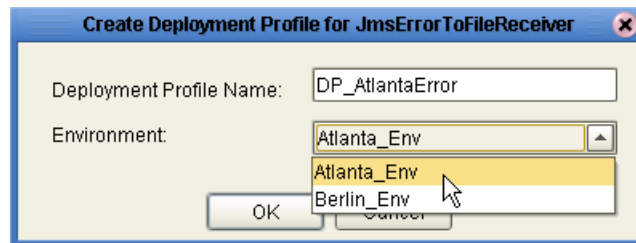
*Before you begin:* Both environments (Atlanta\_Env and Berlin\_Env) must already contain eXchange services that have been configured with crypto information; if necessary, see the [procedure on page 29](#) and/or the [procedure on page 31](#).

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 12.

**Figure 12** Deployment Profile for Error Processing



The Deployment Editor's left pane displays: A long list of services; a topic (JMSErrorHandler1->Host...) and two queues; an inbound eWay for HTTPServer; outbound eWays for HTTP and BatchLocalFile; and eWays for one inbound and four outbound eXchange services.


- 3 Minimize all externals except LogicalHost1 and ExtFileOut.
- 4 Drag the service (JmsErrorToFileBP) into LogicalHost1 and under **IntegrationSvr1**.
- 5 Drag the topic (HostError....->Jms...) into LogicalHost1 and under **SbJmsIQMgr1**.
- 6 One by one, drag the two outbound File eWays into **extFileOut**.
- 7 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**.
- 8 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.*

- 9 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

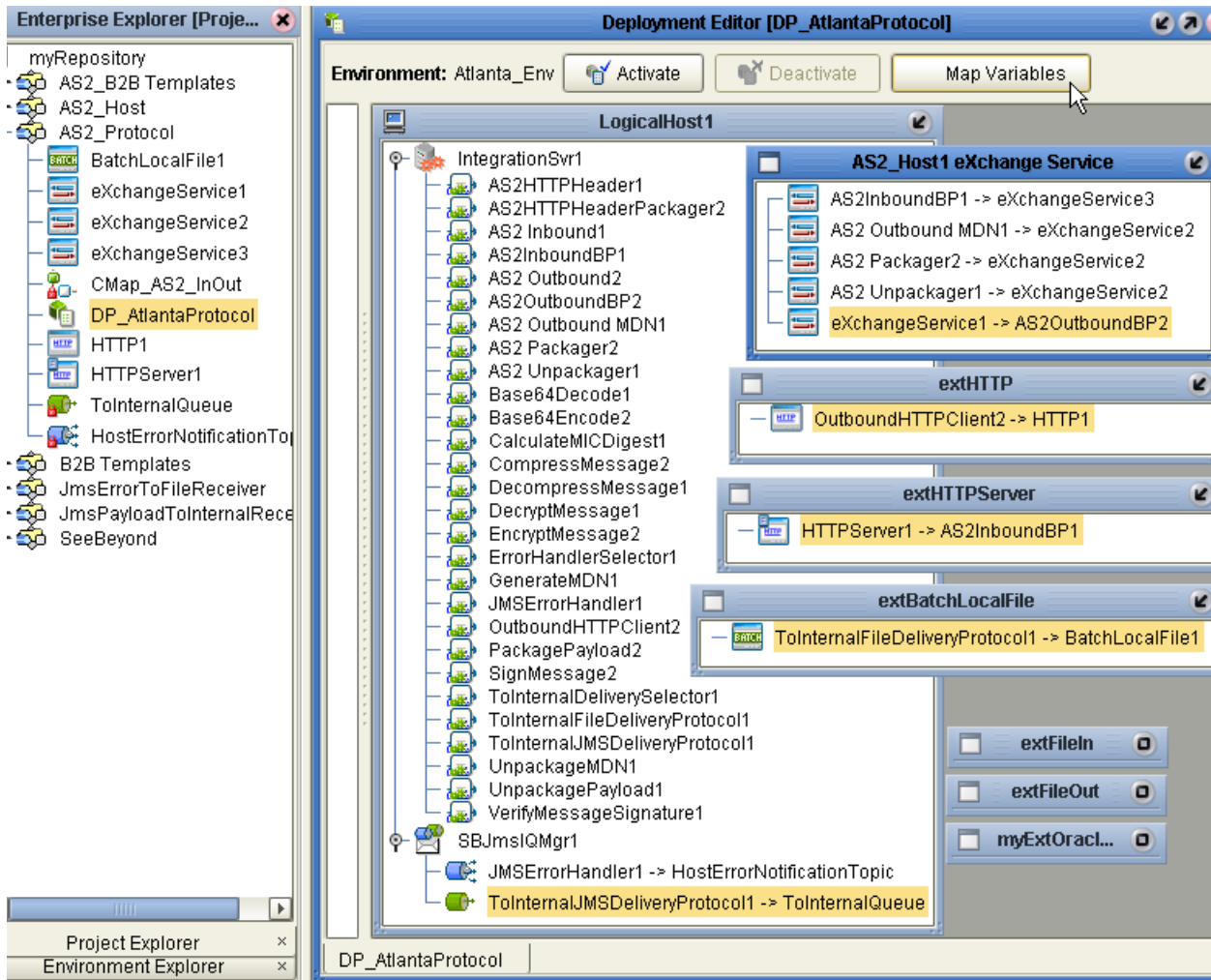
- 1 In the Project Explorer tree, right-click **JmsPayloadToInternalReceiver** and, on the popup context menu, point at **New** and click **Deployment Profile**
- 2 Rename it to **DP\_AtlantaPayload**, point it at **Atlanta\_Env**, and click OK.
- 3 Minimize all externals except LogicalHost1 and ExtFileOut.
- 4 Drag the service (JmsPayloadTo...) into LogicalHost1 and under **IntegrationSvr1**.
- 5 Drag the queue (ToInternalQueue...) into LogicalHost1 and under **SbJmsIQMgr1**.
- 6 Drag the outbound File eWay into **extFileOut**.
- 7 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**.
- 8 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**.
- 9 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 13 on page 34](#).

- 1 In the Project Explorer tree, right-click **AS2\_Protocol** and, on the popup context menu, point at **New** and click **Deployment Profile**
- 2 Rename it to **DP\_AtlantaProtocol**, point it at **Atlanta\_Env**, and click OK.  
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 One by one, drag all services into LogicalHost1 and under **IntegrationSvr1**.
- 4 One by one, drag the topic and queue into LogicalHost1 and under **SbJmsIQMgr1**.
- 5 Drag the outbound BatchLocalFile eWay into **extBatchLocalFile**.
- 6 Drag the inbound HTTPServer eWay into **extHTTPServer**.
- 7 Drag the outbound HTTP eWay into **extHTTP**.
- 8 Drag the eXchange Service eWays into AS2\_Host1 eXchange Service. When you drag and drop the inbound service, specify **AS2** as the protocol. See Figure 13.

Figure 13 Deployment Profile for AS2\_Protocol



9 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).

10 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.

11 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 ePM

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 36](#)
- [Configuring Trading Partner Parameters for the “Berlin” TP on page 36](#)
- [Activating the Trading Partner on page 40](#)
- [Configuring Trading Partner Parameters for the “Atlanta” TP on page 40](#)
- [Activating the Trading Partner on page 44](#)

*Before you begin:* Your repository and your eXchange 5.0.4 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)

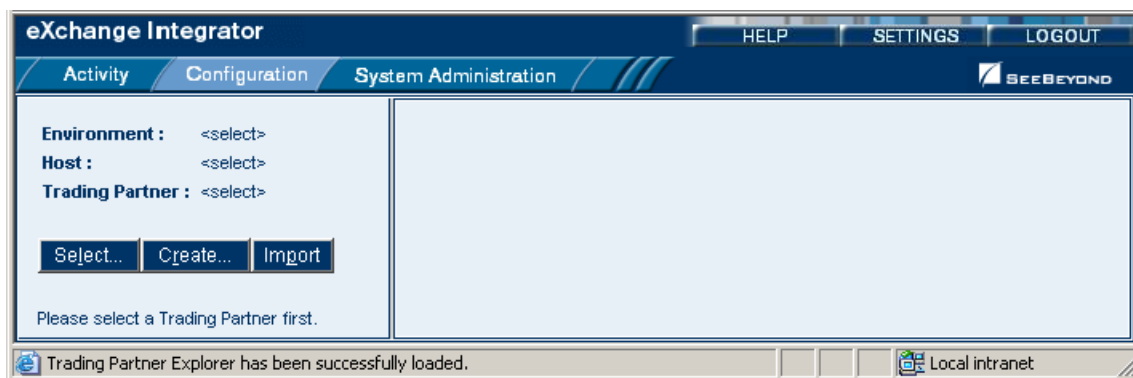
- 1 Start a *new* browser session (that is, do *not* clone a new window of an existing session) pointing it at a repository URL, with **epm** appended. For example:
  - ♦ If your repository were running local on port 12000, the URL would be:  
http://localhost:12000/epm
  - ♦ For a repository running 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 14.

**Figure 14** 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 36](#)
- [TP “Berlin”: Parameters for the Internal Delivery Channels on page 39](#)

#### 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.

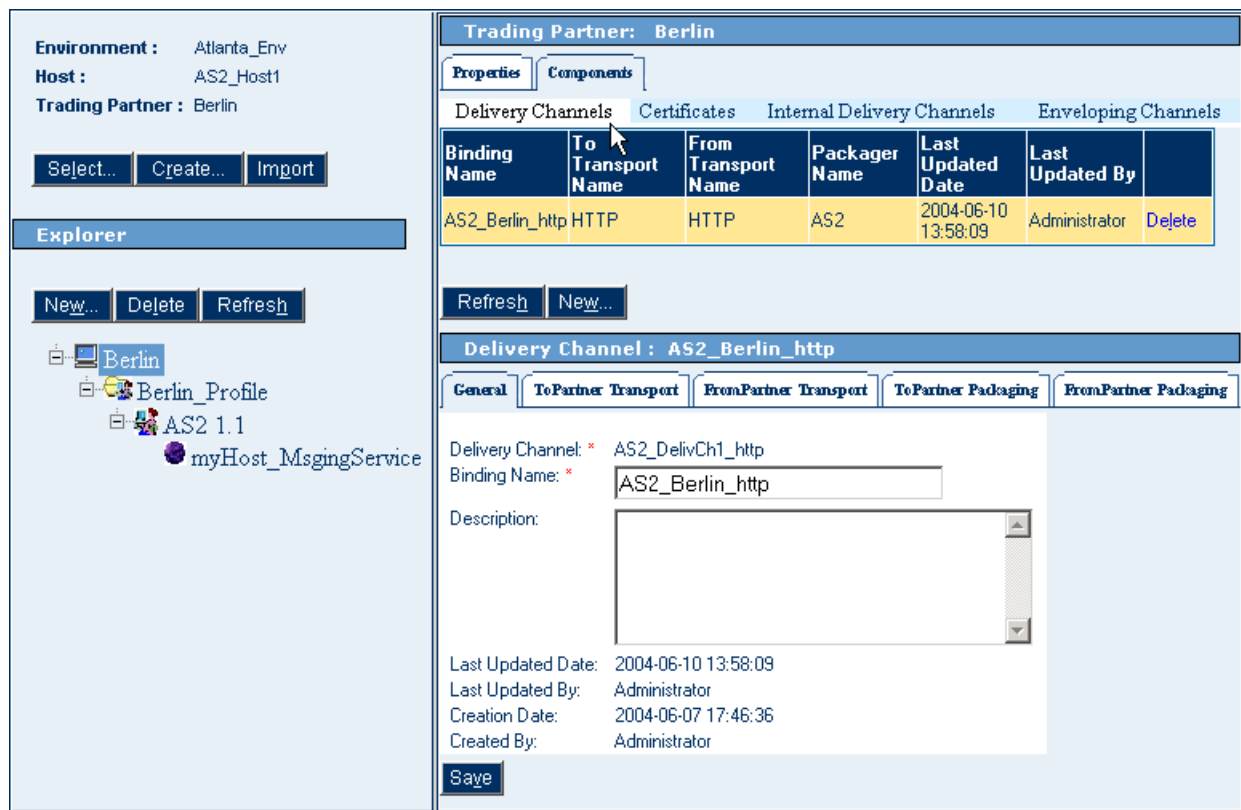
**Table 4** Delivery Channel Parameters for Trading Partner “Berlin”

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

- 3 Click the binding name, **AS2\_Berlin\_http**.

The general properties for this binding of the delivery channel (**AS2\_DelivCh1**) are displayed. See Figure 15.

**Figure 15** 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**.

- Click the **ToPartnerPackaging** tab, make 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>b</b> _pubcert
AS2_FROM: *	Atlanta
AS2_TO: *	Berlin
AS2_HOST: *	AS2_Host1
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:	<code>http://localhost:18004/Deployment1_servlet/Inbound</code>
AS2_SUBJECT: *	EDI Message <b>from Atlanta</b>
AS2_COMPRESSED: *	TRUE
AS2_COMPRESSED_BEFORE_SIGNED: *	FALSE
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>b</b> _pubcert
AS2_FROM: *	Berlin
AS2_TO: *	Atlanta
AS2_REPORTING_UA: *	Atlanta_CA
AS2_POSITIVE_MDN_DISPOSITION_MSG:	Positive DN msg <b>from Atlanta</b>

*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 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 “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** 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 5.0.4 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 41](#)
- [TP “Atlanta”: Parameters for the Internal Delivery Channels on page 43](#)

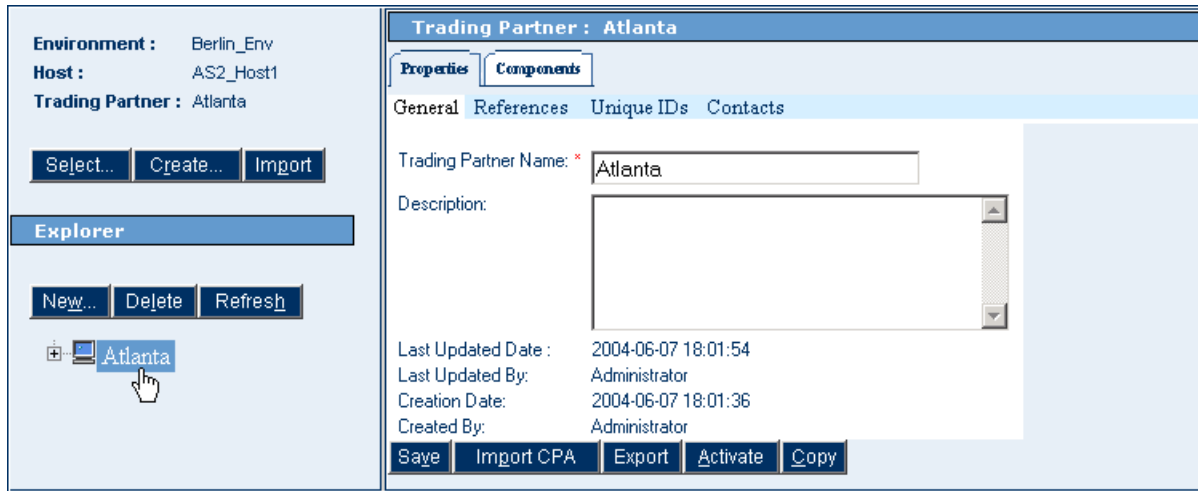
To find the “Atlanta” trading partner (in Berlin\_Env)

- 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 16.



**Figure 16** 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 16.

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**.

- 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**.

- Click the **ToPartnerPackaging** tab, make 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: *	AS2_Host1
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 <b>from Berlin</b>
AS2_COMPRESSED: *	TRUE
AS2_COMPRESSED_BEFORE_SIGNED: *	FALSE
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 13), and then click **Save**.

**Table 13** FromPartnerPackaging Parameters for Trading Partner “Atlanta”

Signature Certificate	companyb_pubcert
AS2_FROM: *	Atlanta
AS2_TO: *	Berlin
AS2_REPORTING_UA: *	Berlin_CA
AS2_POSITIVE_MDN_DISPOSITION_MSG:	Positive DN msg <b>from Berlin</b>

*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** 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 5.0.4 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 45](#)
- [Supplying the Input Data on page 46](#)
- [Using the Message Tracking Facility on page 46](#)

## 4.4.1 Starting the Logical Hosts

These steps assume you have already installed two or more logical hosts. 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 5.0.4 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\_%d.dat  
Error payload is written to ... \Sample\Data\Atlanta\ErrorTopic\_payload\_%d.dat  
Internal messages are received at ... \Sample\Data\Atlanta\toInternal\_%d.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\_%d.dat  
Error payload is written to ... \Sample\Data\Berlin\ErrorTopic\_payload\_%d.dat  
Internal messages are received at ... \Sample\Data\Berlin\toInternal\_%d.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 5.0.4 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:

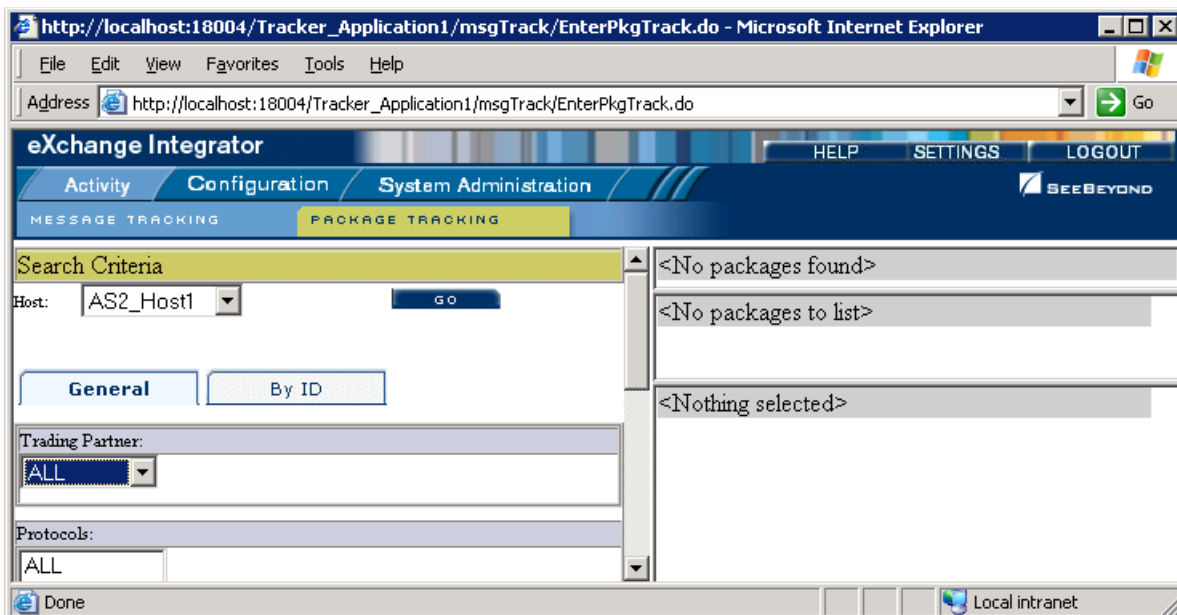
- ♦ *<loghostname>* is the hostname of the physical machine where you ran bootstrap.exe and pointed it at a particular environment and logical host.
- ♦ *<port>* is the port number assigned in that environment and logical host (in the configuration for Integration Server .> ... > Web server).
- ♦ *<TrackerAppName>* is the name of the Message Tracking application as it appears in the connectivity map. 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 17.

Figure 17 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 18.

**Figure 18** Message Tracking, Showing Initial Search Results

The screenshot shows the 'MESSAGE TRACKING' application window. The 'PACKAGE TRACKING' tab is active. The 'Search Criteria' section on the left includes a 'Host' dropdown set to 'AS2\_Host1', a 'Trading Partner' dropdown set to 'Berlin', and a 'Protocols' dropdown set to 'AS2'. 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 Results' section on the right displays a table of ten results. The table has columns for Package ID, Trading Partner, Protocol, Package Type, Conversation ID, Direction, Error Data, Process Date, Response Req, and Ack Date. The results show a mix of inbound and outbound messages from Berlin and Atlanta hosts.

Package ID	Trading Partner	Protocol	Package Type	Conversation ID	Direction	Error Data	Process 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:** 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 19.

**Figure 19** Message Tracking, Showing Filtered Results

The screenshot shows the Message Tracking application interface. On the left, the 'Search Criteria' pane is active, showing the following settings:

- Host: AS2\_Host1
- Trading Partner: Berlin
- Protocols: AS2 (selected)
- Filters:
  - Error Type: Error
  - Direction: Inbound
  - Date Filter: Acknowledgement date
  - From and To: Acknowledgement date

At the bottom of the left pane are 'SEARCH' and 'CLEAR' buttons. The right pane, 'Search Results', displays a table of results for Page 1 of 1. The table has the following columns: Package ID, Trading Partner, Protocol Type, Conversation ID, Direction, Error Data, Process Date, Response Req, and Ack Date.

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

Below the table, it indicates '<Nothing selected>'. Navigation controls at the top of the results pane include 'Print', '<< Previous', 'Page 1 of 1', 'Next >>', and 'Go to Page: 1'.

### 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 20. You can use cut, copy, and paste on any text in the window.

**Figure 20** Message Tracking, Showing Package Details and Message Content

The screenshot shows the Message Tracking interface. At the top, there is a 'Search Results' header with navigation controls: 'Print', '<< Previous', 'Page 1 of 1', 'Next >>', and 'Go to Page: 1' with a 'GO' button. Below this is a table of search results. The first row is highlighted, and its Package ID, '<20040611165132643@Berlin\_Host>', is circled in green. A green arrow points from this ID to the 'Details for package' section below. In this section, the 'Message' tab is selected, and an 'OPEN' button is circled in green. A second green arrow points from this button to a separate window titled 'Message View - Microsoft Internet Explorer'. This window displays the message content for the selected package ID, including headers like 'Content-Type: multipart/signed...' and 'Content-Type: Application/EDI-X12', and a body of EDI data. To the left of the message view window, the text 'Result is displayed in a new window.' is written in green.

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

Details for package: <20040611165132643@Berlin\_Host>

Following are the messages available to the selected item:

Original message:

```

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*S1S1S1S1S1S1S1S1S1R1R
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
    
```

# Index

## A

- Adobe (URL for) 9
- AS2
  - (defined) 14
  - compared to AS1 14
  - specification for 13, 16
- AS2 message structure 14
  - (illustrated) 15
- AS2 sample implementation 23–50

## C

- CodeGenException (JNDI name prefix) 32
- configuring
  - HTTPS 26
  - Oracle eWay 26, 28
  - ports 26, 27, 37, 38, 41
- conventions 8

## D

- database for eXchange
  - supported levels 17
- document conventions 8

## E

- EDI-INT Requirements document 16
- EDI-INT, defined 14
- encryption .jar files
  - updating 22
  - where to download 22
- error "Failed to obtain JNDI name prefix" 32
- eXchange database
  - system requirements 17
- eXchange support
  - for Oracle databases 17
  - for platforms 17

## H

- HTTPS (HTTP on SSL), configuring 26

## I

- IETF (URLs for) 16

## J

- JNDI name prefix (error message) 32

## L

- local\_policy.jar, updating 22

## M

- message structure, AS2 14

## O

- operating systems supported by eXchange 17
- Oracle database for eXchange
  - supported levels 17
- Oracle eWay, configuring 26, 28

## P

- platforms supported by eXchange 17
- policy .jar files
  - updating 22
  - where to download 22
- ports, configuring 26, 27, 37, 38, 41

## S

- sample implementation 23–50
- Secure Messaging Extension (SME)
  - updating policy .jar files for 22
- SME
  - updating policy .jar files for 22
- SSL (Secure Sockets Layer), configuring 26
- structure of an AS2 message 14
- supporting documents 9

## U

- URLs
  - for Adobe Acrobat Reader 9
  - for IETF 16
  - for SeeBeyond 9
- US\_export\_policy.jar
  - updating 22
  - where to download 22

W

writing conventions 8