SUN B2B SUITE 2.0

eXchange Integrator User's Guide

Release 5.1.0

Sun microsystems

Part Number: 820-0264-10

Version 20070209155034

Contents

Chapter 4

Chapter 5

Chapter 6

Chapter 8

Chapter 9

Chapter 10

Appendix E

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides an overview of the User's Guide for Sun B2B Suite eXchange™ Integrator. eXchange is the prerequisite base product for all products in B2B Suite 2.0.

What's in This Chapter

1.1 What's New in This Release

At this release, eXchange Integrator includes the following new features and changes:

- *Streamlining of B2B Host setup in Enterprise Designer.* There is no longer any need to set up Delivery Channels or Internal Delivery Channels.
- *Reorganization of eXchange Partner Manager (ePM).* Parameters are organized in a layered fashion, that makes clear the current value is for each parameter and how the setting inherits from or overrides other layers in the hierarchy.
- Directory Server (LDAP). For performance and scalability, trading partner information is now stored on a required LDAP-compliant directory server rather than in the eXchange Oracle database.

1.2 About This Document

The User's Guide provides instructions and background information for all users of the eXchange product. This guide is designed for managers, system administrators, and others who use eXchange.

The purpose of this guide is to help you understand the basic architecture and functionality of eXchange; understand the relationship of eXchange to other components in the B2B Suite and Java CAPS; and learn about the eXchange components and editors and how to use them in your environment.

1.2.1 What's in This Document

This guide is organized into the following chapters:

- 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 eXchange Integrator"** provides step-by-step instructions for installing the eXchange product and setting it up for use.
- Chapter 4, **"eXchange Features"** highlights and discusses the key features and components supplied with eXchange.
- Chapter 5, **"Using eXchange in Enterprise Designer"** provides step-by-step procedures for working with eXchange at design time and deploying projects.
- Chapter 6, **"Using B2B Web Facilities"** provides step-by-step procedures for working with eXchange's Web-based GUIs—Partner Manager (ePM) and Message Tracking.
- Chapter 7, **"Designing Business Processes"** provides step-by-step procedures for designing and deploying business processes using eXchange Protocol Designer.
- Chapter 8, **"Configuring Exception Handling"** explains the use of business processes for handling exceptions.
- Chapter 9, **"Persistence and Monitoring"** explains how the eInsight engine can be used to persist and monitor data passing through business processes at runtime.
- Chapter 10, **"Monitoring Business Processes at Run Time"** explains how to use Enterprise Manager at runtime to monitor and log the status of individual business processes.
- Chapter 11, **"eXchange Troubleshooting Tips"** lists signs and symptoms of problems sometimes encountered using eXchange and suggests diagnoses and remedies.
- The **Glossary of Acronyms** on page 175 lists and explains special acronyms and initialisms that occur in this guide.
- **Appendix A**, **"Obtaining and Setting Up an LDAP Server"** lists and describes the tools available to you in eXchange Protocol Designer.
- **Appendix B**, **"Method Palette"** lists and describes the tools available to you in eXchange Protocol Designer.
- **Appendix C, "Scalability"** is a version of a white paper published previously as "Sun B2B Suite 2.0 Scalability: Scaling Horizonally to Increase Throughput".
- **Appendix D, "Reliability"** is a version of a white paper published previously as"Sun B2B Suite 2.0 Performance: Optimizing Throughput".
- **Appendix E, "Performance"** is a version of a white paper published previously as "Configuring the System for Continuous Run".

1.2.2 Intended Audience

This guide is intended for experienced computer users who have the responsibility of helping to set up and maintain a fully functioning Java Composite Application Platform Suite (Java CAPS) system. This person must also understand any operating systems on which Java CAPS will be installed , and must be thoroughly familiar with Windows-style GUI operations.

1.2.3 Text Conventions

The following conventions are observed throughout this document.

Table 1 Text Conventions

Text Convention	Used For	Examples
Bold	Names of buttons, files, icons, parameters, variables, methods, menus, and objects	- Click **OK**. - On the **File** menu, click **Exit**. - Select the **eGate.sar** file.
`Monospaced`	Command line arguments, code samples; variables are shown in ***`bold italic`***	`java -jar` ***`filename`***`.jar`
Blue bold	Hypertext links within document	See **Text Conventions** on page 18
Blue underlined	Hypertext links for Web addresses (URLs) or email addresses	http://www.sun.com

1.2.4 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.

1.3 Related Documents

For late-breaking information on the B2B Suite, refer to the B2B_Readme file on the product media.

For information on other products in the B2B Suite, refer to the following guides:

- *ASC X12 OTD Library User's Guide*
- *ASC X12 Protocol Manager User's Guide*
- *HIPAA OTD Library User's Guide*
- *HIPAA Protocol Manager User's Guide*

For more information about Java CAPS products, refer to the following:

Title	Filename
Java CAPS Installation Guide	CAPS_Install_Guide.pdf
Java CAPS Deployment Guide	CAPS_Deployment_Guide.pdf
eGate Integrator User's Guide	eGate_UG.pdf
eGate Integrator System Administration Guide	eGate_Sys_Admin_Guide.pdf
eGate Integrator JMS Reference Guide	eGate_JMS_Ref.pdf
eInsight Business Process Manager User's Guide	eInsight_UG.pdf
eWay Batch Adapter User's Guide	Batch_eWay_UG.pdf
eWay HTTPS Adapter User's Guide	HTTP_eWay_UG.pdf
eWay File Adapter User's Guide	File_eWay_UG.pdf
eWay LDAP Adapter User's Guide	LDAP_eWay_UG.pdf
eWay Adapter for Oracle User's Guide	Oracle_eWay_UG.pdf
ASC X12 Protocol Manager User's Guide	ASC_X12_Manager_UG.pdf
ASC X12 OTD Library User's Guide	ASC_X12_OTD_Lib_UG.pdf
HIPAA Protocol Manager User's Guide	HIPAA_Manager_UG.pdf
HIPAA OTD Library User's Guide	HIPAA_OTD_Lib_UG.pdf
Readme for Java CAPS / eGate Integrator	Readme.txt

1.4 Sun Microsystems, Inc. Web Site

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

http://www.sun.com

1.5 Documentation Feedback

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

CAPS_docsfeedback@sun.com

Chapter 2

Overview

This chapter provides a general overview of eXchange and its place in the Java Composite Application Platform Suite (Java CAPS), including system descriptions, general operation, and basic features.

What's in This Chapter

2.1 Summary of Features

eXchange provides an open B2B protocol framework to support standard EDI and B2B business protocols and delivery protocols. Not only does it support existing standard protocols, with an extensive set of prebuilt business processes (BPs), it also provides the tools and framework to create and adopt new protocols and to build custom BPs.

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

The trading partner management facility, eXchange Partner Manager (ePM), is provided via a Web interface. Trading partner profiles can be created from scratch and configured manually; or, for easy interoperability, they can be imported. Each trading partner profile is identified by a unique ID determined by the enterprise. ePM is also used to configure parameters for acknowledgments, compression, industry-standard encryption and decryption, and nonrepudiation.

At run time, key 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 Java CAPS, this provides the enterprise with a complete solution for troubleshooting and managing all eBusiness activities.

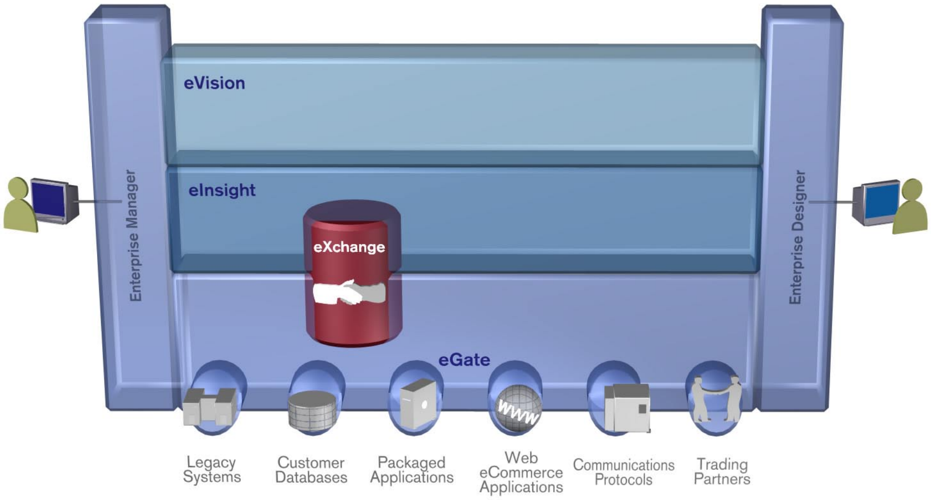
2.2 eXchange and Java CAPS

eXchange relies on the Java Composite Application Platform Suite (Java CAPS). 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.

2.2.1 Integration with Java CAPS

eXchange is tightly integrated with Java CAPS and runs as a component within the Java CAPS environment. Figure 1 illustrates how eXchange and other Java CAPS components work together.

Figure 1 eXchange and the Java CAPS Framework



2.3 Architectural Overview

eXchange centers around the concept of a *transaction profile* for each trading partner relationship. Each transaction profile specifies values for parameters used in three different protocol layers:

- *business protocols* such as X12 or HIPAA that codify agreements at the level of business messages/envelopes: business transaction types, validation, enveloping, batching, acknowledgment, and so forth

- *delivery protocols* that codify agreements at the level of packaging and delivery: compression/decompression, encryption/decryption, signing/verification, and so forth
- *transports* such as HTTP or FTP that codify agreements on where and how to deliver messages to and from trading partners.

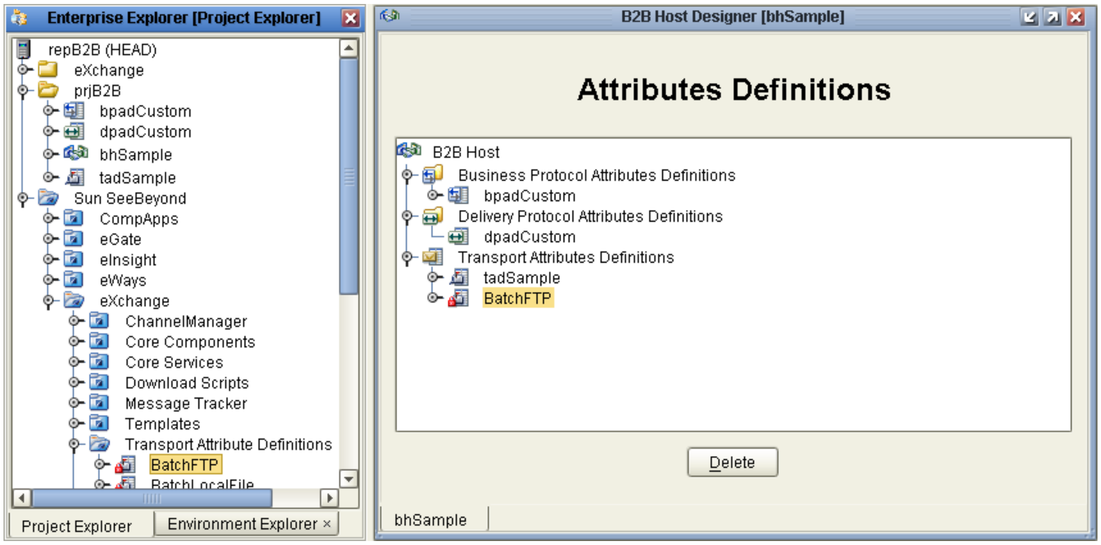
eXchange uses the following key components:

- **B2B Host Designer** and attribute definitions — Using the **Enterprise Designer** GUI framework, eXchange provides an editor for setting up B2B environments, called the *B2B Host Designer*. Each B2B Host provides metadata for transports, delivery protocols, and business protocols, in the form of *attribute definitions*. The attribute definition metadata is stored on a directory server via LDAP.
- **Business Services (BPs, JCDs, OTDs)** — eXchange leverages eGate and eInsight GUIs in the Enterprise Designer framework to provide a substantial library of prebuilt Business Processes (BPs) and Java Collaboration Definitions (JCDs) for selecting and handling protocols and accomplishing such tasks as batching, checking duplicates, and handling errors, as well as Object Type Definitions (OTDs) for communicating with transport-specific eWays and handling the ExStdEvent message.

  In addition to the core services supplied by eXchange, there are protocol-specific BPs, JCDs, and OTDs in each separately installable add-on Protocol Manager, for such industry-standard B2B protocols as X12 and HIPAA.
- **ePM**— eXchange provides a Web-based GUI called eXchange Partner Manager (ePM). This allows you to configure and manage B2B hosts and trading partners. and to assign the parameters that are used in transaction profiles. See **Figure 5 on page 24**.
- **LDAP Server** — eXchange uses an LDAP-compliant directory server to mediate retrieval of trading partner information.
- **eXchange Database** — eXchange uses an Oracle database to store run-time information on correlation and message tracking.
- **Message Tracking**— eXchange provides a Web-based message tracking GUI with powerful filtering and searching capabilities. See **Figure 6 on page 25**.

The illustrations in Figure 2, **Figure 3 on page 23**, and **Figure 4 on page 23** and indicate some of the features provided by the various GUIs.

Figure 2 B2B Host Designer in Enterprise Designer



Enterprise Explorer [Project Explorer]

repB2B (HEAD)
eXchange
prjB2B
bpadCustom
dpadCustom
bhSample
tadSample
Sun SeeBeyond
CompApps
eGate
eInsight
eWays
eXchange
ChannelManager
Core Components
Core Services
Download Scripts
Message Tracker
Templates
Transport Attribute Definitions
BatchFTP
BatchLocalFile

Project Explorer | Environment Explorer ×

B2B Host Designer [bhSample]

Attributes Definitions

B2B Host
Business Protocol Attributes Definitions
bpadCustom
Delivery Protocol Attributes Definitions
dpadCustom
Transport Attributes Definitions
tadSample
BatchFTP

Delete

bhSample

Figure 3 Example of a User-Created Business Process

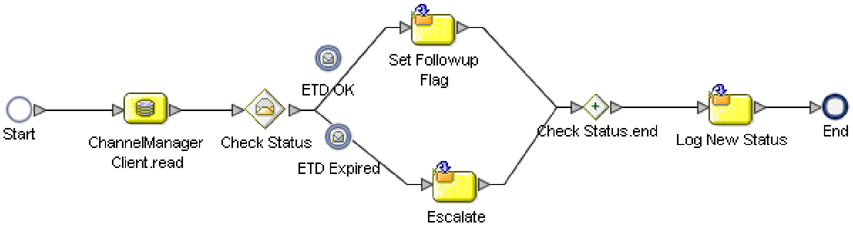


Figure 4 Example of a Prebuilt Business Process (bpEX_DialogTOTP)

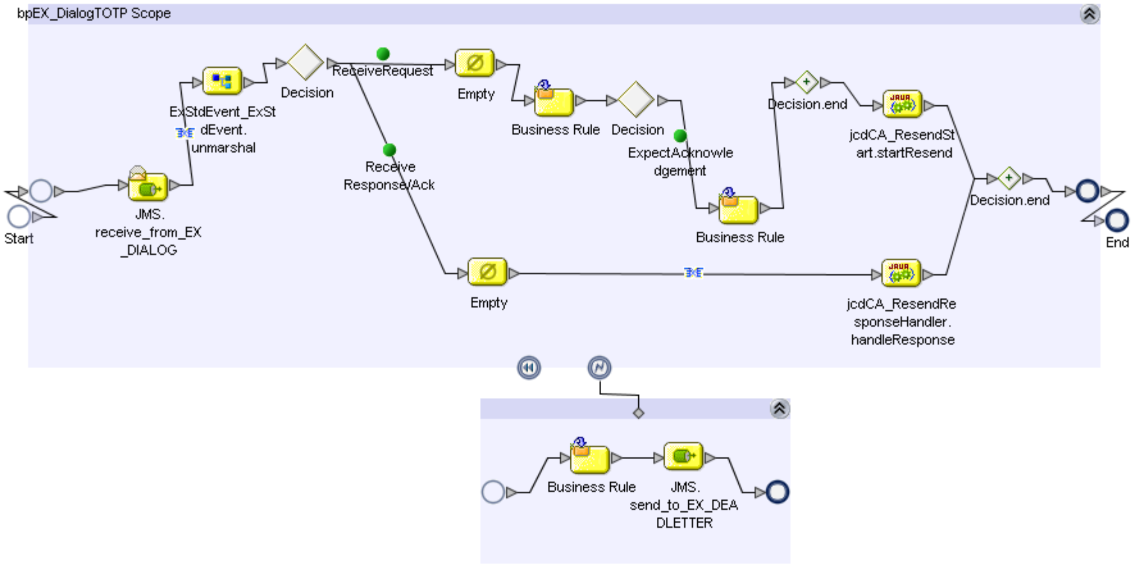
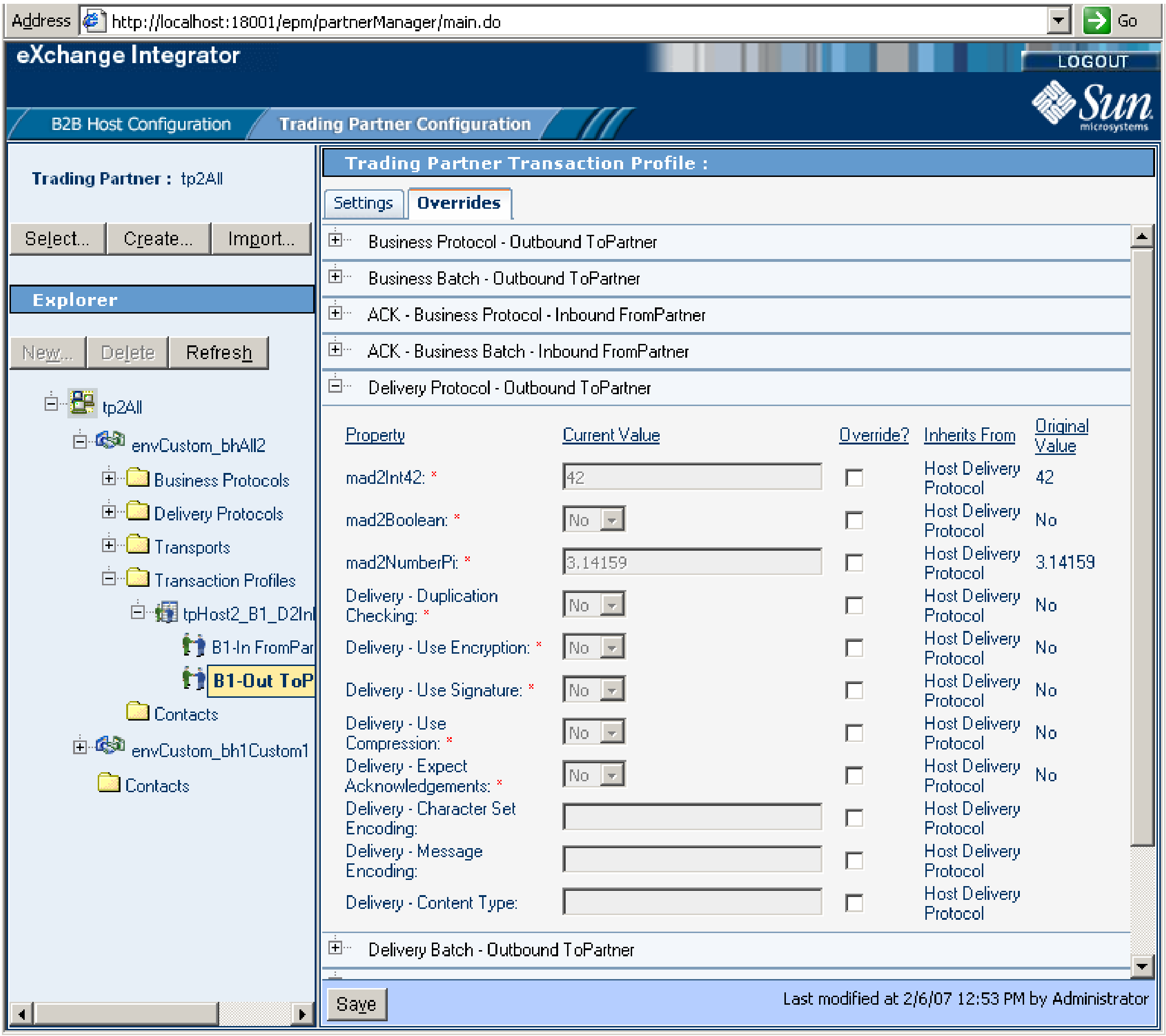


Figure 5 eXchange Trading Partner Configuration



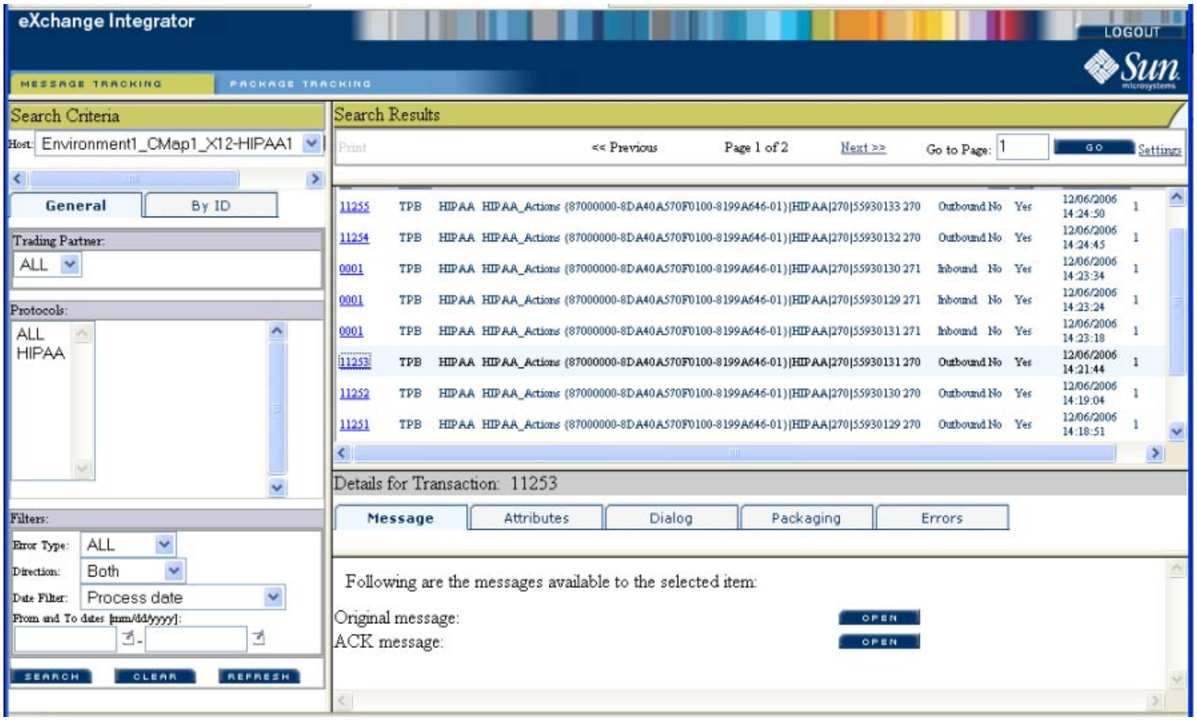
http://localhost:18001/epm/partnerManager/main.do

Trading Partner : tp2All

Property	Current Value	Override?	Inherits From	Original Value
mad2Int42: *	42	□	Host Delivery Protocol	42
mad2Boolean: *	No	□	Host Delivery Protocol	No
mad2NumberPi: *	3.14159	□	Host Delivery Protocol	3.14159
Delivery - Duplication Checking: *	No	□	Host Delivery Protocol	No
Delivery - Use Encryption: *	No	□	Host Delivery Protocol	No
Delivery - Use Signature: *	No	□	Host Delivery Protocol	No
Delivery - Use Compression: *	No	□	Host Delivery Protocol	No
Delivery - Expect Acknowledgements: *	No	□	Host Delivery Protocol	No
Delivery - Character Set Encoding:		□	Host Delivery Protocol	
Delivery - Message Encoding:		□	Host Delivery Protocol	
Delivery - Content Type:		□	Host Delivery Protocol	

Last modified at 2/6/07 12:53 PM by Administrator

Figure 6 eXchange Message Tracking



Host: Environment1_CMap1_X12-HIPAA1

Trading Partner: ALL

Protocols: ALL, HIPAA

Filters:

Error Type: ALL

Direction: Both

Date Filter: Process date

From and To dates [mm/dd/yyyy]:

Page 1 of 2 Go to Page: 1

11255	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930133	270	Outbound	No	Yes	12/06/2006 14:24:50	1
11254	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930132	270	Outbound	No	Yes	12/06/2006 14:24:45	1
0001	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930130	271	Inbound	No	Yes	12/06/2006 14:23:34	1
0001	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930129	271	Inbound	No	Yes	12/06/2006 14:23:24	1
0001	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930131	271	Inbound	No	Yes	12/06/2006 14:23:18	1
11253	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930131	270	Outbound	No	Yes	12/06/2006 14:21:44	1
11252	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930130	270	Outbound	No	Yes	12/06/2006 14:19:04	1
11251	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930129	270	Outbound	No	Yes	12/06/2006 14:18:51	1

Details for Transaction: 11253

Following are the messages available to the selected item:

Original message:

ACK message:

2.4 Process Overview

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

- Design phase within Enterprise Designer
- Design phase within eXchange Partner Manager (ePM)
- Runtime phase

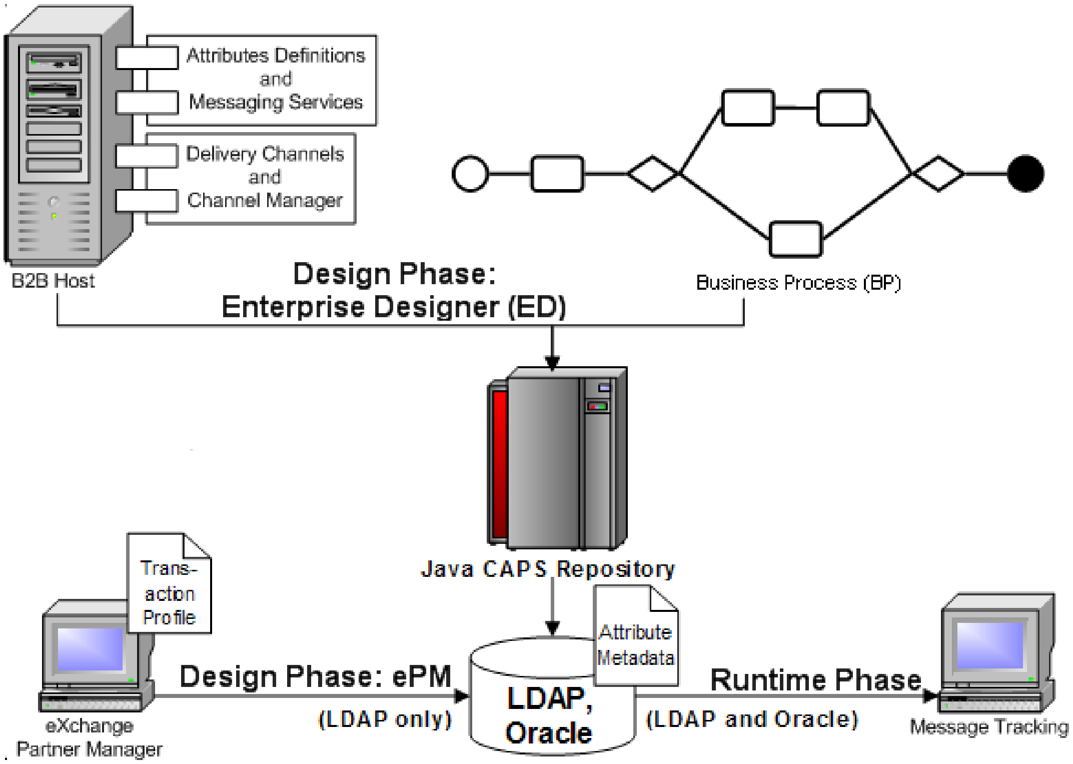
The purpose of the design phases is to: Create metadata for business protocols, delivery protocols, and transports; set up business logic for business services (BPs and JCDs); configure connections with external systems; create and configure trading partners; and associate each trading partner relationship with a fully configured transaction profile. When a trading partner is activated, its transaction profile settings are stored on the LDAP server.

At run time, the Logical Host reads the transaction profile settings from LDAP 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 Oracle database, where they can be filtered and viewed by the Message Tracker facility.

These phases are illustrated in Figure 7 and explained in the following sections:

- **Design Phase: Using Enterprise Designer** on page 26
- **Design Phase: Using ePM** on page 27
- **Runtime Phase** on page 27

Figure 7 eXchange Architecture



2.4.1 Design Phase: Using Enterprise Designer

Within Enterprise Designer, the B2B Host Designer is used to create B2B Hosts. Each B2B Host is a logical collection of business and enveloping attribute definitions, messaging and packaging attribute definitions, and transport attribute definitions.

Attribute definitions supply metadata for a transaction profile—in other words, the *types* of parameters to be supplied for exchanging messages with trading partners.

After the B2B Host is set up, a connectivity map is created to connect it to both an LDAP external and an Oracle external. Building a deployment using this connectivity map and an environment creates an ***eXchangeService*** object in the same environment that contains the LDAP and Oracle externals and a B2B Configurator object. In future releases, the eXchange Service corresponding to the B2B Host will be configurable with keystores, trust stores, and certificates for authentication and nonrepudiation. Other connectivity maps are created, built, and deployed to connect the ePM GUI application with the LDAP external and the Message Tracker application with the Oracle external.

After the eXchangeService is created, it is used in connectivity maps (both user-created and also presupplied) to expose services such as batching/unbatching, dialogs with the trading partner, error-handling, and so forth. When a deployment profile is built and deployed that references the connectivity maps located in the eXchange > **Deployment** project folder, the selector/handler BPs are exposed to ePM and made available for run time.

2.4.2 Design Phase: Using ePM

eXchange Partner Manager (ePM) is used to create and configure trading partners and to create transaction profiles—an association between a particular trading partner and a set of parameters whose metadata are defined by the B2B Host's attribute definitions.

For example, if a B2B Host uses the HTTP transport attributes definition, then a transaction profile for that B2B Host can use HTTP for transport, and must therefore be provided a value for the URL parameter. Or, if it uses the FTP transport, then it must be provided values for hostname, target directory, and so forth.

Activating a trading partner stores all of its transaction profiles' configuration settings into the LDAP database.

2.4.3 Runtime Phase

The Logical Host reads the transaction profile configuration and receives messages from all inbound delivery channels it references. The parameters for each transaction profile dictate how to handle the inbound message, in terms of acknowledgment, decryption, de-enveloping, authentication, and so forth. The business logic of the associated business services (BPs and JCDs connected to the eXchange Service) provide further routing and processing. For an outbound message, the transaction profile parameters dictate how to handle it (in terms of compression, encryption, signature, enveloping, and so forth) and how and where to send it.

The Oracle database keeps track of all messages sent and received. It checks for duplicates and acknowledgments, performs correlations, and also allows you to use the message tracker application to search, filter, and view message-related information, such as receipts, acknowledgments, notifications, errors, and message attributes.

Chapter 3

Installing eXchange Integrator

This chapter explains the prerequisites and steps for installing eXchange Integrator.

What's in This Chapter

3.1 Supported Operating Systems

The *Java CAPS Installation Guide* and the **Readme.txt** files, available on the product media and via Enterprise Manager (Documentation tab), contain up-to-date operating system requirements for each supported platform.

Sun B2B Suite 2.0 is available on the following operating systems:

- Sun Solaris 8, 9, and 10 with required patches (SPARC)
- Sun Solaris 10 (AMD Opteron)
- HP-UX 11i (11.11) on PA-RISC, and 11i v2.0 (11.23) and 11.31 on Itanium with required patches and parameter changes
- IBM AIX 5L, versions 5.2 and 5.3, with required maintenance level patches
- Microsoft Windows 2000 SP3 and SP4, Windows XP SP1a and SP2, and Windows Server 2003 SP1
- Red Hat Enterprise Linux AS 2.1 and AS 3 (Intel x86)
- Red Hat Enterprise Linux AS 3 and AS 4 (AMD Opteron)
- Japanese Sun Solaris 8, 9, and 10 with required patches (SPARC)
- Japanese HP-UX 11i (11.11) on PA-RISC, and 11i v2.0 (11.23) on Itanium with required patches and parameter changes
- Japanese IBM AIX 5L, versions 5.2 and 5.3, with required maintenance level patches

- Japanese Microsoft Windows 2000 SP3 and SP4, Windows XP SP1a and SP2, and Windows Server 2003 SP1
- Korean Sun Solaris 8, 9, and 10 with required patches (SPARC)
- Korean HP-UX 11i (11.11) on PA-RISC, and 11i v2.0 (11.23) on Itanium with required patches and parameter changes
- Korean IBM AIX 5L, version 5.2, with required maintenance level patches
- Korean Microsoft Windows 2000 SP3 and SP4, Windows XP SP1a and SP2, and Windows Server 2003 SP1

3.2 Supported External Applications

This section lists supported database applications for features provided by eXchange.

3.2.1 Database for Message Tracking

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

- Oracle 9.01
- Oracle 9.2
- Oracle 10.1

3.2.2 Database for Persistence and Monitoring via eInsight Engine

In addition, eXchange can optionally use the eInsight engine (supplied with eInsight Business Process Manager) to collect and persist data from your Business Processes. This provides for reliability and recovery, and also enables some monitoring and reporting capabilities in Enterprise Manager. The eInsight engine supports the following databases:

- Oracle 8*i* (8.1.7), 9*i* (9.0.1 and 9.2), and 10*g*

Note: *When creating an Oracle 8.1.7 database, the required minimum db_block_size for eInsight is 16KB*

- Sybase 12.5
- Microsoft SQL Server 2000
- IBM DB2 Universal Database 8.1

3.3 Before You Install

Before you begin installing the B2B Suite, make sure you have done the following:

- You have checked the B2B_Readme file for any late-breaking installation notes.
- You have verified that a Repository server is running on the machine where you will be uploading the product files.

3.4 Installing the Product Files

The steps for installing product files for the B2B Suite are the same as for other products in Java CAPS. You can find general product installation instructions in the *Java CAPS Installation Guide,* which is available on the product media and can also be accessed via Enterprise Manager (Documentation tab).

3.4.1 Uploading B2B Suite Product Files 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 **.sar** files must have already been uploaded to this Repository:
  - eGate Enterprise Designer (**eGate.sar**) 5.1.2
  - eInsight Business Process Manager (**eInsight.sar**) 5.1.2
  - Batch eWay (**BatcheWay.sar**) 5.1.2
  - File eWay (**FileeWay.sar**) 5.1.2
  - LDAP eWay (**LDAPeWay.sar**) 5.1.2
  - HTTPS eWay (**HTTPeWay.sar**) 5.1.2
  - Oracle eWay (**OracleeWay.sar**) 5.1.2

Note: *You need not install eInsight and eWays as a completely separate process; in other words, you can stage them in combination with eXchange.*

To upload product files to the Repository

1. On a Windows machine, start a Web browser and point it at the machine and port (usually 12001) 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:

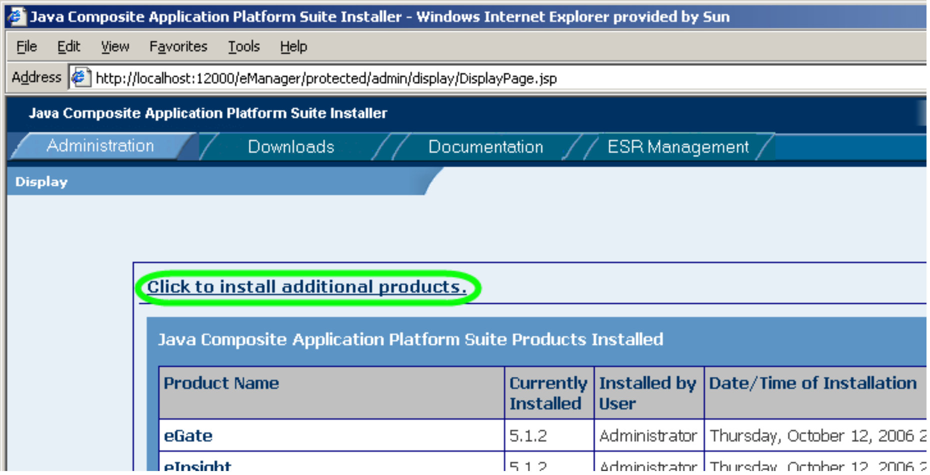
```
http://localhost:12001
http://serv1234.company.com:22001
```

2 On the Suite Installer's **Java CAPS Login** page, enter your username and password.

Note: *The following steps assume you have already uploaded eGate 5.1.2.*

3 When the Suite Installer displays the **Administration** tab, click the link to install additional products. See Figure 8.

Figure 8 Suite Installer: Installing Additional Products


Java Composite Application Platform Suite Installer - Windows Internet Explorer provided by Sun
File Edit View Favorites Tools Help
Address
Java Composite Application Platform Suite Installer
Administration
Downloads
Documentation
ESR Management
Display

http://localhost:12000/eManager/protected/admin/display/DisplayPage.jsp

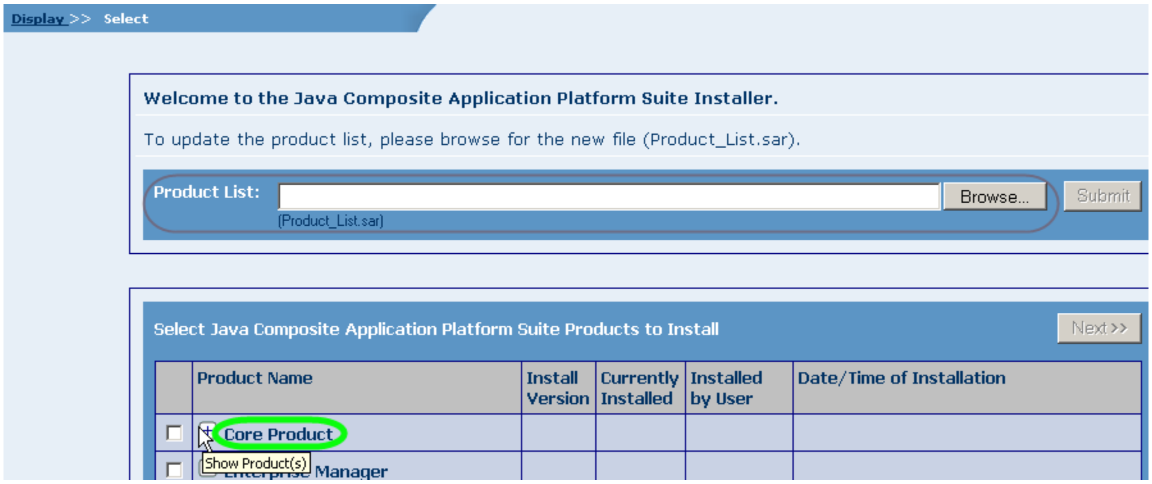
Click to install additional products.

Java Composite Application Platform Suite Products Installed

Product Name	Currently Installed	Installed by User	Date/Time of Installation
eGate	5.1.2	Administrator	Thursday, October 12, 2006 2
eInsight	5.1.2	Administrator	Thursday, October 12, 2006 2

4 In the Display >> Select page open the **Core Products** category; see Figure 9. Depending on the previous installation, eXchange might not appear in the Core Products list; if it is not shown, use the controls near the top of the page to browse to the correct **Product_List.sar** file and submit it.

Figure 9 Suite Installer: Opening the List of Core Products



Welcome to the Java Composite Application Platform Suite Installer.

To update the product list, please browse for the new file (Product_List.sar).

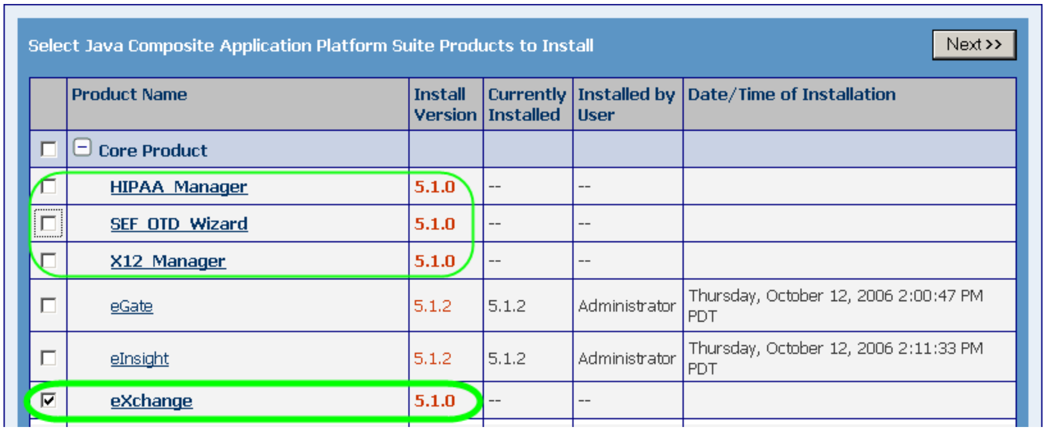
Product List: (Product_List.sar)

Select Java Composite Application Platform Suite Products to Install

	Product Name	Install Version	Currently Installed	Installed by User	Date/Time of Installation
□	Core Product				
□	Enterprise Manager				

5 In the list of core products, select eXchange. See Figure 10.

Figure 10 Suite Installer: List of Core Products Showing eXchange



Select Java Composite Application Platform Suite Products to Install

	Product Name	Install Version	Currently Installed	Installed by User	Date/Time of Installation
□	⊟ Core Product				
□	HIPAA_Manager	5.1.0	--	--	
□	SEF_OTD_Wizard	5.1.0	--	--	
□	X12_Manager	5.1.0	--	--	
□	eGate	5.1.2	5.1.2	Administrator	Thursday, October 12, 2006 2:00:47 PM PDT
□	eInsight	5.1.2	5.1.2	Administrator	Thursday, October 12, 2006 2:11:33 PM PDT
☑	eXchange	5.1.0	--	--	

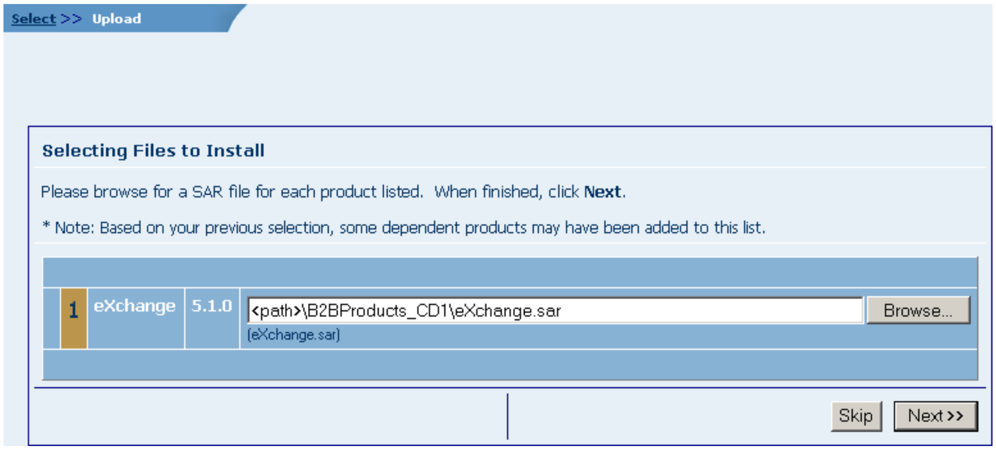
6 Select other core B2B products you want to install, such as the SEF OTD Wizard and any protocol managers for which you are licensed. Optionally, you can also open other categories to select non-core products. For example:

- OTD libraries and protocol managers are often installed along with eXchange.
- If you have not previously installed a required eWay, or a Logical Host, or Enterprise Designer, you can do so now.
- To access the documentation and sample Projects for eXchange and other products, select the appropriate items under the **Documentation** category.

7 When you have selected all the products you want to install, click the **Next** button.

8 In the Select >> Upload page (see Figure 11), browse to and select **eXchange.sar**. Repeat with each of the other SAR files you want to install. When done, click **Next**.

Figure 11 Suite Installer: Selecting Files to Install



Selecting Files to Install

Please browse for a SAR file for each product listed. When finished, click **Next**.

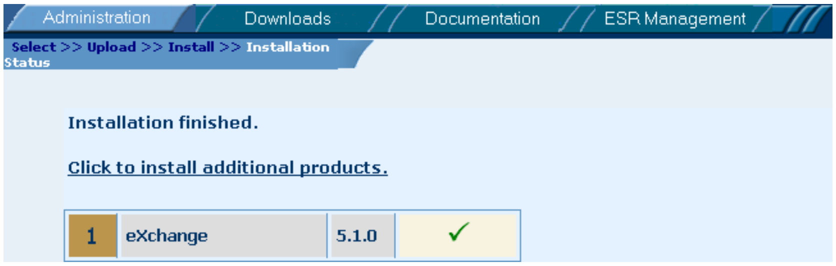
* Note: Based on your previous selection, some dependent products may have been added to this list.

1	eXchange	5.1.0	<path>\B2BProducts_CD1\eXchange.sar (eXchange.sar)

Note: *eXchange.sar is quite large, and requires some time to install.*

Result: The SAR files are loaded to the Repository, and the Select >> Upload >> Install page gives you the opportunity to return to step 3 to install additional products. See Figure 12.

Figure 12 Suite Installer: Successful Installation of eXchange.sar



Installation finished.

Click to install additional products.

1	eXchange	5.1.0	✓

While the Suite Installer is running, you can now do any of the following, as needed:

- Use the **Administration** tab to install additional products and documentation.
- Use the **Downloads** tab to download additional components. For example, all core design work requires Enterprise Designer, and all runtime requires a Logical Host. The sample assumes you have also downloaded Enterprise Manager.
- Use the **Documentation** tab to access documentation files and samples you have installed.

3.4.2 Refreshing Enterprise Designer

The following steps are needed only if you have uploaded (or re-uploaded) a 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 “eGate 5.1.2” besides “**Base ESR**”, you need to take the following steps.

Before you begin

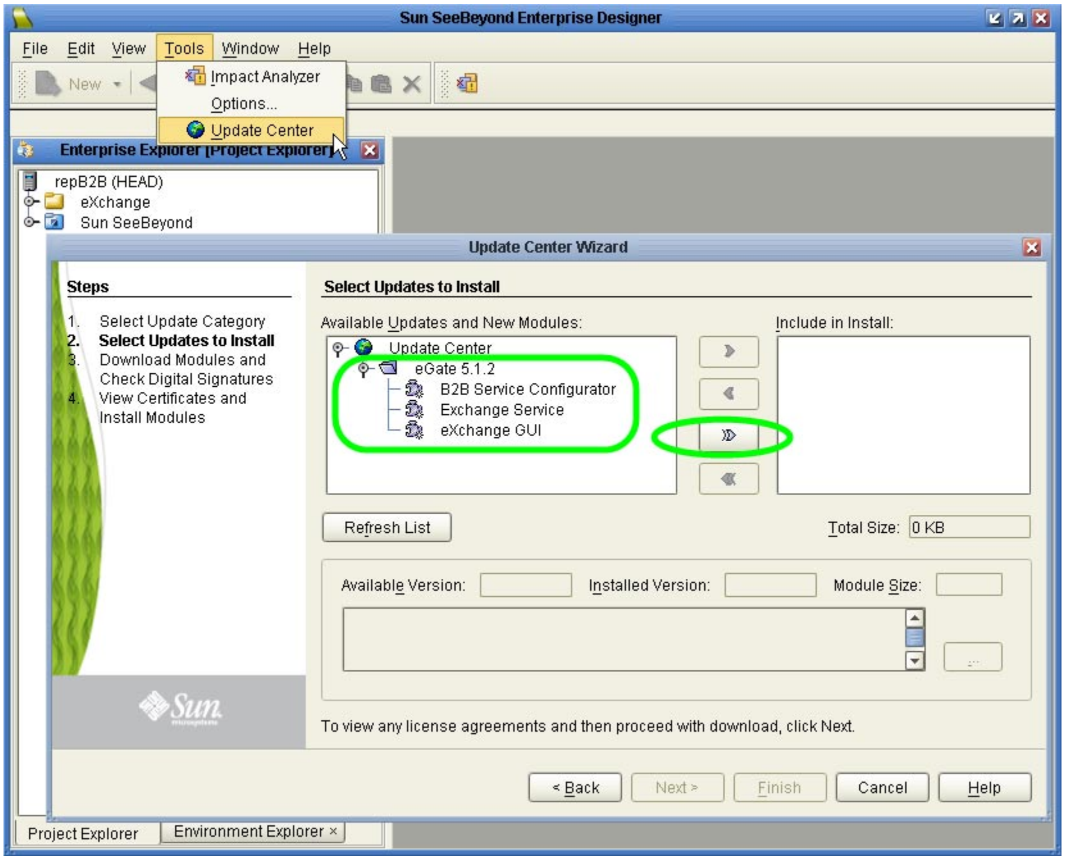
- You must have already downloaded and installed Enterprise Designer.
- A Repository server must be running on the machine where you uploaded the eXchange product files.

To refresh an existing installation of Enterprise Designer

1. Start Enterprise Designer.
2. On the **Tools** menu, click **Update Center**.
3. In the Update Center Wizard, select Check for Available Updates and click **Next**.

   The Update Center shows a list of components ready for updating. See Figure 13.

Figure 13 Update Center Wizard: Select Modules to Install



Steps

1. Select Update Category
2. **Select Updates to Install**
3. Download Modules and Check Digital Signatures
4. View Certificates and Install Modules

Select Updates to Install

Total Size: 0 KB

To view any license agreements and then proceed with download, click Next.

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.*

4 Click **Add All** (the button with a double chevron pointing to the right).

All modules move from the Available/New pane to the **Include in Install** pane.

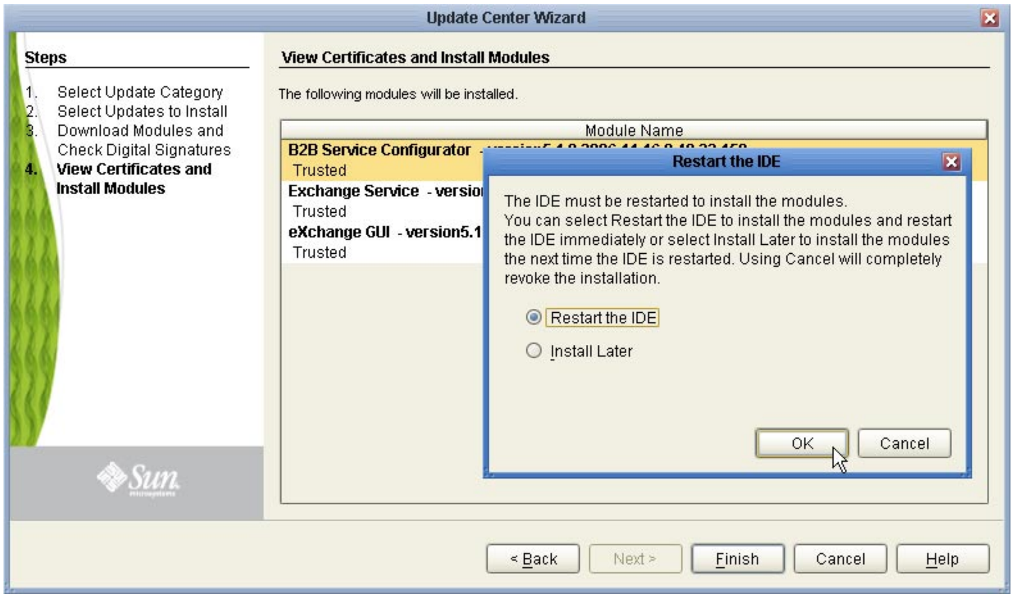
5 Click **Next** and, in the next window, click **Accept** to accept the license agreement.

6 When the progress bars indicate the download has ended, click **Next**.

7 Review the certificates and installed modules, and then click **Finish**.

8 When prompted to restart Enterprise Designer, click **OK**. See Figure 14.

Figure 14 Update Center Wizard: Restart Enterprise Designer


Update Center Wizard
Steps
1. Select Update Category
2. Select Updates to Install
3. Download Modules and Check Digital Signatures
4. View Certificates and Install Modules
Sun microsystems
View Certificates and Install Modules
Module Name
Restart the IDE
OK
Cancel
< Back
Next >
Finish
Cancel
Help

The following modules will be installed.

Module Name
B2B Service Configurator - Trusted
Exchange Service - versio Trusted
eXchange GUI - version5.1 Trusted

The IDE must be restarted to install the modules.
You can select Restart the IDE to install the modules and restart the IDE immediately or select Install Later to install the modules the next time the IDE is restarted. Using Cancel will completely revoke the installation.

● Restart the IDE

○ Install Later

When Enterprise Designer restarts, the installation of eXchange Integrator is complete, and you can use all eXchange tools that require the Enterprise Designer framework.

3.5 After You Install

After you finish installing eXchange, the following additional steps are needed:

- *First-time installation of eXchange:* You must configure an LDAP-compliant directory server to hold Trading Partner information. In section 3.6 (**"LDAP Server"**), follow all necessary steps. For additional information on obtaining, installing, and viewing LDAP Servers, see **Appendix A "Obtaining and Setting Up an LDAP Server" on page 178**.
- *First-time installation of eXchange:* You must create an eXchange database schema and configure a database instance. In section 3.7 (**"Database Scripts"**), follow all necessary steps.
- *For persistence and monitoring:* To use the optional run-time recoverability database schema, you must set up a separate eInsight database instance as described in **"Setting Up the Database Schema for eInsight Engine" on page 154**.

3.6 LDAP Server

eXchange requires communication with an LDAP-compliant directory server (usually known by the shorthand term "LDAP server") to store information on B2B Hosts and Trading Partners.

About LDAP, Directory Services, Servers, and Clients

LDAP (Lightweight Directory Access Protocol) is an Internet protocol for accessing information directories. LDAP runs over TCP/IP and allows clients to access different directory services based on entries. It makes the entries, along with their attributes and values, available to users and other applications, on a controlled-access basis.

A directory service is a distributed database application designed to manage the entries and attributes in a directory. A directory service also makes the entries and attributes available to users and other applications. OpenLDAP server is an example of a directory service. Other directory services include Sun Java™ Directory Service and Microsoft Active Directory.

A directory client accesses a directory service using the LDAP protocol. A directory client may use one of several client APIs available in order to access the directory service.

- *If you do **not** already have an LDAP server:* Download, install, and set up an LDAP server. Sun Java™ System Directory Server 5.2 is recommended. After it is installed and set up (see **Appendix A**), follow all steps below.
- *Otherwise:* Have your LDAP administrator configure the LDAP server as noted immediately below, and then install the eXchange schema using the steps in **procedure on page 37** (or their equivalent on your server).

To configure the Directory Server

- Provide values appropriate for your site (see **Appendix A**; or see eGate Integrator System Administration Guide chapter 13, "LDAP Integration"). The following are supplied for illustrative purposes:

Table 2 Sample Configuration Values for Directory Server

LDAP Item	Illustrative Value	Notes
Server suffix (Root DN)	dc=*host*,dc=*com*	Change domain components appropriately for your domain.
Server bind URL[a]	ldap://*host.sun.com:1389*/ou=*B2B1*,ou=*myName*	Change appropriately[b].
Directory Manager DN	cn=*Manager*,dc=*sun*,dc=*com*	Change common name and domain components appropriately.
Password	*password*	Change to an appropriate value.[c]
	Do not use the values shown in this column; instead, supply values appropriate for your own Directory Server. If necessary, ask your LDAP administrator for assistance.	

[a] If the bind URL is left blank, no filtering is done against bind URL; that is, all connections are accepted.
[b] Best practice is to create a separate organizational unit (**ou**) for storing B2B Host information. If the same B2B Host is shared by multiple users, it can be helpful to create a distinct **ou** value for each user. This is not mandatory, but it provides a convenient way of aggregating and distinguishing information generated.
[c] TLS/SSL is not required.

To install the eXchange LDAP schema to the Sun Java System Directory Server

Before you begin: If you do not already have an existing LDAP server, follow the procedures in **Appendix A** to download and set up a new LDAP server before continuing with the steps below.

1. Shut down the LDAP server.
2. Change directories to one of the following locations:
   - If installed from SunONE:

     `Sun\MPS\slapd-<machine_name>\config\schema`
   - If installed from JES:

     `Sun\MPS\slapd-<machine_name>\config\schema`
3. Move the pre-existing file **28pilot.ldif** to a backup directory.
4. Copy the following files into the current directory.

```
90eXchangeCore.ldif
91eXchange.ldif
92smeks.ldif
```

5. Restart the LDAP server.

3.7 Database Scripts

The **oracle510.zip** file contains scripts for creating a database instance that uses the eXchange database schema. This eXchange database is required; it collects and persists data about messages and delivery history, and it provides information and control over duplicate-checking, batching, and resending. The usual name of the database schema is the default: **eXchange**

The areas to be configured are:

- **Creating and Configuring the eXchange Database Instance** on page 38
- **Extracting, Customizing, and Running Database Setup Scripts** on page 40
- **Running Database Scripts to Set Up the eXchange Database** on page 42

Do not confuse the eXchange database schema (required) with the database schema for the eInsight engine (optional): The eInsight engine allows you to collect and persist data from your business processes; because the data is persisted, you can also use Enterprise Manager to monitor business processes even if logical or physical components are shut down and restarted.

To configure BPs to use the eInsight engine for persistence and monitoring, see **Chapter 9 "Persistence and Monitoring" on page 153**. Areas to be configured are:

3.7.1 Creating and Configuring the eXchange Database Instance

Before you begin: You need to have already created an Oracle database instance with an entry in the **tnsnames.ora** file. Your TNSlistener service must be running, and you will need to know the name of the database instance (default: **eXchange**) and to temporarily use the system username/password (default: **sys/manager** or **system/manager**).

If you have never installed an Oracle database, ask your Oracle database administrator for help. The following constitutes a brief reminder of how to use the Oracle 9*i* wizard.

To create a new database instance for eXchange

1. ("Operations"): Choose **Create a database**.
2. ("Database Templates"): Choose **New Database**.
3. ("Database Identification"): Enter (for example) **eXchange**
4. ("Database Features"): Deselect all checkboxes and reply **Yes** to all prompts.
5. ("Database Connection Options"): Choose **Dedicated** [...].
6. ("Initialization Parameters"): Keep all values unchanged.
7. ("Database Storage"): Under Datafiles, click \{DB_Name}**undotbs01.dbf** (the fifth entry). In the **General** tab, reduce **File Size** from 200 to 100.
8. ("Creation Options"): Choose **Create Database**, and then click Finish.

Modifying the init.ora File for the eXchange Database

If you create a new database, you must increase the **open_cursors** parameter for the eXchange database to a value of 500.

- Some versions of Oracle allow you to do this by using a text editor to modify the **init.ora** file; see the procedure immediately below.
- Other versions of Oracle require you to use the configuration utility; see **procedure on page 39**.

To edit the value of open_cursors in the init.ora file for the eXchange database

1. Navigate to <*Oracle home*>**\admin**<*eXchange database name*>**\pfile**. For example:

   ```
   cd C:\oracle\admin\exchange\pfile
   ```

2. Use a text editor to open the **init.ora** file in this folder. For example:

   ```
   notepad init.ora
   ```

3. Search for the text **open_cursors**; if not found, add a new line. Edit the line so that it reads as follows:

   ```
   open_cursors = 500
   ```

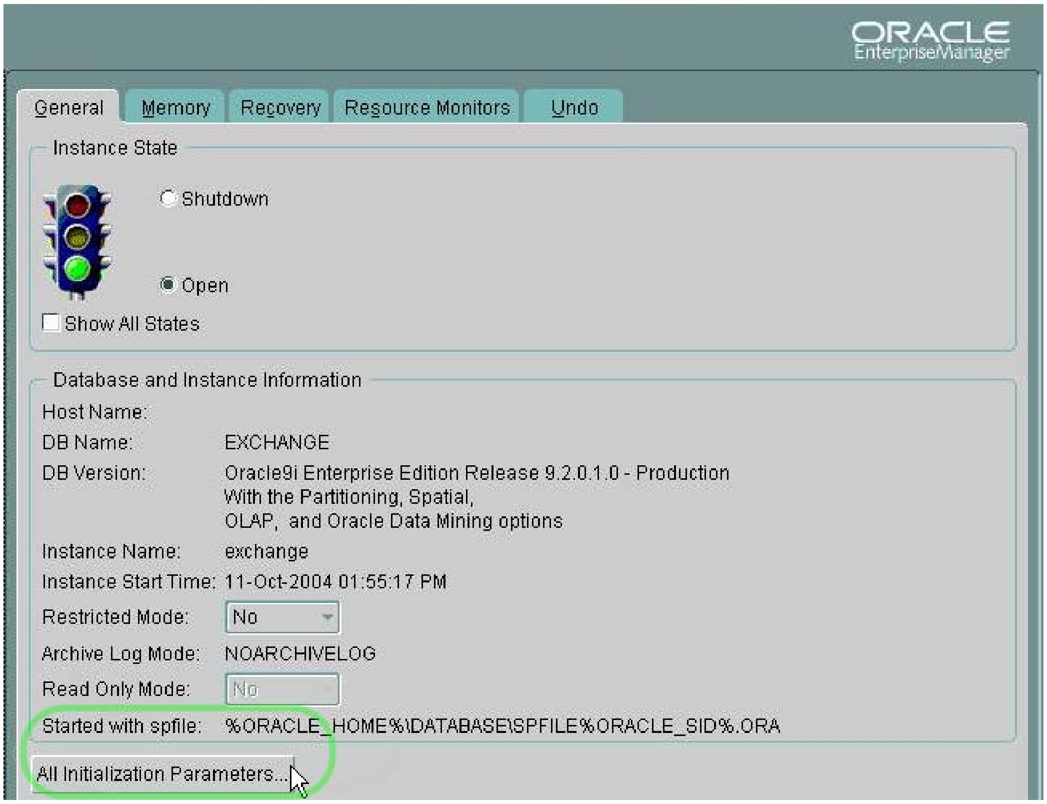
4. Save the file.
5. Restart the database.

Next: Continue with the steps in **"Extracting, Customizing, and Running Database Setup Scripts" on page 40**; at this release, you must extract and run database scripts whether you are installing from scratch or upgrading a previous release of eXchange.

To configure the value of open_cursors in the eXchange database

1. Start the Oracle configuration utility and open the eXchange database.
2. Navigate to Databases > *(EXCHANGE...)* > Instance > Configuration.
3. In the General tab, at the bottom, verify the "Started with spfile" parameter has a value such as %ORACLE_HOME%\DATABASE\SPFILE%ORACLE_SID%.ORA.
4. Click the **All Initialization Parameters** button. See Figure 15.

Figure 15 Oracle Configuration of Initialization Parameters: "General" Tab



Instance State

○ Shutdown

● Open

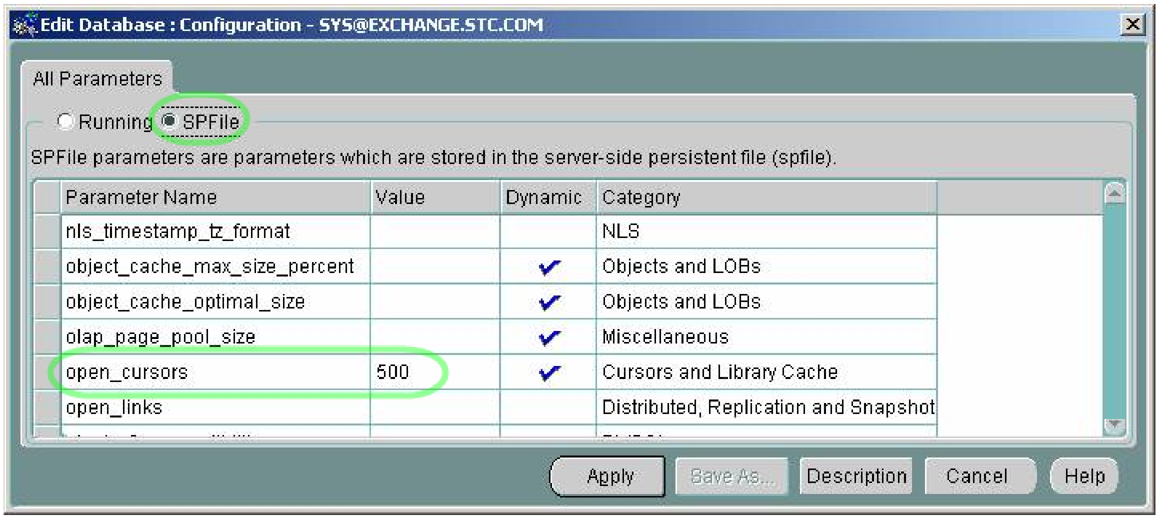
□ Show All States

Database and Instance Information

Host Name:	
DB Name:	EXCHANGE
DB Version:	Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production With the Partitioning, Spatial, OLAP, and Oracle Data Mining options
Instance Name:	exchange
Instance Start Time:	11-Oct-2004 01:55:17 PM
Restricted Mode:	No
Archive Log Mode:	NOARCHIVELOG
Read Only Mode:	No
Started with spfile:	%ORACLE_HOME%\DATABASE\SPFILE%ORACLE_SID%.ORA

5. In the "Edit Database: Configuration [...]" dialog box, click the SPFile option button, scroll to the **open_cursors** parameter, and change its value to 500. See Figure 15.

Figure 16 Oracle Configuration (SPFile) Modifying the Value for “open_cursors”



○ Running ● SPFile

SPFile parameters are parameters which are stored in the server-side persistent file (spfile).

Parameter Name	Value	Dynamic	Category
nls_timestamp_tz_format			NLS
object_cache_max_size_percent		✓	Objects and LOBs
object_cache_optimal_size		✓	Objects and LOBs
olap_page_pool_size		✓	Miscellaneous
open_cursors	500	✓	Cursors and Library Cache
open_links			Distributed, Replication and Snapshot

6 Click the **Apply** button. In response to the prompt (“Would you like to apply this change to the current database?”), click **Yes**.

Next: To set up tables and users for the eXchange database, you will extract the files supplied in the Database Scripts project folder, edit the scripts to reflect your system environment, and then run the scripts to set up and initialize the database.

3.7.2 Extracting, Customizing, and Running Database Setup Scripts

Note: *Do not skip this section. To use eXchange, you must extract and eventually run the* ***createdb*** *script to set up the eXchange database.*

What scripts are supplied, and what do they do?

eXchange supplies the file **oracle510.zip** in the Project Explorer tree under **SeeBeyond > eXchange** > **Download Scripts**. The oracle510.zip file contains a collection of command scripts (**.cmd** files) and SQL scripts (**.sql** files).

You install the eXchange schema on the database by doing one of the following:

- Edit **setenv.cmd** so it reflects your system environment (see **procedure on page 42**) and then run the two other command scripts (see **procedure on page 42**); *—or—*
- Run the SQL scripts directly, supplying system information at run time; see **procedure on page 43**.

Assumptions

The scripts assume they are run on a machine whose command path includes sqlplus. The network\admin\tnsnames.ora file must include a stanza such as the following:

```
eXchange_myOracleHostname.domain
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = TCP)
```

```
            (HOST = myOracleHostname)
            (PORT = 1521))
            ...
        ) )
      (CONNECT_DATA =
        (SID = ORCL)
        ...
    ) )
```

Some scripts and samples assume defaults or supply values as shown in Table 3.

Table 3 Typical, Default, or Presupplied Values for eXchange Database Setup

Item	Typical or presupplied value
UserID/password combination for Oracle system	**sys/manager** or **system/manager**
Name of eXchange database instance (see **procedure on page 38**, step 3)	**eXchange**
SID (also called service ID, servicename, or database name)	**ORCL** (or not presupplied)
TNS name	(not presupplied)
UserID/password for administering the eXchange database	**ex_admin/ex_admin**
UserID/password for first generic eXchange end user	**ex510a/ex510a**
UserID/password for second generic eXchange end user	**ex510b/ex510b**

To extract the scripts

1. In Enterprise Explorer, in the project tree, expand the following folders: **SeeBeyond > eXchange > Download Scripts**
2. Right-click **oracle510.zip** and, on the popup context menu, click **Export**; then use the **Save** dialog box to save the file to a local directory, such as **C:\JC512\Exported\Oracle510\**.
3. Extract the files in **oracle510.zip** into this local directory, yielding:
   - CleanTrackData.sql
   - Cleanup.sql
   - cleanup_coreServices_tables.sql
   - create_coreServices_tables.sql
   - createdb.cmd
   - createdb.sql
   - createtablespaces.cmd
   - createtablespaces.sql
   - createuser.sql
   - eXchange50Runtime.sql
   - in_user_seq.sql
   - setenv.cmd

To edit the setenv command script

1. Open a command prompt and change directories to the local directory where you saved the scripts in the previous procedure.
2. Use a text editor to edit the as-supplied version of **setenv.cmd**:

```
@REM SET YOUR DATABASE CONNECTION INFORMATION HERE
*
echo * This file should be edited to use appropriate
echo * database connection settings. *
echo * SETENV.CMD

@REM TNS_NAME
@set TNS_NAME=< TNS NAME >

@REM ORACLE_SID
@set ORACLE_SID=< SID >

@REM Oracle system login password
@set SYSTEMPWD=< PWD >

@set USERID=ex_admin
@set USERPWD=ex_admin
```

3. Supply the appropriate values for TNS_NAME, ORACLE_SID, and SYSTEMPWD. For example:

```
@set TNS_NAME=eXchange_myOracleHostname
@set ORACLE_SID=ORCL
@set SYSTEMPWD=manager
@set USERID=ex_admin
@set USERPWD=ex_admin
```

4. If your Oracle location is not **c:\oracle\oradata**, or if your database instance name (SID) is other then **eXchange**, then open the **createtablespaces.sql** file and make the appropriate change or changes in the first line.

Important: *The database user who runs the SQL scripts must have permission to create tables.*

Running Database Scripts to Set Up the eXchange Database

You install the eXchange schema on the database by doing one of the following:

- Edit **setenv.cmd** so it reflects your system environment (see **procedure on page 42**) and then run the two other command scripts (see **procedure on page 42**); *—or—*
- Run the SQL scripts directly, supplying system information at run time; see **procedure on page 43**.

To run the command scripts that call SQL scripts to install the schema

1. Open a command prompt and change directories to the local directory where you saved the **.cmd** scripts in the previous procedure. (It is assumed you have already edited **setenv.cmd** appropriately, and **createtablespaces.sql** if necessary.)
2. Enter the following command:

```
createtablespaces
```

The script starts SQL*Plus, invokes an SQL script to create table spaces, and ends.

3 Enter the following command:

```
createdb
```

The script starts SQL*Plus and invokes an SQL script to create a new user entry:

- In response to the first prompt, supply an end username, such as: **ex510A**
- In response to the prompt, supply a password for this end user, such as: **ex510A**

The script creates a new user/password combination, invokes other SQL scripts to update the database instance, and then ends.

4 Repeat step 3 as needed to create other user/password entries for eXchange users.

Result: You have installed the eXchange schema onto the eXchange database instance and created user/password combinations. End users can create Oracle OTDs based on this database, and can use it for message tracking and other eXchange functions.

To directly run the SQL scripts that install the schema

Note: *These steps are an alternative to the command scripts described in the previous procedure. Do not use both procedures.*

1 Open a command prompt and change directories to the local directory where you saved the **.sql** scripts in the previous procedure.

2 If your Oracle location is not **c:\oracle\oradata**, or if your database instance name (SID) is other then **eXchange**, then open the **createtablespaces.sql** file and make the appropriate change or changes in the first line.

3 Enter the following SQL*Plus command:

```
<path>\sqlplus system/<SYSTEMPWD>@<TNSNAME> @createtablespaces.sql
```

where:

<SYSTEMPWD> is the password for the **system** login ID

<TNSNAME> is the name of the Oracle database instance you created for eXchange.

Here are two examples of valid commands, depending on the password and name:

```
C:\oracle\ora92\bin\sqlplus system/manager1@eX50 @createtablespaces.sql
sqlplus system/oraclePW@eXchange @createtablespaces.sql
```

When this finishes, you will have created new tablespaces.

4 In the command prompt, enter the following SQL*Plus command:

```
sqlplus system/<SYSTEMPWD>@<TNSNAME> @createuser.sql
```

where, as before, *<SYSTEMPWD>* is the password for the **system** login ID and *<TNSNAME>* is the name of the Oracle database instance you created for eXchange.

Here is an example of a valid command:

```
\oracle\ora92\bin\sqlplus system/myPassWd@eX505DB @createuser.sql
```

5 In response to the system prompt for value #1, enter the username. For example: **ex_admin**

6 In response to the system prompt for value #2, enter the password. For example: **ex_admin**

7 Repeat steps 5 and 6 as needed to create user/password entries for eXchange users.

8 After you run the createtablespaces and createuser SQL scripts, there is one more. In the command prompt, enter the following SQL*Plus command:

```
sqlplus ex_admin/ex_admin@<TNSNAME> @createdb.sql
```

where, as before, *TNSNAME* is the name of the eXchange Oracle database instance, and your eXchange administrator username and password are both **ex_admin**.

Result: After the **createdb.sql** script ends, you are done—you do not need to run any further SQL scripts. The system populates the tables, and you are ready to use the database instance as your eXchange database. End users can create Oracle OTDs based on this database, and can use it for message tracking and other eXchange functions.

To reinitialize the database

- Open a command prompt, change directories to the location where you extracted the **.sql** scripts from **oracle510.zip** (see **"Extracting, Customizing, and Running Database Setup Scripts" on page 40**), and enter the following SQL*Plus command:

```
sqlplus ex_admin/ex_admin@<TNSNAME> @createdb.sql
```

where, as before, <*TNSNAME*> is the name of the eXchange Oracle database instance, and your eXchange administrator username and password are both assumed to be **ex_admin**.

Chapter 4

eXchange Features

This chapter provides brief descriptions of components packaged with eXchange and the B2B Suite.

What's in This Chapter

4.1 Project Tree Organization

Initial installation of eXchange populates the project tree with the following folders:

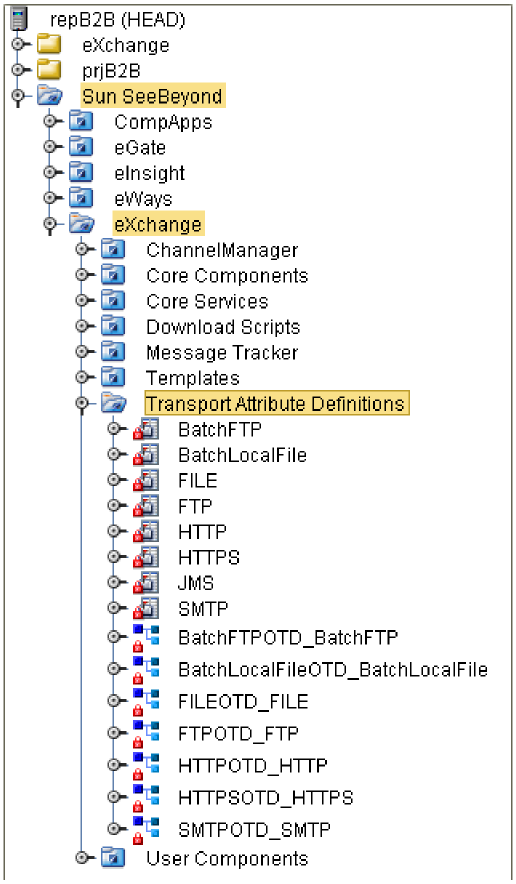
- eXchange > **Deployment**—a preconfigured project containing connectivity maps, containers for standard servers (BatchFTP, BatchLocalFile, ExConfigSvc, HTTP, and eXchangeService itself), and eXchange-specific JMS topics.
- Sun SeeBeyond > CompApps > **Core Services**—contains JCDs and OTDs for batch services and reliable messaging, as well as associated SQL scripts and JAR files.
- Sun SeeBeyond > eInsight > **(various)**—By virtue of having installed eInsight, this folder contains scripts for setting up the eInsight engine database schema, used for persistence and monitoring; for more information, see **Chapter 9 on page 153**.
- Sun SeeBeyond > eWays > **(various)**—By virtue of having installed the five required eWays, this folder contains OTDs for Batch, File, HTTP, and LDAP; it also contains a placeholder for the B2B Configurator, called ExConfigSvc.
- Sun SeeBeyond > **eXchange**—contains all of the other components used by eXchange, including: Channel Manager; core components (BPs, collaborations, OTDs, and the ePM application), core Web services (selectors and handlers), scripts for LDAP and Oracle, the message-tracker application, templates of the as-shipped versions of customizable components, transport attribute definitions, and additional "user" components for error handling and delivery channels.

4.2 Transport Attribute Definitions

Transport attribute definitions provide the metadata required at the transport layer. (In this context, "metadata" means the *categories* of information, not any actual values.) Once a transport attributes definition has been included in a B2B host, it is exposed to ePM so that specific values can be supplied for specific trading partner configurations.

The Sun SeeBeyond > eXchange > **Transport Attribute Definitions** folder (see Figure 17) contains presupplied transport attribute definitions (TADs) and corresponding OTDs.

Figure 17 Sun SeeBeyond > eXchange > Transport Attribute Definitions Folder



Overview

Different transport protocols require different types of attributes; for example, HTTP requires little more than a URL, but FTP requires a username, password, hostname, port, path, and file pattern, and possibly other attributes as well. For this reason, the metadata for HTTP-based and FTP-based TADs are quite different. When a TAD is referenced by a delivery channel, its attributes govern the appearance and behavior of ePM for users who supply values for that channel.

At run time, a TAD’s metadata is made available to the application through the two methods of its associated OTD: **unmarshal** parses an inbound stream into an internal data structure, and **marshal** serializes the internal data into a linear outbound stream.

All TADs define their metadata using the format shown in Table 4.

Table 4 Metadata for All Transport Attribute Definitions

Field Name	Explanation
Name	The (internal) parameter name. Used programmatically; never seen in ePM.
Display	The parameter label as seen by the ePM user.
Type	Data type. Used programmatically; never seen in ePM.
Required	Checkbox governing whether a value must be supplied in ePM. If yes, ePM displays an red asterisk to signal the user that this is a required value.
Direction	FromPartner, ToPartner, or Both. Used programmatically.
Default	The value supplied before the ePM user takes action, or takes no action.
List of Values	Items to display in a drop-down list for the ePM user to choose from
Fixed	(not used in any of presupplied TADs)
Format String	(not used in any of presupplied TADs)

4.2.1 BatchFTP

The Batch eWay uses the BatchFTP transport attributes definition to read from a remote file or write to a remote file. When designating a pattern of files to be read, the * (asterisk) is a wildcard meaning “zero or more characters.”

Table 5 lists the attributes of the BatchFTP TAD and corresponding OTD.

Table 5 Attributes for the BatchFTP Transport Attributes Definition

Name	Display	Type	Req?	Direction	Default	List
Append	Append	Boolean	No	Both		
HostName	HostName	String	Yes	Both		
ServerPort	ServerPort	Integer	No	Both		
CommandConnectionTimeout	(same as Name)	Integer	No	Both		
DataConnectionTimeout	(same as Name)	Integer	No	Both		
DirectoryListingStyle	[...]	String	No	Both		
ClientClassName		String	No	Both		
ProviderClassName		String	No	Both		
Mode		String	No	Both		
UsePASV		Boolean	No	Both		
UserName		String	No	Both		

Table 5 Attributes for the BatchFTP Transport Attributes Definition (Continued)

Name	Display	Type	Req?	Direction	Default	List
Password		Password	No	Both		
UserPropertyFile		String	No	Both		
TargetDirectoryName		String	Yes	Both		
TargetDirectoryNameIsPattern		Boolean	No	Both		
TargetFileName		String	Yes	Both		
TargetFileNameIsPattern		Boolean	No	Both		
MaxSequenceNumber		Integer	No	Both		
StartingSequenceNumber		Integer	No	Both		
PreDirectoryName		String	No	Both		
PreDirectoryNameIsPattern		Boolean	No	Both		
PreFileName		String	No	Both		
PreFileNameIsPattern		Boolean	No	Both		
PreTransferCommand		String	No	Both		
PreTransferRawCommands		String	No	Both		
PostDirectoryName		String	No	Both		
PostDirectoryNameIsPattern		Boolean	No	Both		
PostFileName		String	No	Both		
PostFileNameIsPattern		Boolean	No	Both		
PostTransferCommand		String	No	Both		
PostTransferRawCommands		String	No	Both		
ActionOnMalformedCommand		String	No	Both		
IncludeOrderRecordInErrorRecord		Boolean	No	Both		
IncludePayloadInErrorRecord		Boolean	No	Both		
PublishStatusRecordOnError		Boolean	No	Both		
PublishStatusRecordOnSuccess		Boolean	No	Both		
SocksEnabled		Boolean	No	Both		
SocksHostName		String	No	Both		
SocksServerPort		Integer	No	Both		
SocksVersion		Integer	No	Both		
SocksUserName		String	No	Both		
SocksPassword		Password	No	Both		
SshTunnelingEnabled		Boolean	No	Both		
SshListenHost	[...]	String	No	Both		
SshListenPort	(same as Name)	Integer	No	Both		

Table 5 Attributes for the BatchFTP Transport Attributes Definition (Continued)

Name	Display	Type	Req?	Direction	Default	List
SshCommandLine	(same as Name)	String	No	Both		
SshUserName	SshUserName	String	No	Both		
SshPassword	SshPassword	Password	No	Both		

4.2.2 BatchLocalFile

The Batch eWay uses the BatchLocalFile transport attributes definition to read from a local file or write to a local file. When designating a pattern of files to be read, the * (asterisk) is a wildcard meaning "zero or more characters."

Table 6 lists the attributes of the BatchLocalFile TAD and corresponding OTD.

Table 6 Attributes for the BatchLocalFile Transport Attributes Definition

Name	Display	Type	Req?	Direction	Default	List
Append	Append	Boolean	No	Both		
TargetDirectoryName	(same as Name)	String	Yes	Both		
TargetDirectoryNameIsPattern	(same as Name)	Boolean	No	Both		
TargetFileName	[...]	String	Yes	Both		
TargetFileNameIsPattern		Boolean	No	Both		
MaxSequenceNumber		Integer	No	Both		
StartingSequenceNumber		Integer	No	Both		
PreDirectoryName		String	No	Both		
PreDirectoryNameIsPattern		Boolean	No	Both		
PreFileName		String	No	Both		
PreFileNameIsPattern		Boolean	No	Both		
PreTransferCommand		String	No	Both		
PostDirectoryName		String	No	Both		
PostDirectoryNameIsPattern		Boolean	No	Both		
PostFileName		String	No	Both		
PostFileNameIsPattern		Boolean	No	Both		
PostTransferCommand		String	No	Both		
ResumeReadingEnabled		Boolean	No	Both		
ActionOnMalformedCommand		String	No	Both		
IncludeOrderRecordInErrorRecord		Boolean	No	Both		

Table 6 Attributes for the BatchLocalFile Transport Attributes Definition (Continued)

Name	Display	Type	Req?	Direction	Default	List
IncludePayloadInErrorRecord	[...]	Boolean	No	Both		
PublishStatusRecordOnError	(same as Name)	Boolean	No	Both		
PublishStatusRecordOnSuccess	(same as Name)	Boolean	No	Both		

4.2.3 FILE

The File eWay uses the FILE transport attributes definition to read from a file or write to a file. When designating a pattern of files to be read, the * (asterisk) is a wildcard meaning "zero or more characters."

Table 7 lists the attributes of the FILE TAD and corresponding OTD.

Table 7 Attributes for the FILE Transport Attributes Definition

Name	Display	Type	Req?	Direction	Default	List
FilePattern	FilePattern	String	Yes	Both		
Directory	Directory	String	Yes	Both		

4.2.4 FTP

For File Transfer Protocol, the BatchFTP eWay uses the FTP transport attributes definition to read from a file or write to a file in a remote location. When designating a pattern of files to be read, the * (asterisk) is a wildcard meaning "zero or more characters."

Table 8 lists the attributes of the FTP TAD and corresponding OTD.

Table 8 Attributes for the FTP Transport Attributes Definition

Name	Display	Type	Req?	Direction	Default	List
FilePattern	FilePattern	String	Yes	Both		
Directory	Directory	String	Yes	Both		
UserName	UserName	String	Yes	Both		
Password	Password	Password	Yes	Both		
HostName	HostName	String	Yes	Both		
PortNumber	PortNumber	Integer	No	Both		
SocksEnabled	SocksEnabled	Boolean	No	Both	false	
SocksHostName	SocksHostName	String	No	Both		

Table 8 Attributes for the FTP Transport Attributes Definition (Continued)

Name	Display	Type	Req?	Direction	Default	List
SocksUserName	SocksUserName	String	No	Both		
SocksPassword	SocksPassword	Password	No	Both		
SocksServerPort	SocksServerPort	String	No	Both		

4.2.5 HTTP

For Hypertext Transfer Protocol, the HTTP(S) eWay can use the HTTP transport attributes definition to access Web pages. The HTTP TAD has no attributes and has no corresponding OTD.

4.2.6 HTTPS

For Hypertext Transfer Protocol over SSL (Secure Sockets Layer), the HTTP(S) eWay uses the HTTPS transport attributes definition to access secure Web page.

Table 9 lists the attributes of the HTTPS TAD and corresponding OTD.

Table 9 Attributes for the HTTPS Transport Attributes Definition

Name	Display	Type	Req?	Direction	Default	List
ClientCertAlias	ClientCertAlias	String	Yes	Both		
UserName	UserName	String	Yes	Both		
Password	Password	Password	Yes	Both		

Note: *To use HTTPS, two environment components require nondefault settings: in the Integration Server, the Web server configuration must have its "Enable SSL" parameter set to "True", and in the HTTP(S) external, the SSL configuration must supply a value for its "TrustStore" parameter.*

4.2.7 JMS

The JMS transport attributes definition is used to transport data into and out of JMS topics and queues. The JMS TAD has no attributes and has no corresponding OTD.

4.2.8 SMTP

For Simple Mail Transfer Protocol, the e-mail eWay uses the SMTP transport protocol to send and receive e-mail.

Table 10 lists the attributes of the SMTP TAD and corresponding OTD.

Table 10 Attributes for the SMTP Transport Attributes Definition

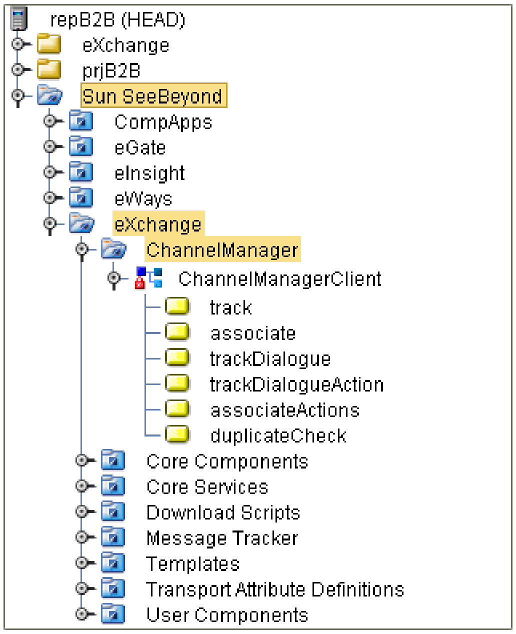
Name	Display	Type	Req?	Direction	Default	List
SenderAddress	SenderAddress	String	Yes	ToPartner		
Host	Host	String	Yes	ToPartner		
PortNumber	PortNumber	String	Yes	ToPartner		
UserName	UserName	String	No	Both		
Password	Password	Password	No	Both		

4.3 Channel Manager

The Channel Manager facility provides several services to access or write information in the eXchange database. It tracks messages and packages, associates responses to requests and tracks them, and retrieves trading partner information.

The Sun SeeBeyond > eXchange > **ChannelManager** folder (see Figure 18) contains the ChannelManagerClient OTD.

Figure 18 Sun SeeBeyond > eXchange > ChannelManager Folder



In this section

- **trackDialogue** on page 55
- **trackDialogueAction** on page 56

associate

ChannelManagerClient.**associate** is used to associate a response to a request. This operation can only be used for message level documents—in other words envelopes, as opposed to business documents.

The service associates the response to the request using a message identifier to tie the two messages to each other.

Table 11 Input Containers for ChannelManagerClient.associate

Name	Description
OrigPkgHdrId	Database ID of the original message
AckPkgHeaderId	Database ID of the acknowledgement message
PkgType	Name of the messaging or packaging envelope used for the message, such as **ISA** or **GS**.
TPId	The database's unique ID for the trading partner; in other words, the foreign key to ex_trading_partner.
MsgUniqId	Unique ID for the message.
ErrorFlag	A value of **Y** signifies that the message contains a "business" type of error: could not decrypt, could not verify signature, and so forth.
ErrorNo	(reserved)
ErrorStr	A description of the error.

Table 12 Output Container for ChannelManagerClient.associate

Name	Description
PkgAssocId	Association ID used to associate the response package to the request package.

associateActions

ChannelManagerClient.**associateActions** is similar to the associate operation, in that it associates a document response to a document request (for example, in X12, a 997 or 855 response to an original 850 request).

Table 13 Input Containers for ChannelManagerClient.associateActions

Name	Description
OrigPkgHdrId	Database ID of the original message
AckPkgHeaderId	Database ID of the acknowledgment message
PkgType	Name of the messaging or packaging envelope used for the message, such as **ISA** or **GS**.
TPId	The database's unique ID for the trading partner; in other words, the foreign key to ex_trading_partner.
MsgUniqId	Unique ID for the message.
ErrorFlag	A value of **Y** signifies that the message contains a "business" type of error: could not decrypt, could not verify signature, and so forth.
ErrorNo	(reserved)
ErrorStr	A description of the error.

Table 14 Output Container for ChannelManagerClient.associateActions

Name	Description
isAssociated	A value of **Y** signifies that an associated action exists.

duplicatecheck

ChannelManagerClient.**duplicatecheck** is used to check for duplicates of a generic inbound or outbound message.

track

ChannelManagerClient.**track** performs a track operation to store the message to the eXchange database

Table 15 Input Containers for ChannelManagerClient.track

Name	Description
Protocol	Name of the protocol being used to handle the message.
ReceiveFlag	A value of **Y** signifies that the request message was inbound.
BufferId	*ebXML only*. Conversation ID.
OrderNumInBuffer	*ebXML only*. Reserved for use in message ordering.
MsgUniqId	Unique ID for the message.
TPId	The database's unique ID for the trading partner; in other words, the foreign key to ex_trading_partner.
OrdMsgId	(not currently used)
Multiple Content	(not currently used)

Table 15 Input Containers for ChannelManagerClient.track (Continued)

Name	Description
PkgType	Name of the messaging or packaging envelope used for the message, such as ISA or GS.
ErrorFlag	A value of **Y** signifies that the message contains a “business” type of error: could not decrypt, could not verify signature, and so forth.
RespRequired	A value of **Y** signifies that a response to this message is required.
MsgBlob	Container for the message payload.
SignedFlag	A value of **Y** signifies that the message is signed.
CompressedFlag	A value of **Y** signifies that the message is compressed.
EncryptedFlag	A value of **Y** signifies that the message is encrypted.
MessageType	Message type for the message, such as **Message** or **Ack**.
Resendable	A value of **Y** signifies that the message can be re-sent.
Service	Service name for the request message for which the response is received.
Action	Action name for the request message for which the response is received.

Table 16 Output Container for ChannelManagerClient.track

Name	Description
MsgHdrId	Message header ID, used for message association.

trackDialogue

ChannelManagerClient.**trackDialogue** is used to write the initial message—that is, the first business document in a conversation—to message tracking. To write subsequent messages in the same conversation, the trackDialogueAction operation is used.

Table 17 Input Containers for ChannelManagerClient.trackDialogue

Name	Description
tpNetworkId	eXchange-generated unique ID identifying the trading partner.
dialogueID	Database-assigned unique ID identifying the business conversation.
dialogueIdentifier	Dialog ID in the message.
serviceName	Name of the messaging service or business service being used to handle the message.
activeFlag	A value of **Y** signifies that the business conversation is active.
Status	Status of the business conversation.
startDate	Timestamp recording when the business conversation initiated.
endDate	Timestamp recording when the business conversation terminated.
protocol	Name of the protocol being used to handle the message.

Table 17 Input Containers for ChannelManagerClient.trackDialogue (Continued)

Name	Description
hostNetworkId	eXchange-generated unique ID identifying the B2B host.
isResponse	A value of **true** signifies that the message is a response to a previous message.

Table 18 Output Containers for ChannelManagerClient.trackDialogue

Name	Description
tpNetworkId	eXchange-generated unique ID identifying the trading partner.
dialogueID	Database-assigned unique ID identifying the business conversation.
dialogueIdentifier	Dialog ID in the message.
serviceName	Name of the messaging service or business service being used to handle the message.

trackDialogueAction

ChannelManagerClient.**trackDialogueAction** also writes to message tracking, but it writes subsequent messagesin a business conversation (after the initial message was written by trackDialogue operation).

Table 19 Input Containers for ChannelManagerClient.trackDialogueAction

Name	Description
messageId	*(deprecated)* Duplicate of actionMessageId
actionName	Name of the messaging action that is processing the message.
receiveFlag	A value of **Y** signifies that it is an inbound message.
resendFlag	A value of **Y** signifies that this is a re-send of the message
sendCount	A value of **Y** signifies that the business conversation is active.
sequenceNum	Status of the business conversation.
referToType	(not used)
actStatus	(not used)
pkgMsgHdrId	Database-assigned unique ID for the message packaging.
msgType	Message type for the message, such as **Message** or **Ack**.
msgEncoding	Encoding to which the message conforms.
compressedFlag	A value of **Y** signifies that the message is compressed.
encryptedFlag	A value of **Y** signifies that the message is encrypted.
envelopedFlag	A value of **Y** signifies that the message is enveloped.
signedFlag	A value of **Y** signifies that the message is signed.

Table 19 Input Containers for ChannelManagerClient.trackDialogueAction (Continued)

Name	Description
msgContent	The payload of the message.
attributeMap	Extended attributes for the message.
isStoreOriginal	*(not used)* A value of **Y** signifies that the original (raw) message is to be stored in the database.
errorFlag	A value of **Y** signifies that the message has an error associated with it.
respRequiredFlag	A value of **Y** signifies that a response is required for the message.
actionMessageId	Message ID.
messageType	A value of **msg** signifies a message; **ack** signifies an acknowledgment.

Table 20 Output Containers for ChannelManagerClient.trackDialogueAction

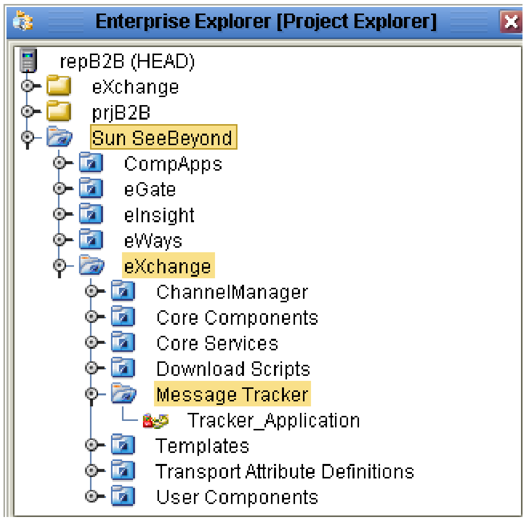
Name	Description
ActionId	ID of the service action (business transaction)
dialogueID	ID of the business dialog.

4.4 Message Tracker

The tracker application is used to record processing, packaging, and error information about messages and acknowledgments as they flow through the eXchange system.

The Sun SeeBeyond > eXchange > **Message Tracker** folder (see Figure 19) contains only one item: the tracker application itself.

Figure 19 eXchange > Message Tracker Folder



Enterprise Explorer [Project Explorer]

```
repB2B (HEAD)
  eXchange
  prjB2B
  Sun SeeBeyond
    CompApps
    eGate
    eInsight
    eWays
    eXchange
      ChannelManager
      Core Components
      Core Services
      Download Scripts
      Message Tracker
        Tracker_Application
      Templates
      Transport Attribute Definitions
      User Components
```

The application is entirely self-contained. All you need to do is connect an instance of the application to the B2B host you want to track and to a well-configured eXchange database. Activating this project generates an eXchange service that can be used in any other project contained in the same Repository.

For information on creating and activating a connectivity map containing a tracker application, see **"Connecting the B2B Host to Oracle and LDAP Externals" on page 84**. For information on using the Message Tracker web client to view tracking information that has been written to the database, see **"Message Tracking" on page 136**.

4.5 B2B Protocols for X12 and HIPAA

If you have licensed and installed any of the protocol manager composite applications, your Sun SeeBeyond > **eXchange > B2B Protocols** folder will contain additional folders. Each additional protocol folder includes a library of collaborations, OTDs, and B2B protocols that are custom-tailored for the specific protocol. The contents of each Sun SeeBeyond > eXchange > **B2B Protocols <*protocol*>** are extensive, and vary according to version. See the corresponding Protocol Manager User's Guides.

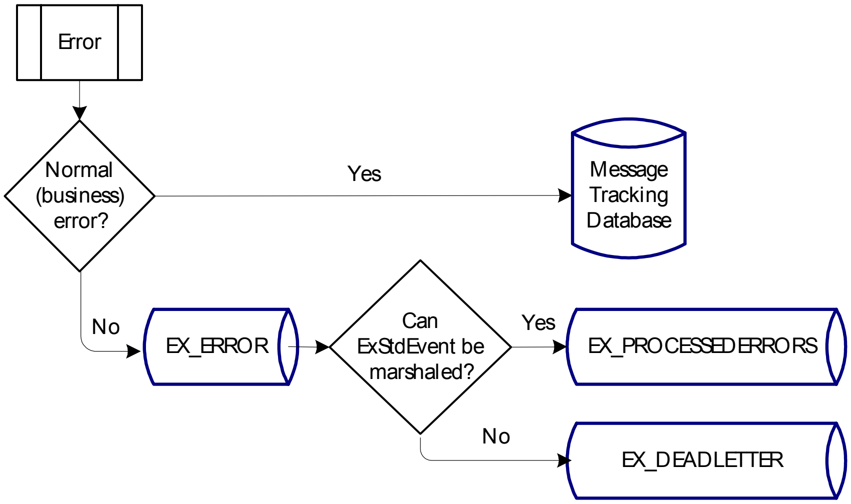
To gain a sense of how the protocol processes and collaborations/OTDs can be used, it is strongly recommended that you download and run the sample implementations. See the user's guide corresponding to the Protocol Manager(s) you have installed.

4.6 Handling Errors

This section describes the behavior of the standard error-handling BP. If you want, you can create a custom error-handling BP and can sort and redirect errors in any way you prefer. Custom error-handlers are outside the scope of this document, and are discussed in the *eXchange Integrator Developer's Guide*.

Figure 20 is a flowchart that illustrates the default error handling.

Figure 20 Error-Handling Flowchart



4.6.1 Message Tracking

Errors that occur at the business protocol level, such as an error within the payload of a single message resulting in a negative acknowledgment (such as an X12 997), or errors within the enveloping layer that can be reported at the messaging level (such as an X12 TA1), are a normal part of B2B conversation, and are not considered faults. Such errors are reported to the message-tracker application associated with the B2B host when it was activated. For information on linking the B2B host to a tracker application, see **"Connecting the B2B Host to Oracle and LDAP Externals" on page 84**.

4.6.2 eXchange Standard Error-Handling Topics

System-level errors, and business errors that cannot be handled by message tracking, are considered faults; eXchange catches such faults and publishes the ExStdEvent—with the error information now contained inside the ExStdEvent—to a standard eXchange topic, EX_ERROR. If further processing is possible, the messages are finally published to the EX_PROCESSEDERRORS topic; otherwise, they are finally published to the EX_DEADLETTER topic.

EX_PROCESSEDERRORS

When EX_ERROR contains a marshallable ExStdEvent message, the message is presumed to contain all necessary troubleshooting details, specified in a clear enough way to allow the fault to be diagnosed. In this case, the ExStdEvent is published to the EX_PROCESSEDERRORS topic.

EX_DEADLETTER

When EX_ERROR contains an ExStdEvent message that cannot be unmarshaled, there is no way to determine the cause of the fault. In this case, the ExStdEvent is published to the EX_DEADLETTER topic.

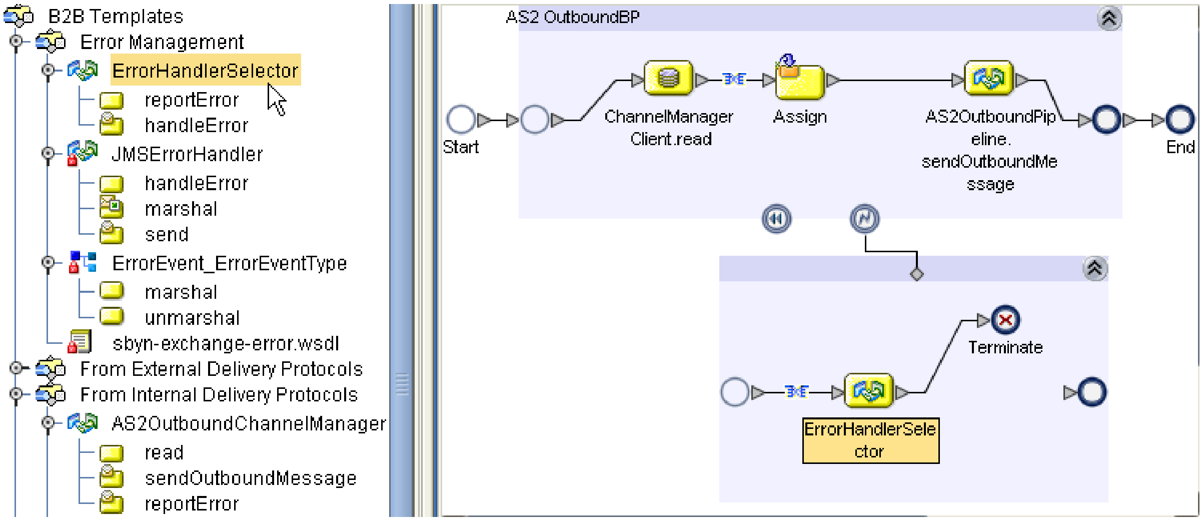
4.6.3 Error Logs

For help in tracking down subtle or persistent errors, the integration server provides extensive log files whose reporting sensitivity can be tuned to various levels (ERROR, INFO, DEBUG). For more information, see the “Monitoring Logs” chapter in the *eGate Integrator System Administration Guide*.

4.6.4 Creating a Nondefault Error Handler

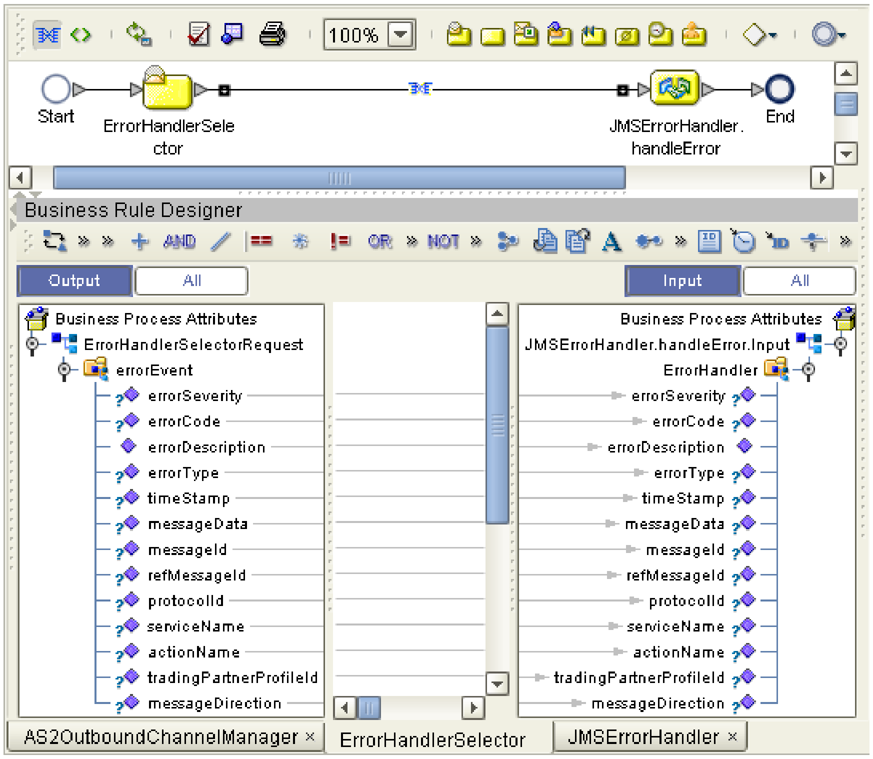
Within each of the standard BPs that provides for error handling, at the lower right margin of the main scope is a Catch Named Exception activity, connecting it to a scope containing an instance of a B2B protocol process: **ErrorHandlerSelector**. See Figure 21.

Figure 21 Template B2B Protocol Process “ErrorHandlerSelector“



If you open the ErrorHandlerSelector BP, you find that by default, it uses JMS for handling errors, mapping fields under the ErrorHandlerSelectorRequest’s **errorEvent** container to the handleError.Input’s **ErrorHandler** container. See Figure 22.

Figure 22 Mapping from ErrorHandlerSelector to ErrorHandler



The purpose of the JMSErrorHandler B2B protocol process is to send error messages to JMS, using the ErrorEvent_ErrorEventType OTD. However, in place of JMS, you could substitute SMTP to e-mail the text of the error message or FTP to write it to a file on a remote server. To create an SMTP-based error handler, for example, you would follow these steps:

1. Export the **sbyn-exchange-error.wsdl** file.
2. In the Error Management folder, create a new B2B protocol process and name it **SMTPErrorHandler** (for example).
3. Open the properties of SMTPErrorHandler and use the **WSDL** tab to load the .wsdl file you exported, the **Partners** tab to add a new partner named ErrorSelector, and the **Business Process Attributes** tab to create a new attribute named ErrorEvent (namespace **urn:sbyn-exchange-err**).
4. Drag activities onto the Protocol Designer canvas for **SMTPErrorHandler**, connect them, and configure business rules in the same way as for JMSErrorHandler.

After you have created a new error handler, you can go back to ErrorHandlerSelector and replace JMSErrorHandler with the new error handler, for example, if you wanted AS2OutboundChannelManager to deliver error messages via SMTP instead of JMS.

In the same way, you can add or modify other components in the B2B Templates folder, making them part of the toolset used by eXchange projects in this Repository.

4.7 SEF OTD Wizard

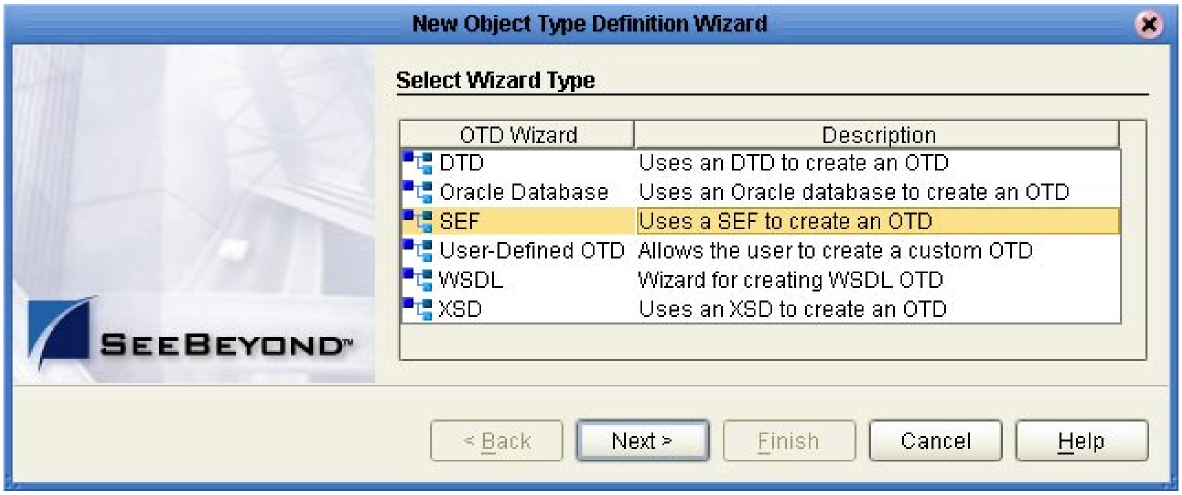
Included with eXchange is the SEF OTD Wizard. The wizard allows you to generate OTDs for ASC X12, HIPAA, UN/EDIFACT, and related standard EDI formats using Standard Exchange Format (SEF) files—***.sef**—as input.

Note: *For detailed information on SEF and **.sef** files, including automated tools that can be used to generate **.sef** files for a variety of EDI standards: contact the Foresight Corporation:* **http://www.foresightcorp.com**

To create a SEF-based OTD

1 Right-click the project and, on the popup context menu, point at **New**, and click **Object Type Definition**. Then, in the Select Wizard Type step, click **SEF** and then click **Next**. See Figure 23.

Figure 23 Selecting SEF from the New Object Type Definition Wizard

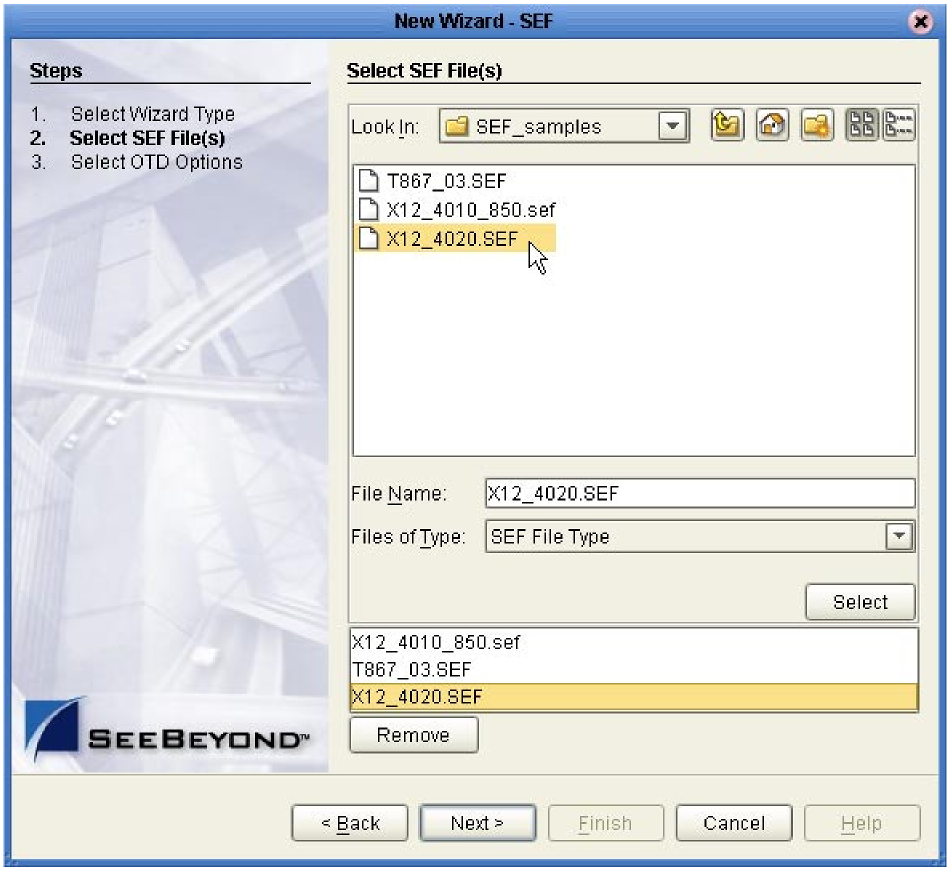


Select Wizard Type

OTD Wizard	Description
DTD	Uses an DTD to create an OTD
Oracle Database	Uses an Oracle database to create an OTD
SEF	Uses a SEF to create an OTD
User-Defined OTD	Allows the user to create a custom OTD
WSDL	Wizard for creating WSDL OTD
XSD	Uses an XSD to create an OTD

2 In step 2 (**Select SEF Files**), navigate to the location(s) of the **.sef** file(s) you want to create OTDs from, select one or more **.sef** files, and then click **Next**. See Figure 24.

Figure 24 SEF Wizard: Selecting One or More SEF Files for OTD Creation



Steps

1. Select Wizard Type
2. **Select SEF File(s)**
3. Select OTD Options

Look In: SEF_samples

- T867_03.SEF
- X12_4010_850.sef
- X12_4020.SEF

File Name: X12_4020.SEF

Files of Type: SEF File Type

Selected files:

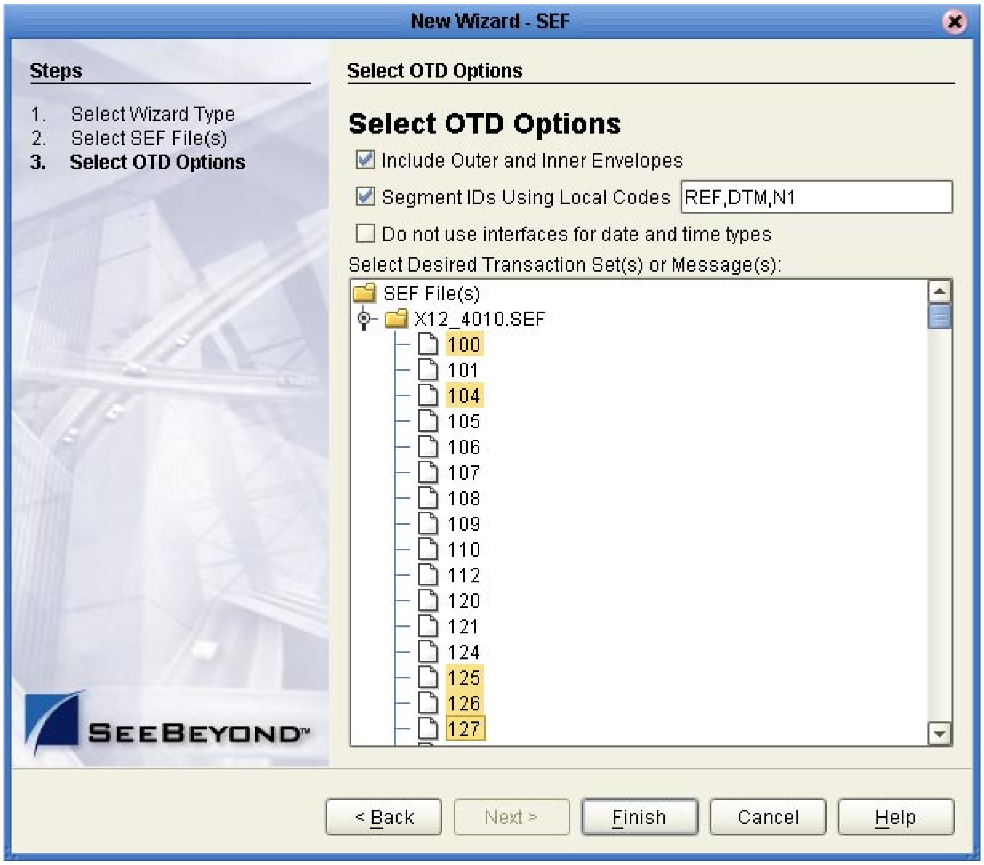
- X12_4010_850.sef
- T867_03.SEF
- X12_4020.SEF

3 In step 3 (**Select OTD Options**), specify:

- **Include Outer and Inner Envelopes**—When this check box is selected, the generated OTD(s) will include the outer and inner envelope segments:
  - For an X12 OTD, these segments include interchange envelope segments (ISA, IEA) and functional group envelope segments (GS, GE).
  - For an EDIFACT OTD (batch only; EDIFACT interactive messages are not supported), these segments include interchange group envelope segments (UNA, UNB, UNZ) and functional group envelope segments (UNG and UNE).
- **Segment IDs Using Local Codes**—When this check box is selected, the text entry area allows you to enter a list of segments separated by commas (such as "REF,DTM,N1"). This means the OTD runtime will use local codes specified for each of these segments to parse incoming data. Local codes are often used to distinguish adjacent segment data that share the same segment ID.
- In the list displayed, you can use the SHIFT key to select a range and the CTRL key to add or remove specific transaction sets / messages.

An example appears in Figure 25.

Figure 25 SEF Wizard: Selecting Options for OTD Creation



Select OTD Options

Select OTD Options

☑ Include Outer and Inner Envelopes

☑ Segment IDs Using Local Codes REF,DTM,N1

□ Do not use interfaces for date and time types

Select Desired Transaction Set(s) or Message(s):

SEF File(s)
X12_4010.SEF
100
101
104
105
106
107
108
109
110
112
120
121
124
125
126
127

4 When you have finished indicating your choices, click **Finish**.

Result: The generated OTD is displayed in the OTD Editor.

Chapter 5

Using eXchange in Enterprise Designer

This chapter provides step-by-step procedures for using the Enterprise Designer tools, editors, components, and prebuilt protocols and libraries provided by eXchange.

What's in This Chapter

5.1 Overview

You use Enterprise Designer to set up the following:

- Environments containing external servers, such as for Oracle and LDAP (required) and whichever transport protocols you use (File, BatchFTP, HTTP, and so forth).
- One or more B2B Hosts configured with attribute definitions (metadata).
- *B2B Host projects*, where each B2B Host is mapped to an Environment and built. Building a **B2B Host project** accomplishes two things:
    - It populates the LDAP database with metadata for the B2B Host.
    - It creates an eXchangeService that communicates with the Oracle database. The eXchangeService is itself used to connect with key business processes (BPs) and Java Collaborations (JCDs).
- *GUI projects—ePM and Tracker:*
    - Building and deploying the **ePM project** creates an application, eXchange Partner Manager (ePM), that allows trading partner information to be viewing and updated via the Web.
    - Building and deploying the **Tracker project** creates another application (Message Tracker) that allows message data to be retained, retrieved, filtered, and viewed via the Web.

- *Error-handling projects.* Building and deploying the **Sub_DLQ** and **Sub_ProcErrors projects** is optional, but recommended. They take undeliverable messages (both "dead-letter" messages that cannot be parsed as well as poorly formed messages that do parse, but with errors) and write them into files for easy access and review.
- *Deployment projects.* Building and deploying a **Deployment project** connects the logic of your own business rules with the BPs and JCDs that constitute the core services of the B2B Suite (batching, delivery, error-handling, and so forth), through the mediation of the environment-specific eXchangeService.

eXchange centers around the concept of a *transaction profile* for each trading partner relationship. Transaction profile information is stored on an LDAP-compliant directory server (usually called an LDAP server) through the following workflow:

- Metadata for a transaction profile is defined in Enterprise Designer in the B2B Host.
- Connecting the B2B Host to LDAP and building the B2B Host creates an eXchangeService and causes the metadata to be stored on LDAP.
- Connecting the ePM GUI application to LDAP and building/deploying the application allows eXchange Partner Manager (ePM) to communicate with LDAP.
- Values are eventually supplied using the eXchange Partner Manager (ePM) GUI, and activation of the trading partner produces transaction profiles stored on LDAP.

You use Enterprise Designer to set up the B2B Host, connect it to LDAP, and build it. The B2B Host includes one or more sets of *attribute definitions* for protocol-related and transport parameters:

- *Business protocol* attribute definitions (BPADs) that contain business and enveloping attribute definitions (BADs and EADs).
- *Delivery protocol* attribute definitions (DPADs) contain messaging and packaging attribute definitions (MADs and PADs).
- *Transport* attribute definitions are called TADs.

The business protocol manager applications available for eXchange, such as X12 or HIPAA, come equipped with prebuilt BPADs; or you can create your own BPADs for custom business protocols. Similarly for transport: You can use either the standard TADs supplied with eXchange (HTTP, FTP, ...) or create custom ones that you set up yourself.

After the B2B Host is set up with all its attribute definitions, a connectivity map is created to connect it to an LDAP server that is also connected to the ePM GUI. In the same connectivity map, the B2B Host is connected to an Oracle Server that will also be connected to a Message Tracker application. Building the connectivity map causes the metadata to be stored on the LDAP server, and also creates an external, an *eXchange Service*, in the same Environment that contains the Oracle and LDAP externals.

For business logic, many prebuilt business processes BPs are supplied with the eXchange product to handle such B2B functions as batching, tracking, and dialogs; all of these BPs can be customized if you want. In addition, the eXchange Protocol Designer can be used to design and configure custom BPs that you create.

BPs for inbound and/or outbound messages are dragged into a connectivity map, where they are represented as services. There, they are connected in usual fashion with externals (including the eXchangeService) and with other services. Activation of a corresponding Deployment Profile exposes the connectivity map's components for processing by Logical Hosts. All information is stored to LDAP, thus making it available for viewing or modification by ePM.

The chapter provides information on the following procedures:

- Steps for creating a B2B Host and populating it with attribute definitions.
- Steps for configuring the external systems in the environment, including the B2B Configurator.
- Optional steps for creating and configuring custom attribute definitions.
- Steps for building a B2B Host connected to an LDAP external and an Oracle external, thus creating the eXchangeService.
- Steps for building and deploying projects for the GUI and for error-handling.

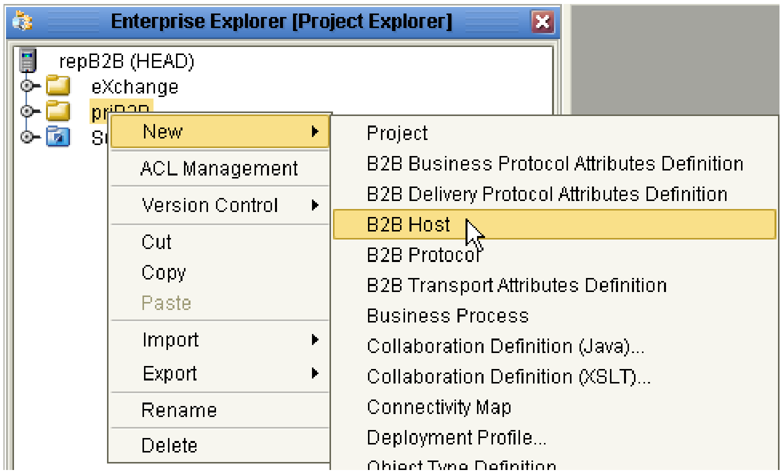
5.2 Setting Up a B2B Host and Its Components

This section explains how to create a B2B Host and populate it with attribute definitions. The editor used for configuring B2B Hosts is the **B2B Host Designer**.

To create a B2B host

1. In Enterprise Designer with the Project Explorer tab active, in the project tree, right-click the Project or subproject where the B2B Host will reside, click **New**, and then click **B2B Host**. See Figure 26.

Figure 26 Creating a New B2B Host



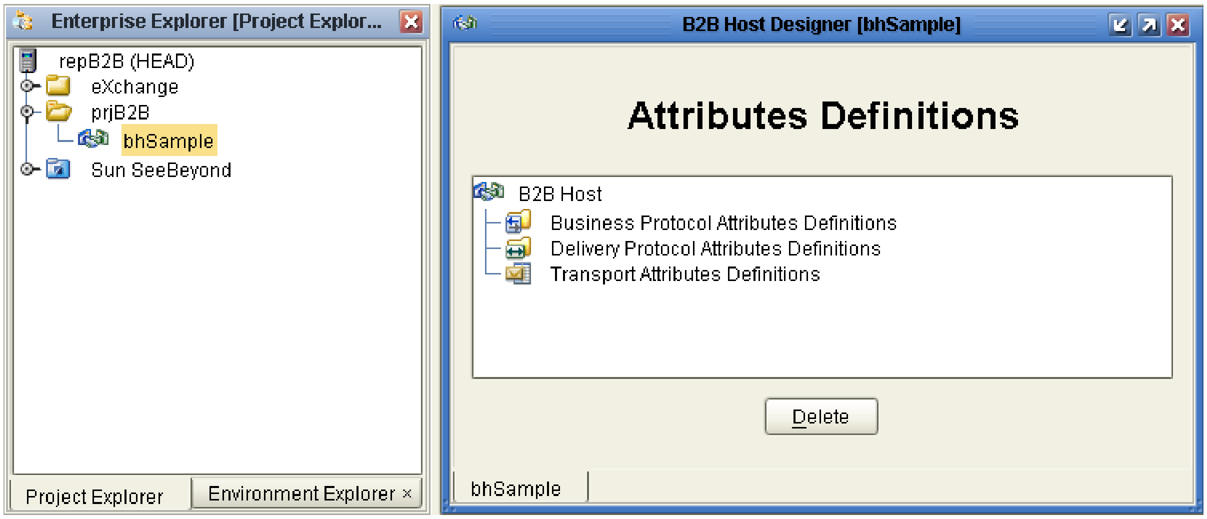
Enterprise Explorer [Project Explorer]
repB2B (HEAD)
eXchange
prjB2B
S
New
ACL Management
Version Control
Cut
Copy
Paste
Import
Export
Rename
Delete
Project
B2B Business Protocol Attributes Definition
B2B Delivery Protocol Attributes Definition
B2B Host
B2B Protocol
B2B Transport Attributes Definition
Business Process
Collaboration Definition (Java)...
Collaboration Definition (XSLT)...
Connectivity Map
Deployment Profile...
Object Type Definition

2. When the new component appears in the project tree, you can click-pause-click to rename it from **B2BHost1** to whatever you want. It is good practice to use a short name beginning with "bh" to indicate that the component is a B2B Host.

Result: The project tree displays the new component with a "handshake" icon. Also, the **B2B Host Designer** opens to display a tree of Attribute Definitions, with containers for

Business Protocol Attribute Definitions, Delivery Protocol Attribute Definitions, and Transport Attribute Definitions. See Figure 27.

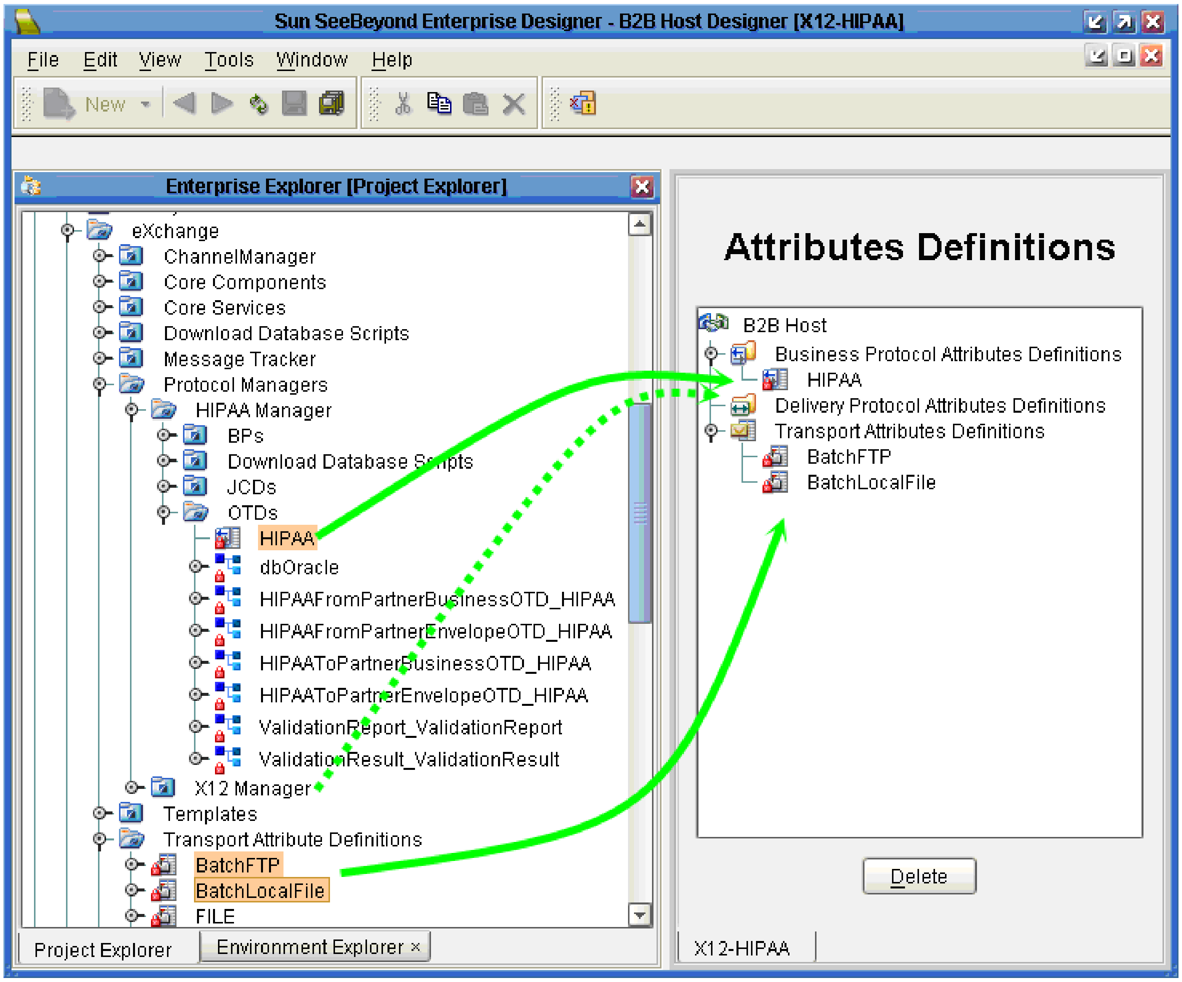
Figure 27 B2B Host Designer


Enterprise Explorer [Project Explor...
repB2B (HEAD)
eXchange
prjB2B
bhSample
Sun SeeBeyond
Project Explorer
Environment Explorer
B2B Host Designer [bhSample]
Attributes Definitions
B2B Host
Business Protocol Attributes Definitions
Delivery Protocol Attributes Definitions
Transport Attributes Definitions
Delete
bhSample

To populate a B2B host with attribute definitions

- For *custom* BPADs, DPADs, or TADs: Open the project or subproject and drag the attributes definition into the Attribute Definitions window. Repeat as needed for other custom attribute definitions.
- For *presupplied* attribute definitions: In the project explorer tree, open the SeeBeyond > eXchange folder, and:
  - *For X12:* Open the **Protocol Managers > X12 Manager > OTDs** folder and drag the **X12** BPAD into the B2B Host Designer window. See Figure 28.
  - *For HIPAA:* Open the **Protocol Managers > HIPAA Manager > OTDs** folder and drag the **HIPAA** BPAD into the B2B Host Designer window. See Figure 28.
  - *For standard transports:* Open the **Transport Attribute Definitions** folder and drag one or more TADs into the B2B Host Designer window. See Figure 28.

Figure 28 Business Protocols Window Showing Presupplied BPADs and TADs


Sun SeeBeyond Enterprise Designer - B2B Host Designer [X12-HIPAA]
File
Edit
View
Tools
Window
Help
New
Enterprise Explorer [Project Explorer]
eXchange
ChannelManager
Core Components
Core Services
Download Database Scripts
Message Tracker
Protocol Managers
HIPAA Manager
BPs
Download Database Scripts
JCDs
OTDs
HIPAA
dbOracle
HIPAAFromPartnerBusinessOTD_HIPAA
HIPAAFromPartnerEnvelopeOTD_HIPAA
HIPAAToPartnerBusinessOTD_HIPAA
HIPAAToPartnerEnvelopeOTD_HIPAA
ValidationReport_ValidationReport
ValidationResult_ValidationResult
X12 Manager
Templates
Transport Attribute Definitions
BatchFTP
BatchLocalFile
FILE
Project Explorer
Environment Explorer
Attributes Definitions
B2B Host
Business Protocol Attributes Definitions
HIPAA
Delivery Protocol Attributes Definitions
Transport Attributes Definitions
BatchFTP
BatchLocalFile
Delete
X12-HIPAA

Result: Figure 28 (above) shows a B2B Host populated with attribute definitions. You can combine several protocols in one B2B Host, and you can combine custom BPADs, DPADs, and TADs with standard ones.

5.3 Setting Up Attribute Definitions

This section explains how to create and configure custom attribute definitions:

- **Creating and Configuring TADs** on page 70
- **Creating and Configuring DPADs** on page 73
- **Creating and Configuring BPADs** on page 75

5.3.1 Creating and Configuring TADs

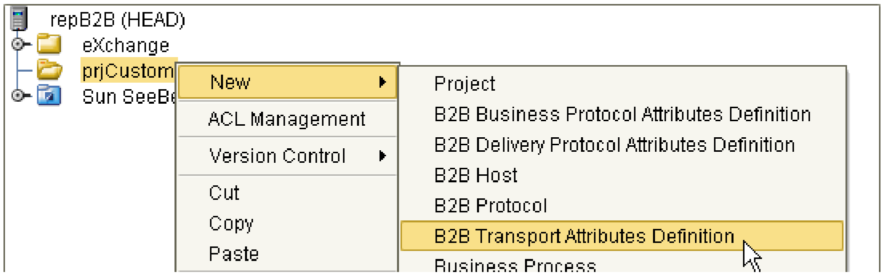
In general, a *protocol* is a code of behavior: a framework for interpretation and communication that is agreed upon by all parties. It specifies rules for interacting with others who are using the same protocol.

A *transport protocol* provides a way of specifying how data is to be delivered from one system to another. For example, FTP (file transfer protocol) requires the client to specify a transfer mode (such as ASCII or binary), a target directory, a target filename or file pattern, and so forth; in eXchange, these parameters are specified by the standard *transport attributions definition* for FTP. eXchange supplies attribute definitions for the following standard transport protocols: BatchFTP, BatchLocalFile, File, FTP, HTTP, HTTPS, JMS, and SMTP.

In addition to the attribute definitions for the standard transport protocols noted above, you can use ***custom transport attribute definitions*** that specify custom modifications or extensions of the standard transport protocols.

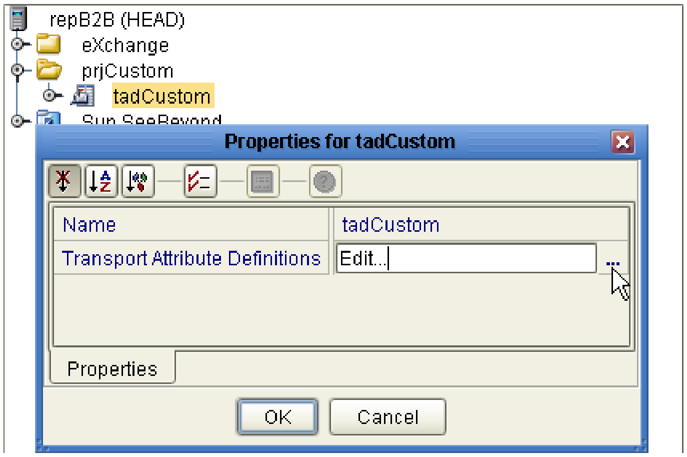
To create a custom transport attributes definition

- In Enterprise Designer with the Project Explorer tab active, in the project tree, right-click the project or subproject where the transport attributes definition will reside, click **New**, and then click **B2B Transport Attributes Definition**:



Result: The project tree displays the new component, and a properties dialog allows you to edit the attribute definitions; see Figure 29. It is good practice to rename it from **B2BTransportAttributesDefinition<*n*>** to a meaningful short name beginning with **tad** to indicate that the component is a Transport Attributes Definition.

Figure 29 Custom Transport Attributes Definition



Name	tadCustom
Transport Attribute Definitions	Edit...

This new component is useful only by virtue of its configuration—you will need to add and define attributes that govern the nodes in the OTD that will be generated from it. Once attributes are defined, they can be exposed to eXchange Partner Manager (ePM) for delivery channel configuration.

To configure a custom transport attributes definition

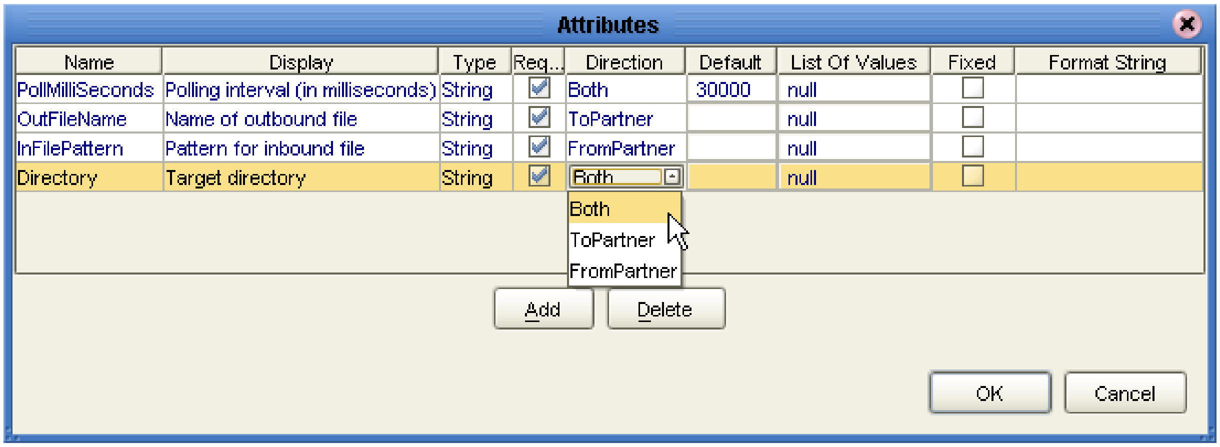
Note: *If the component is locked, you must check it out before you can modify it.*

1 In the project tree, right-click the transport attributes definition you want to modify, and click **Properties**.

2 In the properties dialog, to the far right of the value for Transport Attribute Definitions, click the ellipsis [**...**] button; see Figure 29.

The **Attributes** dialog appears; Figure 30 shows attributes for a sample bidirectional transport attributes definition that is a modification of the basic File TAD.

Figure 30 Custom Attribute Definitions



Name	Display	Type	Req...	Direction	Default	List Of Values	Fixed	Format String
PollMilliSeconds	Polling interval (in milliseconds)	String	☑	Both	30000	null	□	
OutFileName	Name of outbound file	String	☑	ToPartner		null	□	
InFilePattern	Pattern for inbound file	String	☑	FromPartner		null	□	
Directory	Target directory	String	☑	Both		null	□	

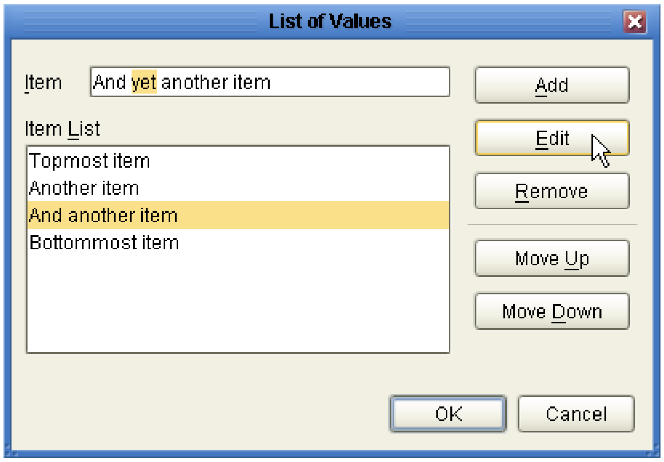
You use this dialog to create and set attributes. These values govern the appearance and behavior of the parameters displayed in ePM when configuring external delivery channels for a trading partner profile, in the ToPartner Transport and FromPartner Transport subtabs.

3 Click the **Add** button as many times as needed and then, for each row created:

- Change **Name** to a meaningful node name for the OTD you will generate.
- Change **Display** to the text you want to display as a prompt or label for the parameter in ePM.
- For **Type**, select the data type for this attribute. The default, **String**, allows the ePM user to enter any character data; **Password** also accepts any ePM input, and masks the input; **Integer** accepts positive or negative whole numbers only; **Number** extends this to also accept decimal numbers (floating-point numbers); **Boolean** requires the ePM user to make a yes-or-no choice; **List of Values** presents the ePM user with a drop-down list restricted to the items you have set up (see below); and **DateTime** prompts the ePM user to supply a date and/or time value, based on the formatting you provide (see Format String, below).
- For **Required**, select or clear the box according to whether you want the parameter to be a required or optional entry. (In the ePM GUI, parameters that have been designated as required are flagged with a red asterisk.)
- For **Direction**, choose **ToPartner**, **FromPartner**, or **Both** according to whether you want the parameter to appear with the ToPartner parameters, FromPartner parameters, or both.
- For **Default**, you can optionally enter a default value that will appear in ePM before the user enters data or makes a selection. This is the value that will be used if it is not overridden by the ePM user.
- For **List of Values**, which is available only for an attribute whose data type is “List of Values” (see Type, above), double-click on the ellipsis [**...**] button to the far right and use the **List of Values** dialog box to add entries to the drop-down list that will be seen by the end user. Clicking Add appends a new item to the end of the list; Edit allows you to modify the currently selected item (see Figure 31); Move Up and Move Down move it higher or lower in the list.

Figure 31 Attributes Definition - List of Values

f



Item: And yet another item

Item List:
Topmost item
Another item
And another item
Bottommost item

- **Format String** allows you to use special characters as shorthand for certain often-used information; for example, **%f** is the working filename, **%M** is the current month, **%d** the current day, and so forth. For more information, see the *Batch eWay Intelligent Adapter User's Guide*; the chapter on understanding OTDs has a section on using special characters.

4 When you have finished adding and modifying attributes, click **OK**.

5 Click **OK** to close the properties dialog.

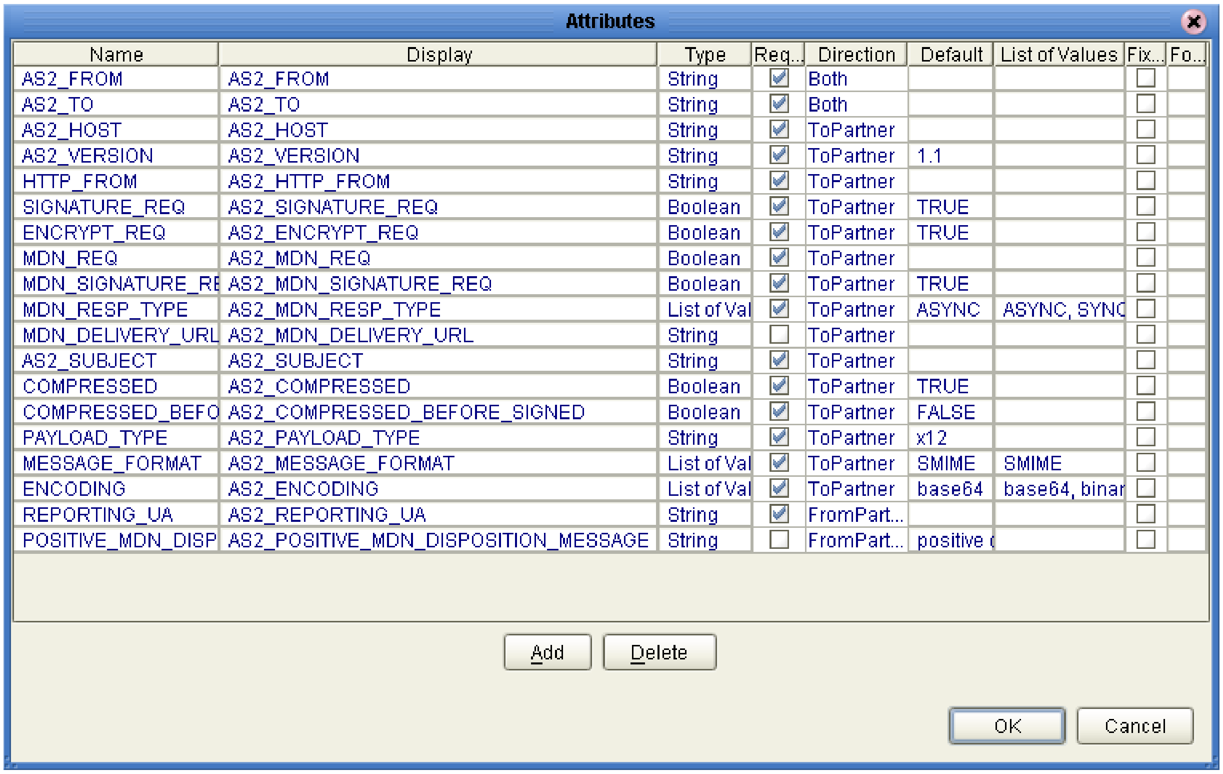
Result: The completed TAD can be dragged into the B2B Host; after other steps are also completed, the metadata in its attribute definitions will be stored on the LDAP server, and the parameters it defines will appear in eXchange Partner Manager (ePM).

5.3.2 Creating and Configuring DPADs

Delivery protocol attribute definitions, or DPADs, are used by delivery protocols such as AS2 and ebXML. DPADs consist of messaging attribute definitions (MADs) and packaging attribute definitions (PADs). If you do not use delivery protocols, your B2B Host does not need to define any DPADs.

Generally, DPADs provide a way of specifying how data is bundled and unbundled—for example, in AS2 and ebXML, it is at this level that encryption, acknowledgment, and nonrepudiation are addressed. Figure 32 provides an example of the MAD for AS2 version 1.1.

Figure 32 Default Messaging Attributes Definition (MAD) for AS2 v1.1



Attributes

Name	Display	Type	Req...	Direction	Default	List of Values	Fix...	Fo...
AS2_FROM	AS2_FROM	String	☑	Both			□	
AS2_TO	AS2_TO	String	☑	Both			□	
AS2_HOST	AS2_HOST	String	☑	ToPartner			□	
AS2_VERSION	AS2_VERSION	String	☑	ToPartner	1.1		□	
HTTP_FROM	AS2_HTTP_FROM	String	☑	ToPartner			□	
SIGNATURE_REQ	AS2_SIGNATURE_REQ	Boolean	☑	ToPartner	TRUE		□	
ENCRYPT_REQ	AS2_ENCRYPT_REQ	Boolean	☑	ToPartner	TRUE		□	
MDN_REQ	AS2_MDN_REQ	Boolean	☑	ToPartner			□	
MDN_SIGNATURE_RE	AS2_MDN_SIGNATURE_REQ	Boolean	☑	ToPartner	TRUE		□	
MDN_RESP_TYPE	AS2_MDN_RESP_TYPE	List of Val	☑	ToPartner	ASYNC	ASYNC, SYNC	□	
MDN_DELIVERY_URL	AS2_MDN_DELIVERY_URL	String	□	ToPartner			□	
AS2_SUBJECT	AS2_SUBJECT	String	☑	ToPartner			□	
COMPRESSED	AS2_COMPRESSED	Boolean	☑	ToPartner	TRUE		□	
COMPRESSED_BEFO	AS2_COMPRESSED_BEFORE_SIGNED	Boolean	☑	ToPartner	FALSE		□	
PAYLOAD_TYPE	AS2_PAYLOAD_TYPE	String	☑	ToPartner	x12		□	
MESSAGE_FORMAT	AS2_MESSAGE_FORMAT	List of Val	☑	ToPartner	SMIME	SMIME	□	
ENCODING	AS2_ENCODING	List of Val	☑	ToPartner	base64	base64, binar	□	
REPORTING_UA	AS2_REPORTING_UA	String	☑	FromPart...			□	
POSITIVE_MDN_DISP	AS2_POSITIVE_MDN_DISPOSITION_MESSAGE	String	□	FromPart...	positive (		□	

To create a new DPAD

1. In Enterprise Designer with the Project Explorer tab active, in the project tree, right-click the project or subproject where the DPAD will reside.
2. On the popup context menu, point at **New**, and then click **B2B Delivery Protocol Attributes Definition**.
3. In the properties, enter a name (good practice is to enter a short name beginning with "dpad", such as **dpadCustom1**) and click **OK**.

   The project tree displays the new component.

This new component is useful only by virtue of its configuration—you will need to add name/value pairs for the parameters you want it to define.

To configure a DPAD

1. In Enterprise Designer, with the Project Explorer tab active, right-click the DPAD you want to modify.

Note: *If the component is locked, you must check it out before you can modify it.*

2. To the far right of the value for **Messaging Attribute Definitions**, click the ellipsis [**...**] button.

   The **Messaging Attribute Definitions** dialog appears. You use this dialog to create and set the messaging attributes.
3. Click the **Add** button as many times as needed and then, for each row created:
   - Change **Name** to a meaningful node name for the OTD you will generate.
   - Change **Display** to the text you want to display as a prompt or label for the parameter in ePM.
   - For **Type**, select the data type for this attribute. The default, **String**, allows the ePM user to enter any character data; **Password** also accepts any ePM input, and masks the input; **Integer** accepts positive or negative whole numbers only; **Number** extends this to also accept decimal numbers (floating-point numbers); **Boolean** requires the ePM user to make a yes-or-no choice; **List of Values** presents the ePM user with a drop-down list restricted to the items you have set up (see below); and **DateTime** prompts the ePM user to supply a date and/or time value, based on the formatting you provide (see Format String, below).
   - For **Required**, select or clear the box according to whether you want the parameter to be a required or optional entry. (In the ePM GUI, parameters that have been designated as required are flagged with a red asterisk.)
   - For **Direction**, choose **ToPartner**, **FromPartner**, or **Both** according to whether you want the parameter to appear with the ToPartner parameters, FromPartner parameters, or both.
   - For **Default**, you can optionally enter a default value that will appear in ePM before the user enters data or makes a selection. This is the value that will be used if it is not overridden by the ePM user.

- For **List of Values**, which is available only for an attribute whose data type is "List of Values" (see Type, above), double-click on the ellipsis [**...**] button to the far right and use the **List of Values** dialog box to add entries to the drop-down list that will be seen by the end user. Clicking Add appends a new item to the end of the list; Edit modifies the currently selected item (see Figure 31); Up and Down move it higher or lower in the list.
- **Format String** allows you to use special characters as shorthand for certain often-used information; for example, **%f** is the working filename, **%M** is the current month, **%d** the current day, and so forth. For more information, see the *Batch eWay Intelligent Adapter User's Guide*; the chapter on understanding OTDs has a section on using special characters.

4 When you have finished adding and modifying attributes, click **OK**.

5 Repeat the corresponding steps for the packaging attributes definition (PAD).

6 Click **OK** to close the properties dialog.

The DPAD can now be used to generate an OTD.

After you have completed these steps, the DPAD appears as a choice in the drop-down list of DPADs when you configure the delivery protocols of your B2B Host.

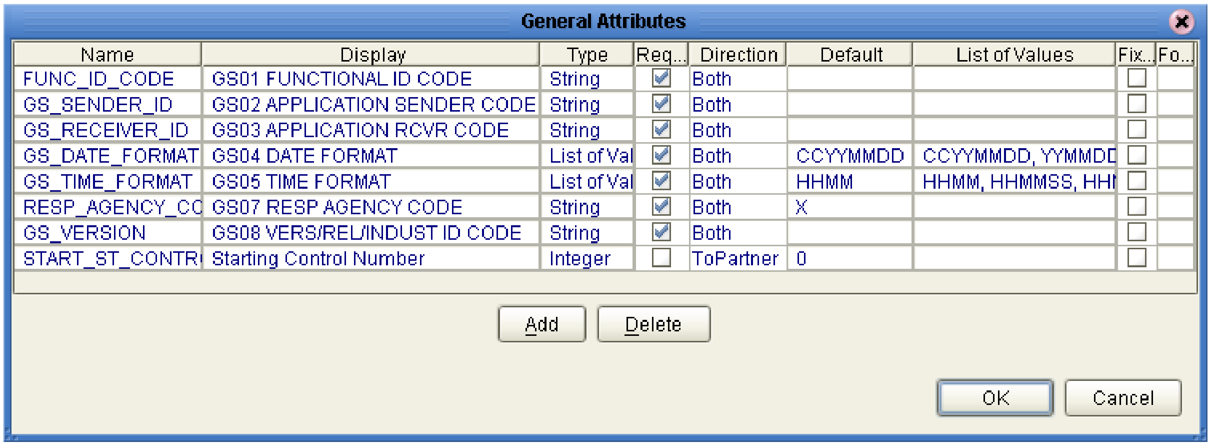
5.3.3 Creating and Configuring BPADs

Business protocol attribute definitions (BPADs) consist of business attribute definitions (BADs) and enveloping attribute definitions (EADs). BPADs are used by business protocols, such as X12, HIPAA, and EDIFACT. If you do not use business protocols, your B2B Host does not need to define any BPADs.

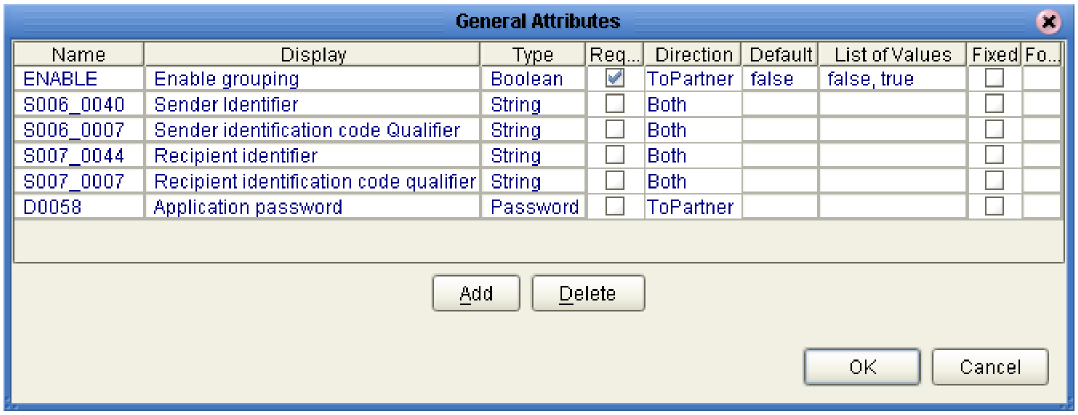
Generally, BADs specify business protocol parameters at the group level (that is, the inner envelope), and EADs specify business protocol parameters at the interchange level (that is, the outer envelope).

Figure 33 provides examples of a BAD for X12 (named "X12") and a BAD for EDIFACT (named "EDIFACT-UNG"). The attributes in the BAD specify the label, type, and so forth of parameters that will be displayed in ePM.

Figure 33 Default Business Attribute Definitions (BADs) for X12 and EDIFACT



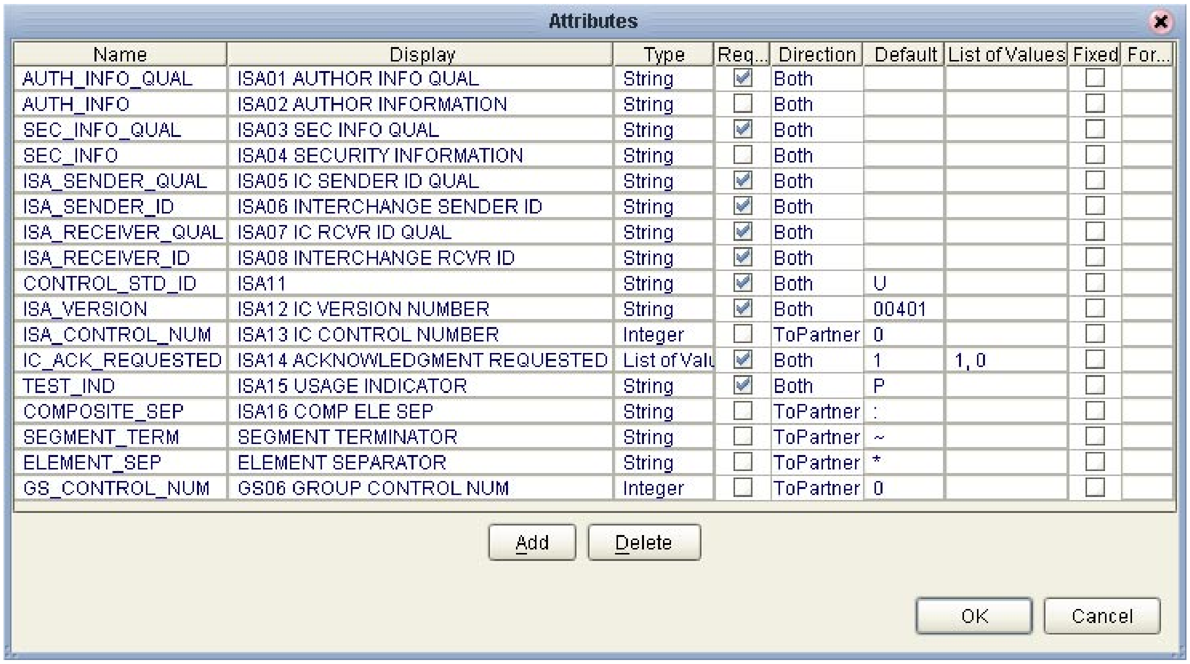
Name	Display	Type	Req...	Direction	Default	List of Values	Fix...	Fo...
FUNC_ID_CODE	GS01 FUNCTIONAL ID CODE	String	☑	Both			□	
GS_SENDER_ID	GS02 APPLICATION SENDER CODE	String	☑	Both			□	
GS_RECEIVER_ID	GS03 APPLICATION RCVR CODE	String	☑	Both			□	
GS_DATE_FORMAT	GS04 DATE FORMAT	List of Val	☑	Both	CCYYMMDD	CCYYMMDD, YYMMDD	□	
GS_TIME_FORMAT	GS05 TIME FORMAT	List of Val	☑	Both	HHMM	HHMM, HHMMSS, HHI	□	
RESP_AGENCY_CC	GS07 RESP AGENCY CODE	String	☑	Both	X		□	
GS_VERSION	GS08 VERS/REL/INDUST ID CODE	String	☑	Both			□	
START_ST_CONTRI	Starting Control Number	Integer	□	ToPartner	0		□	



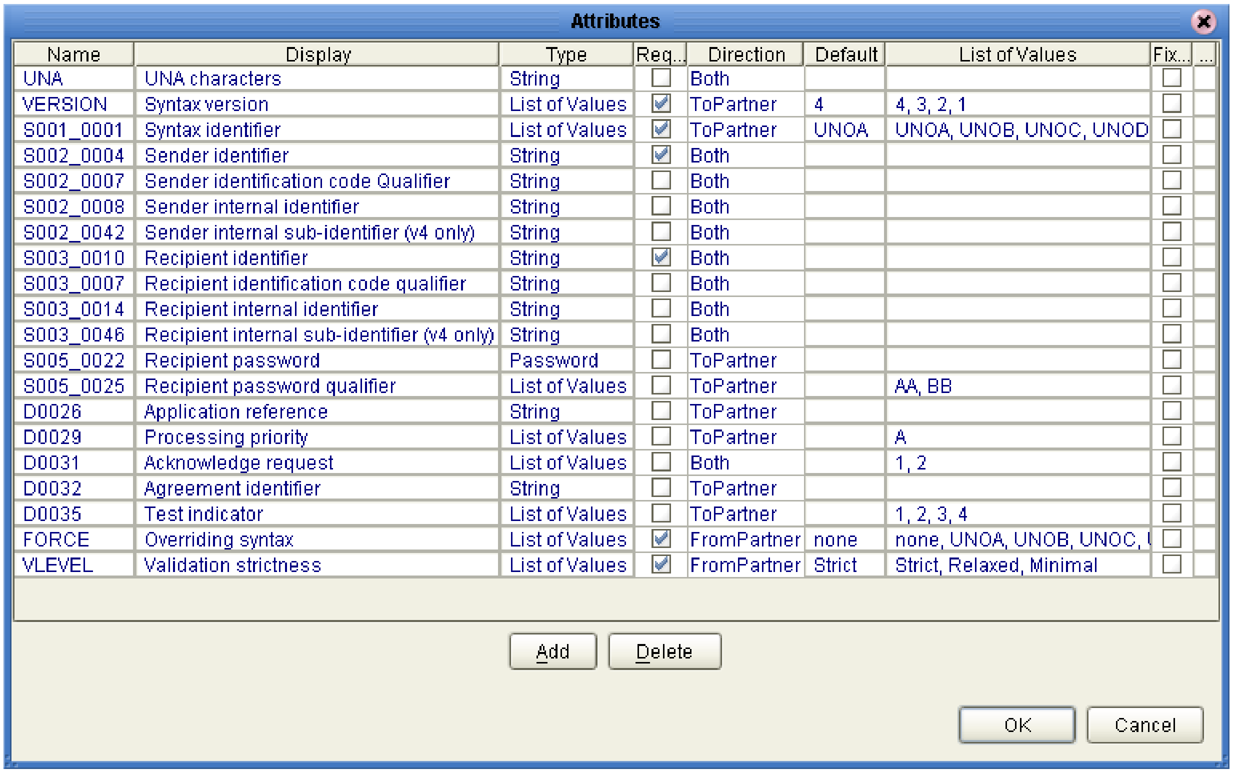
Name	Display	Type	Req...	Direction	Default	List of Values	Fixed	Fo...
ENABLE	Enable grouping	Boolean	☑	ToPartner	false	false, true	□	
S006_0040	Sender Identifier	String	□	Both			□	
S006_0007	Sender identification code Qualifier	String	□	Both			□	
S007_0044	Recipient identifier	String	□	Both			□	
S007_0007	Recipient identification code qualifier	String	□	Both			□	
D0058	Application password	Password	□	ToPartner			□	

Figure 34 provides examples of an EAD for X12 (named "Envelope") and an EAD for EDIFACT (named "EDIFACT-UNB"). The attributes in the EAD specify the label, type, and so forth for parameters that will be displayed in ePM.

Figure 34 Default Enveloping Attribute Definitions (EADs) for X12 and EDIFACT



Name	Display	Type	Req...	Direction	Default	List of Values	Fixed	For...
AUTH_INFO_QUAL	ISA01 AUTHOR INFO QUAL	String	☑	Both			□	
AUTH_INFO	ISA02 AUTHOR INFORMATION	String	□	Both			□	
SEC_INFO_QUAL	ISA03 SEC INFO QUAL	String	☑	Both			□	
SEC_INFO	ISA04 SECURITY INFORMATION	String	□	Both			□	
ISA_SENDER_QUAL	ISA05 IC SENDER ID QUAL	String	☑	Both			□	
ISA_SENDER_ID	ISA06 INTERCHANGE SENDER ID	String	☑	Both			□	
ISA_RECEIVER_QUAL	ISA07 IC RCVR ID QUAL	String	☑	Both			□	
ISA_RECEIVER_ID	ISA08 INTERCHANGE RCVR ID	String	☑	Both			□	
CONTROL_STD_ID	ISA11	String	☑	Both	U		□	
ISA_VERSION	ISA12 IC VERSION NUMBER	String	☑	Both	00401		□	
ISA_CONTROL_NUM	ISA13 IC CONTROL NUMBER	Integer	□	ToPartner	0		□	
IC_ACK_REQUESTED	ISA14 ACKNOWLEDGMENT REQUESTED	List of Valu	☑	Both	1	1, 0	□	
TEST_IND	ISA15 USAGE INDICATOR	String	☑	Both	P		□	
COMPOSITE_SEP	ISA16 COMP ELE SEP	String	□	ToPartner	:		□	
SEGMENT_TERM	SEGMENT TERMINATOR	String	□	ToPartner	~		□	
ELEMENT_SEP	ELEMENT SEPARATOR	String	□	ToPartner	*		□	
GS_CONTROL_NUM	GS06 GROUP CONTROL NUM	Integer	□	ToPartner	0		□	



Name	Display	Type	Req...	Direction	Default	List of Values	Fix...	...
UNA	UNA characters	String	□	Both			□	
VERSION	Syntax version	List of Values	☑	ToPartner	4	4, 3, 2, 1	□	
S001_0001	Syntax identifier	List of Values	☑	ToPartner	UNOA	UNOA, UNOB, UNOC, UNOD	□	
S002_0004	Sender identifier	String	☑	Both			□	
S002_0007	Sender identification code Qualifier	String	□	Both			□	
S002_0008	Sender internal identifier	String	□	Both			□	
S002_0042	Sender internal sub-identifier (v4 only)	String	□	Both			□	
S003_0010	Recipient identifier	String	☑	Both			□	
S003_0007	Recipient identification code qualifier	String	□	Both			□	
S003_0014	Recipient internal identifier	String	□	Both			□	
S003_0046	Recipient internal sub-identifier (v4 only)	String	□	Both			□	
S005_0022	Recipient password	Password	□	ToPartner			□	
S005_0025	Recipient password qualifier	List of Values	□	ToPartner		AA, BB	□	
D0026	Application reference	String	□	ToPartner			□	
D0029	Processing priority	List of Values	□	ToPartner		A	□	
D0031	Acknowledge request	List of Values	□	Both		1, 2	□	
D0032	Agreement identifier	String	□	ToPartner			□	
D0035	Test indicator	List of Values	□	ToPartner		1, 2, 3, 4	□	
FORCE	Overriding syntax	List of Values	☑	FromPartner	none	none, UNOA, UNOB, UNOC, U	□	
VLEVEL	Validation strictness	List of Values	☑	FromPartner	Strict	Strict, Relaxed, Minimal	□	

You can also create custom BPADs, either from scratch or by modifying/extending the versions shipped with X12 or HIPAA protocol managers.

To create a BPAD

1 In Enterprise Designer with the Project Explorer tab active, in the project tree, right-click the project or subproject where the BPAD will reside.

2 On the popup context menu, point at **New**, and then click **B2B Business Protocol Attributes Definition**.

3 In the properties, enter a name (good practice is to enter a short name beginning with “bpad”, such as **bpadCustom1**) and click **OK**.

The project tree displays the new component.

This new component is useful only by virtue of its configuration—you will need to add name/value pairs for the parameters you want it to define.

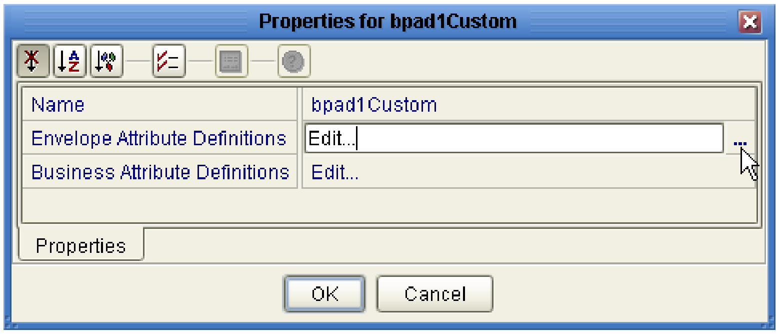
To configure a BPAD

1 In Enterprise Designer, with the Project Explorer tab active, right-click the BPAD you want to modify.

Note: *If the component is locked, you must check it out before you can modify it.*

2 To the far right of the value for **Envelope Attribute Definitions**, click the ellipsis **[...]** button (see Figure 36) to display the **Envelope Attribute Definitions** dialog box. You use this dialog box to create and set the envelope attribute definitions (EAD) of the current BPAD.

Figure 35 Accessing a BPAD’s Envelope Attribute Definitions (EAD)



Properties for bpad1Custom

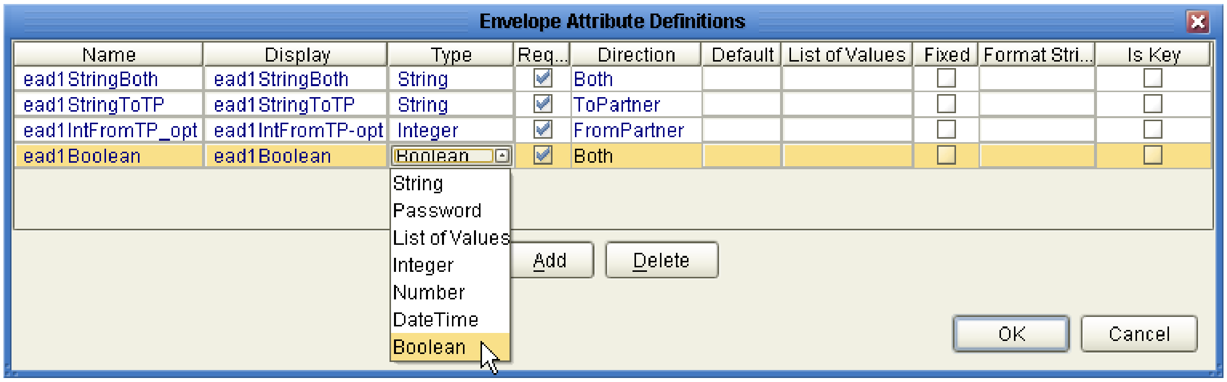
Name	bpad1Custom
Envelope Attribute Definitions	Edit...
Business Attribute Definitions	Edit...

3 Click the **Add** button as many times as needed (a four-row EAD is illustrated in Figure 36) and then, for each row created:

- Change **Name** to a meaningful node name for the OTD you will generate.
- Change **Display** to the text you want to display as a prompt or label for the parameter in ePM.
- For **Type**, select the data type for this attribute. The default, **String**, allows the ePM user to enter any character data; **Password** also accepts any ePM input, and masks the input; **Integer** accepts positive or negative whole numbers only; **Number** extends this to also accept decimal numbers (floating-point numbers); **Boolean** requires the ePM user to make a yes-or-no choice; **List of Values** presents the ePM user with a drop-down list restricted to the items you have set up (see below); and **DateTime** prompts the ePM user to supply a date and/or time value, based on the formatting you provide (see Format String, below).

- For **Required**, select or clear the box according to whether you want the parameter to be a required or optional entry. (In the ePM GUI, parameters that have been designated as required are flagged with a red asterisk.)
- For **Direction**, choose **ToPartner**, **FromPartner**, or **Both** according to whether you want the parameter to appear with the ToPartner parameters, FromPartner parameters, or both.
- For **Default**, you can optionally enter a default value that will appear in ePM before the user enters data or makes a selection. This is the value that will be used if it is not overridden by the ePM user.
- For **List of Values**, which is available only for an attribute whose data type is “List of Values” (see Type, above), double-click on the ellipsis **[...]** button to the far right and use the **List of Values** dialog box to add entries to the drop-down list that will be seen by the end user. Clicking Add appends a new item to the end of the list; Edit modifies the currently selected item (see Figure 31); Up and Down move it higher or lower in the list.
- **Format String** allows you to use special characters as shorthand for certain often-used information; for example, **%f** is the working filename, **%M** is the current month, **%d** the current day, and so forth. For more information, see the *Batch eWay Intelligent Adapter User’s Guide*; the chapter on understanding OTDs has a section on using special characters.

Figure 36 Custom Envelope Attribute Definitions (EAD)



Name	Display	Type	Req...	Direction	Default	List of Values	Fixed	Format Stri...	Is Key
ead1StringBoth	ead1StringBoth	String	☑	Both			□		□
ead1StringToTP	ead1StringToTP	String	☑	ToPartner			□		□
ead1IntFromTP_opt	ead1IntFromTP-opt	Integer	☑	FromPartner			□		□
ead1Boolean	ead1Boolean	Boolean	☑	Both			□		□

4. When you have finished adding and modifying envelope attributes, click **OK**.
5. Repeat the corresponding steps for the business attributes definition (BAD).
6. Click **OK** to close the properties dialog.

   The BPAD can now be used to generate an OTD.

After you have completed these steps, the BPAD appears as a choice in the drop-down list of BPADs when you configure the delivery protocols of your B2B Host.

After you have completed these steps, activation of the B2B Host will cause the BPAD parameters (that is, the ones you defined for the EAD and the BAD) to appear in ePM.

5.4 Building a B2B Host

The B2B host plays a dual role: It functions both as an *object*—that is, a project-level (logical) component in the project tree that can be dragged into a Connectivity Map—, and also as a *server*—that is, an environment (physical) component, called eXchangeService.

Each B2B Host project joins a particular B2B Host instance to a particular Environment. The name of the created eXchangeService reflects this: For example, if the B2B Host instance is named *cm1myHost123*, and the Environment is named *envA*, the name of the new external added to the Environment is *envA_cm1myHost123 eXchange Service*.

5.4.1 Creating an Environment

These steps below set up the minimal environment required for building a B2B Host that can be used by eXchange Partner Manager (ePM).

To create and populate the environment

1. In Enterprise Designer with the **Environment Explorer** tab active, right-click the Repository and, on the pop-up context menu, click **New Environment**.

   The explorer tree displays a new environment, and the Environment Editor opens. Optionally, you can rename the environment to something meaningful.

2. In the environment explorer tree, right-click the new environment and, on the pop-up context menu, point at **New** and click **Logical Host**.

   The explorer tree and editor canvas display the new Logical Host. Optionally, you can use the tree to rename the Logical Host to something meaningful.

3. In the environment explorer tree, right-click the new Logical Host and, on the pop-up context menu, point at **New** and click **Sun SeeBeyond Integration Server**.

   The explorer tree displays the new Integration Server, and the canvas displays it inside the Logical Host. Optionally, you can rename it to something meaningful.

4. If appropriate, right-click the Integration Server, click **Properties**, and configure its parameters as needed for use at your site.

5. In the environment explorer tree, right-click the new Logical Host and, on the pop-up context menu, point at **New** and click **Sun SeeBeyond JMS IQ Manager**.

   The explorer tree displays the new JMS IQ Manager, and the canvas displays it inside the Logical Host. Optionally, you can rename it to something meaningful.

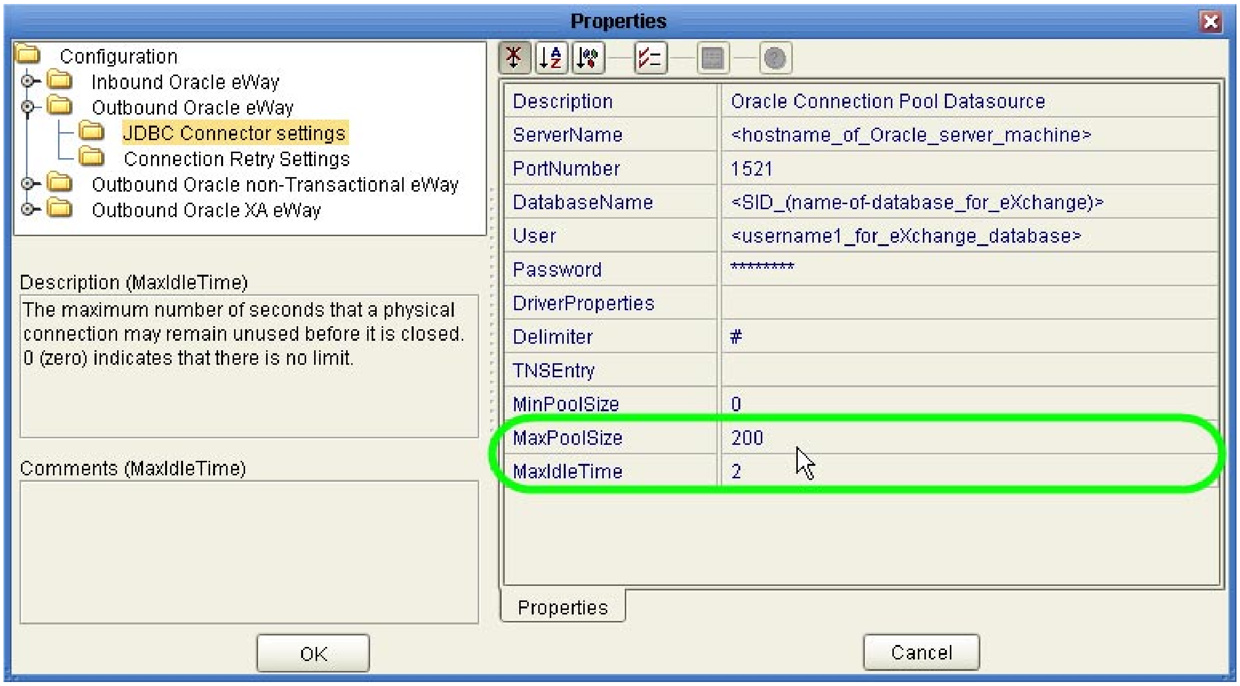
6. If appropriate, right-click the JMS IQ Manager, click **Properties**, and configure its parameters as needed for use at your site.

7. In the environment explorer tree, right-click the environment and, on the pop-up context menu, point at **New** and click **Oracle External System**.

8. In the **Create an External System** dialog, enter a meaningful name, set the system type to **Outbound Oracle eWay**, and then click **OK**.

9 Configure the Oracle external with the values for your eXchange database instance. For details on configuring the Oracle external, see the Chapter 3 of the *Oracle eWay User's Guide*. For sample settings typical of an eXchange database, see Figure 37.

Figure 37 Environment Configuration for Oracle External System



Description (MaxIdleTime)

The maximum number of seconds that a physical connection may remain unused before it is closed. 0 (zero) indicates that there is no limit.

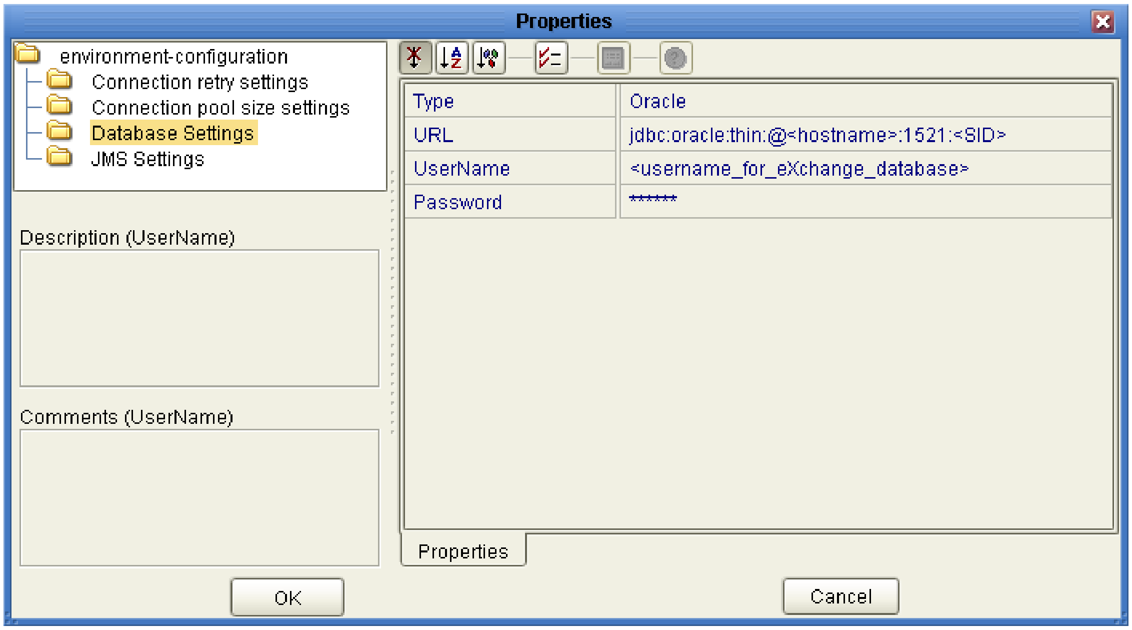
Description	Oracle Connection Pool Datasource
ServerName	<hostname_of_Oracle_server_machine>
PortNumber	1521
DatabaseName	<SID_(name-of-database_for_eXchange)>
User	<username1_for_eXchange_database>
Password	********
DriverProperties	
Delimiter	#
TNSEntry	
MinPoolSize	0
MaxPoolSize	200
MaxIdleTime	2

10 In the environment explorer tree, right-click the environment and, on the pop-up context menu, point at **New** and click **B2BServiceConfigurator External System**.

11 In the **Please input the name** dialog, enter a meaningful name and then click **OK**.

12 Open the properties of the B2BService Configurator and, for Database Settings, supply values for URL, UserName, and Password that match the values you supplied for the Oracle external; for example, see Figure 38.

Figure 38 Environment Configuration for Oracle External System



Type	Oracle
URL	jdbc:oracle:thin:@<hostname>:1521:<SID>
UserName	<username_for_eXchange_database>
Password	******

13 In similar fashion, create an external system for LDAP, name it appropriately, and configure it appropriately for your operating environment.

14 Also create external systems for BatchFTP, BatchLocalFile, File, and HTTP (as well as any other external systems corresponding to other transport protocols used by your B2B Host), and configure them appropriately for your operating environment.

Once you finish these steps, the environment now has all you need to build a B2B Host as well as the GUI, Tracker, and error-handling projects.

To create the eXchange Service, you create a connectivity map that links the B2B Host to an Oracle eWay and then build the corresponding Deployment Profile; see **"Connecting the B2B Host to Oracle and LDAP Externals" on page 84**

5.4.2 Optional: Configuring an Environment to Use HTTPS

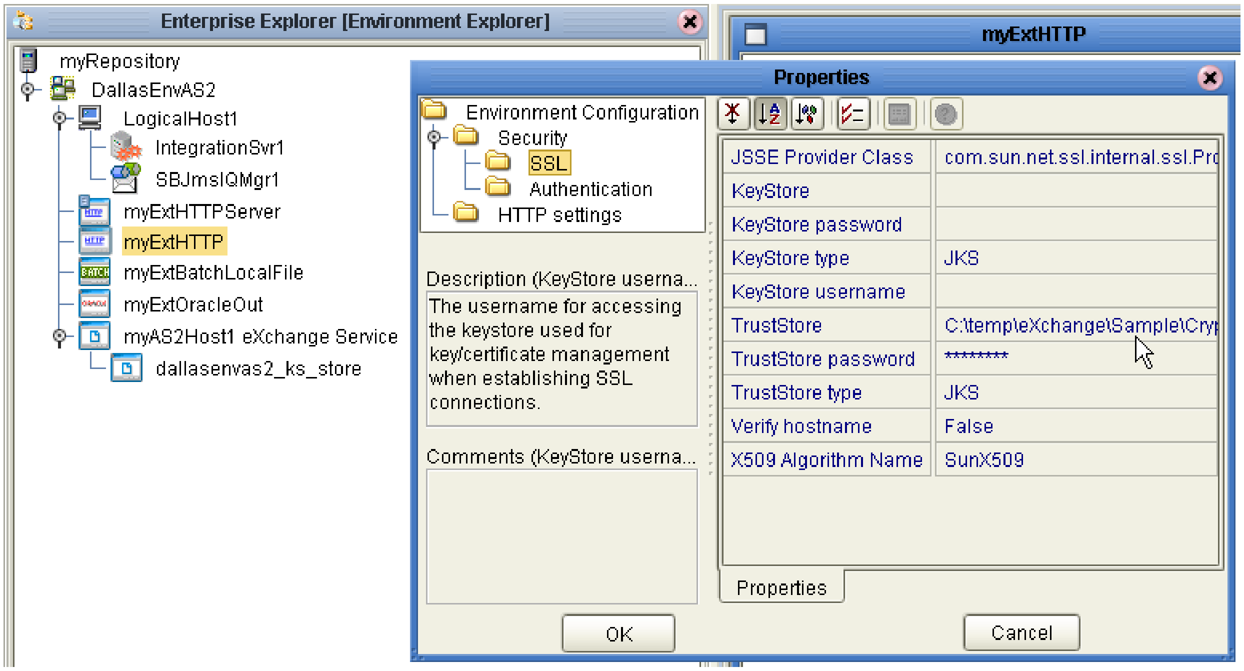
HTTPS means "HTTP over SSL" (secure sockets layer). If you want to use the HTTPS transport attributes definition, you must take additional steps to enable HTTPS.

- To enable an HTTP external to use SSL, it must have its SSL configuration settings edited appropriately. See the *eWay HTTPS Adapter User's Guide*, especially the "Configuring the HTTPS eWay" chapter. For an example of settings specific to eXchange, see the **procedure on page 83**.
- To enable the Integration Server to communicate using HTTPS, it needs to be associated with appropriate configuration settings. See the *eGate Integrator System Administration Guide* chapter on implementing security, especially the section "Configuring SSL Support". For an example of settings specific to eXchange, see the **procedure on page 84**.

To configure the HTTP external to use SSL

1. In the Environment Explorer tree, right-click the HTTP external and, on the popup context menu, click **Properties**.
2. In the Properties sheet, open Security and click **SSL**.
3. For TrustStore, provide the path and filename of the default truststore to be used when establishing SSL connections. For example:
   C:\temp\eXchange\Crypto\companyb.ssl.keystore
4. For TrustStore password, provide the correct password for this truststore.
   For example: **companyb** (see Figure 39). For TrustStoreType, keep the default: **JKS**

Figure 39 Configuring the SSL Properties of the HTTP External



Description (KeyStore userna...

The username for accessing the keystore used for key/certificate management when establishing SSL connections.

Comments (KeyStore userna...

JSSE Provider Class	com.sun.net.ssl.internal.ssl.Pro
KeyStore	
KeyStore password	
KeyStore type	JKS
KeyStore username	
TrustStore	C:\temp\eXchange\Sample\Cryp
TrustStore password	*********
TrustStore type	JKS
Verify hostname	False
X509 Algorithm Name	SunX509

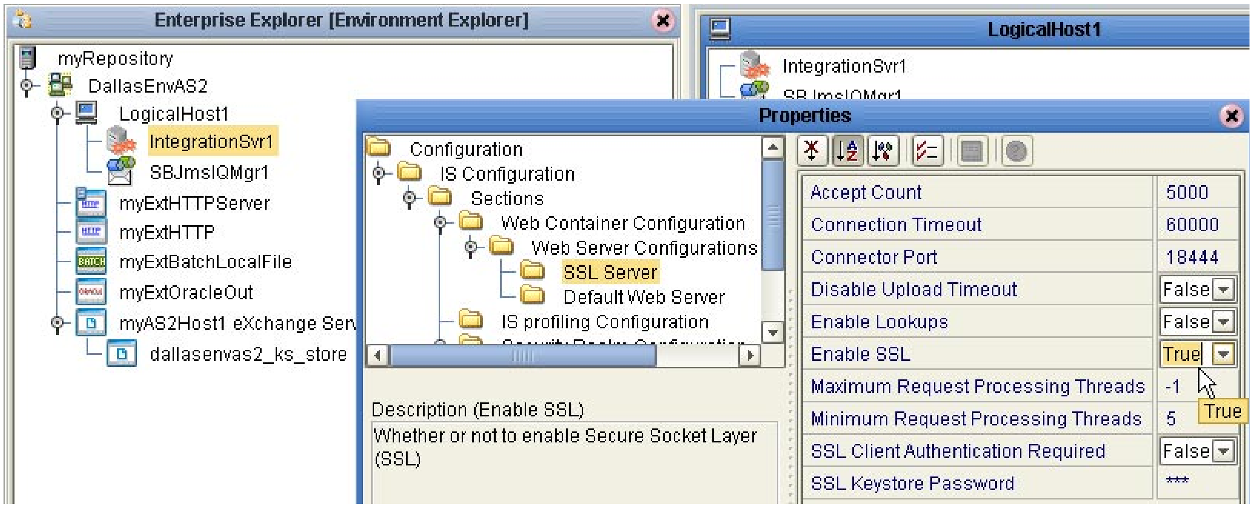
5. *AIX only.* Make the following additional changes for Logical Hosts running on AIX.
   - Change JSSE Provider Class from com.sun.net.ssl.internal.ssl.Provider to: **com.ibm.jsse.IBMJSSEProvider**
   - Change X509 Algorithm Name from SunX509 to (case-sensitive): **IbmX509**
6. Also make other configuration changes as needed. For more information, consult the *HTTP(S) eWay Intelligent Adapter User's Guide*, especially the "Setting HTTP(S) eWay Properties" chapter.
7. When you are finished, click OK.

Important: *Before you start the domain, ensure that its ...\keystore\ directory contains the correct **.keystore** file and that its alias (unless you edit server.xml otherwise) is **tomcat**. For an example, copy <jc512-repository>\server\**sbyn.keystore** to your <jc512-logicalhost>\keystore\ directory and rename it from sbyn.keystore to <yourIntegrationServername>**.keystore**. The keystore file must be of type JKS.*

To configure the Integration Server to use SSL

1. In the Environment Explorer tree, right-click **IntegrationSvr1** and click **Properties**
2. In the Properties sheet, open Configuration > IS Configuration > Sections > Web Container Configuration > **Web Server Configurations**
3. Do one of the following:
   - *(not recommended)* To have your Integration Server always use SSL web service connections by default, change the setting for **Enable SSL** to **True** and, if needed, change the setting for **SSL Client Authentication Required** to **False**.
   - *(recommended; see Figure 40)* Right-click **Web Server Configurations** and click **Create New Section**. Name the new section SSL Server. In its properties, set the **Enable SSL** property to **True** (see Figure 40) and the **SSL Client Authentication Required** property to **False**. Verify that the **Connector Port** property does not conflict with the default IS port settings (18000-18009) or any other ports.

Figure 40 Configuring the SSL Properties of the Integration Server



Accept Count	5000
Connection Timeout	60000
Connector Port	18444
Disable Upload Timeout	False
Enable Lookups	False
Enable SSL	True
Maximum Request Processing Threads	-1
Minimum Request Processing Threads	5
SSL Client Authentication Required	False
SSL Keystore Password	***

Description (Enable SSL)
Whether or not to enable Secure Socket Layer (SSL)

4. Also make other configuration changes as needed. For more information, see the *eGate Integrator User's Guide* chapter on **Environments** (especially the "Integration Servers" section) and the *eGate Integrator System Administration Guide* chapter on **Implementing Security** (especially the "Configuring SSL Support" section).
5. When you are finished, click **OK**.

5.4.3 Connecting the B2B Host to Oracle and LDAP Externals

These steps create a map that establishes a connection between the B2B Host and the two externals the B2BService communicates with: Oracle and LDAP.

To create and populate the B2B Host Connectivity Map

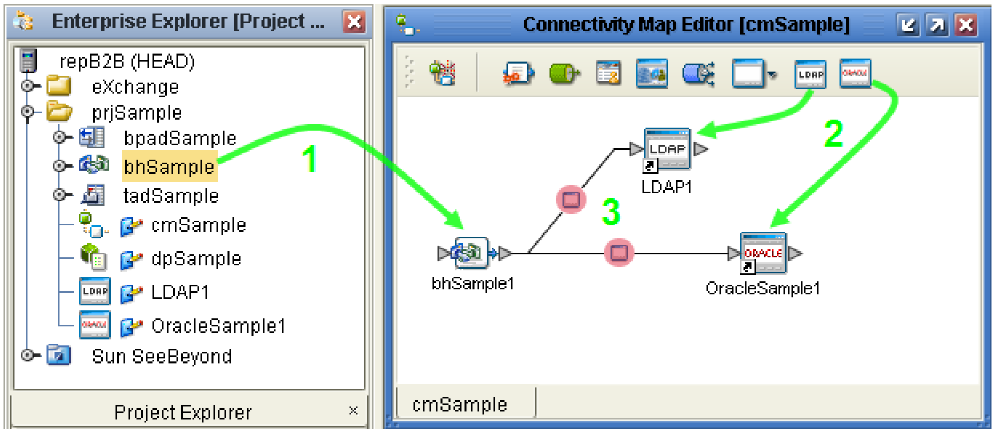
1. With the **Project Explorer** tab active, in the project tree, right-click the project and, on the popup context menu, point at click **New** and click **Connectivity Map**.

The project tree displays a new map, and the Connectivity Map Editor opens. Optionally, you can rename the map to something meaningful.

2 In the toolbar along the top of the canvas, click the **External Applications** tool and, from the drop-down list, select the checkboxes for **Oracle External Application** and for **LDAP External Application**.

3 From the project tree, drag your B2B Host onto left side of the canvas; from the toolbar, drag an Oracle external and then an LDAP external onto the right side of the canvas; (optionally, you can rename components to something meaningful); and then connect as shown in Figure 41.

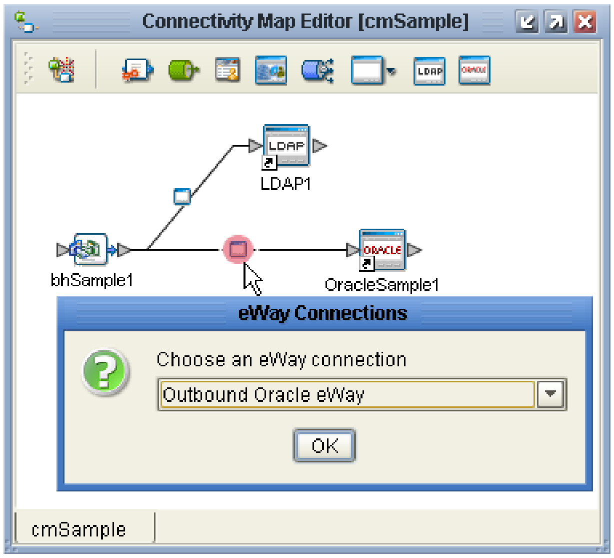
Figure 41 Connectivity Map Showing B2B Host Connected to Externals



4 For the B2B-LDAP connection: double-click the pink dot. When the Properties dialog box appears, make no changes, but click OK to set the default configuration.

5 For the B2B-Oracle connection: double-click the pink dot and configure the Oracle eWay as outbound (see Figure 42). When the Properties dialog box appears, make no changes, but click OK to set the default configuration.

Figure 42 Configuring the Oracle External as Outbound



Choose an eWay connection
Outbound Oracle eWay

Now that the map is populated and configured, the B2B Host is ready to be built.

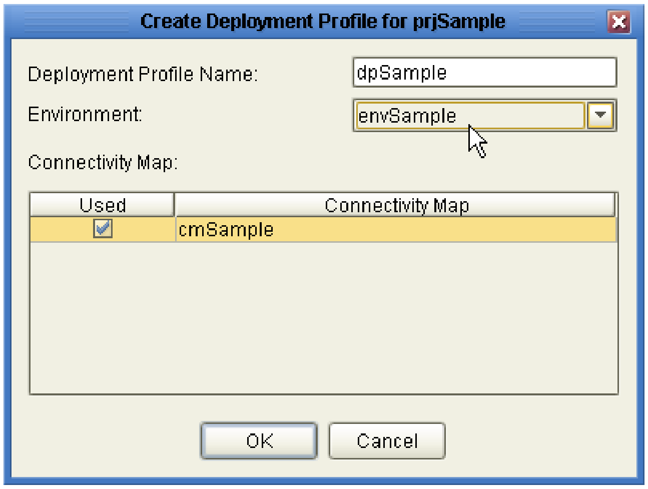
5.4.4 Building the B2B Host

These steps create, populate, and build a Deployment Profile that maps the resources named in the Connectivity Map to the resources provided in an Environment.

To create and build the Deployment Profile for the B2B Host

1. With the **Project Explorer** tab active, in the project tree, right-click the project and, on the popup context menu, point at click **New** and click **Deployment Profile**.
2. In the Create Deployment Profile dialog box, rename the deployment profile to something meaningful, point it at the correct Environment, and be sure that it is referencing the Connectivity Map that contains the B2B Host. See Figure 43.

Figure 43 Creating the Deployment Profile for the B2B Host Project



Deployment Profile Name: dpSample
Environment: envSample
Connectivity Map:

Used	Connectivity Map
☑	cmSample

3 In the Deployment Editor, click **Automap** to map the three components to their respective external systems (see Figure 44), and then click **Build**. See Figure 45.

Figure 44 Deployment Profile Before Components Are Mapped to Externals

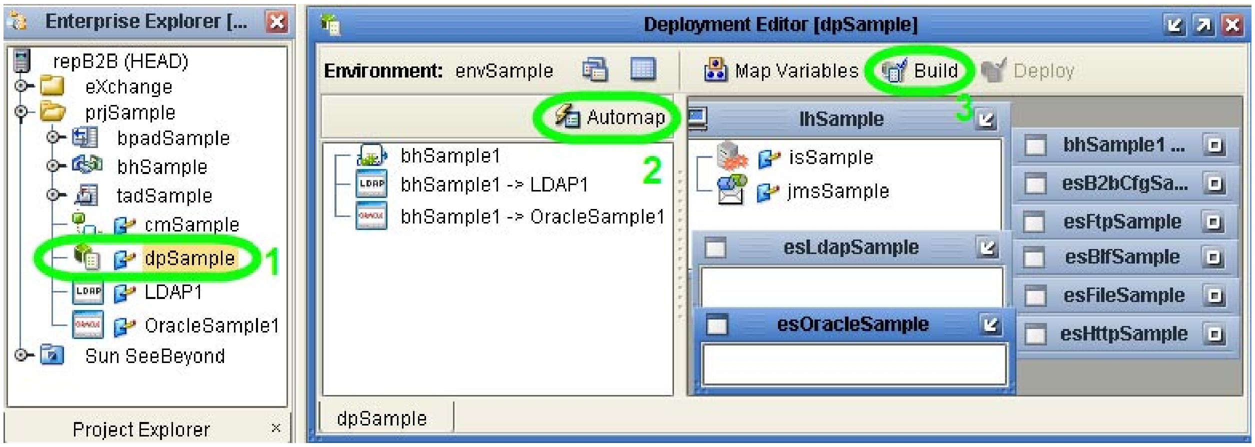
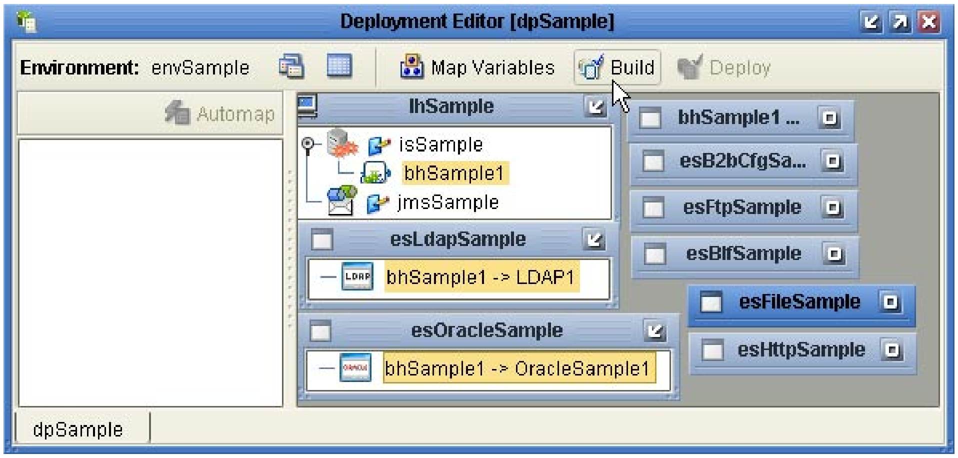
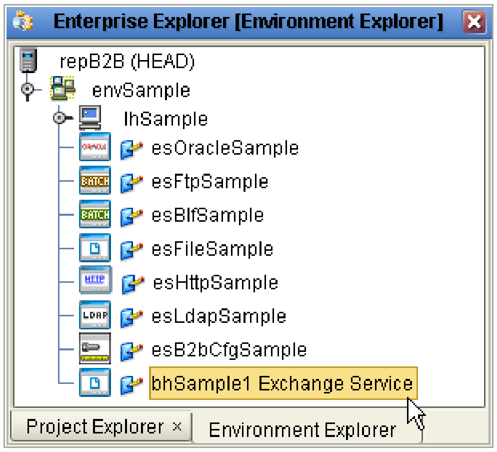


Figure 45 Deployment Profile Being Built



Result: In the Environment targeted by the Deployment Profile, an eXchangeService is added that contains all the metadata of the B2B Host; see Figure 46.

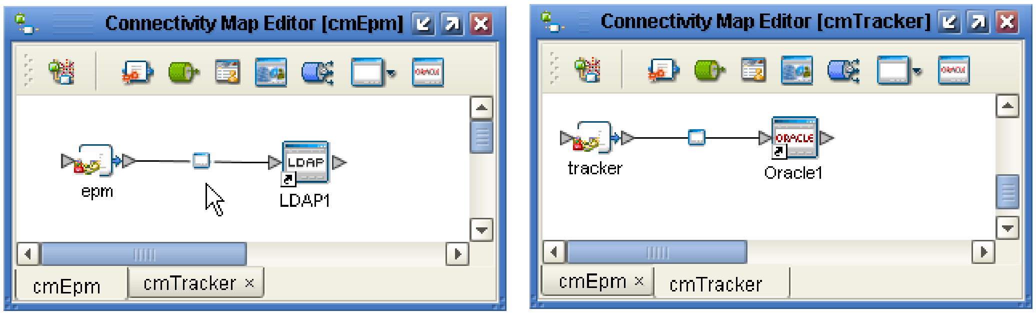
Figure 46 Environment with Newly Created eXchangeService



5.5 Building and Deploying the GUI Projects

These procedures build and deploy Deployment Profiles that map the resources named in the **cmEpm** and **cmTracker** Connectivity Maps to the externals in an Environment. The **cmEpm** map associates the **epm** application with an LDAP server; the **cmTracker** map associates the **epm** application with an Oracle server; see Figure 47.

Figure 47 Connectivity Maps for ePM and Tracker

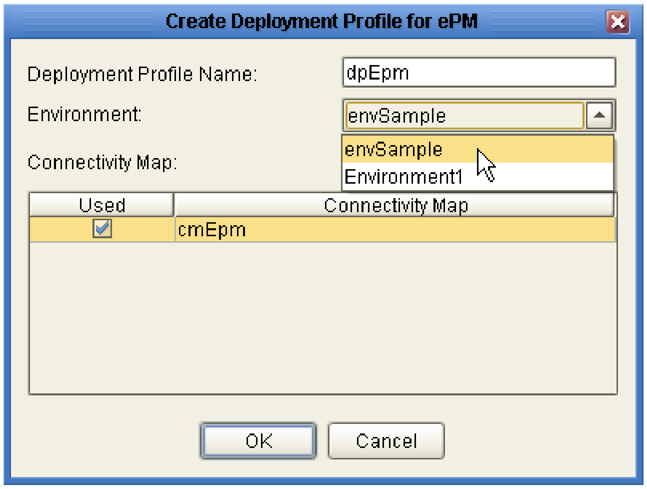

Connectivity Map Editor [cmEpm]
epm
LDAP
LDAP1
cmEpm
cmTracker
Connectivity Map Editor [cmTracker]
tracker
ORACLE
Oracle1
cmEpm
cmTracker

Before you begin: With the **Project Explorer** tab active, in the project tree, open the **eXchange > GUI** project. Also: If you have not already done so, start the domain associated with the Environment's Integration Server.

To build and deploy the Deployment Profile for the ePM Project

1. In the project tree, under **eXchange > GUI > ePM**, open **cmEpm** and configure its LDAP eWay appropriately for your operating environment.
2. Right-click the **ePM** project and, on the pop-up context menu, point at **New** and click **Deployment Profile**.
3. In the **Create Deployment Profile** dialog box, name the Deployment Profile to **dpEpm**, point it at the correct Environment, be sure that it is referencing **cmEpm** (the Connectivity Map for the ePM GUI), and then click OK. See Figure 48.

Figure 48 Deployment Profile for ePM Project



Create Deployment Profile for ePM

Deployment Profile Name: dpEpm

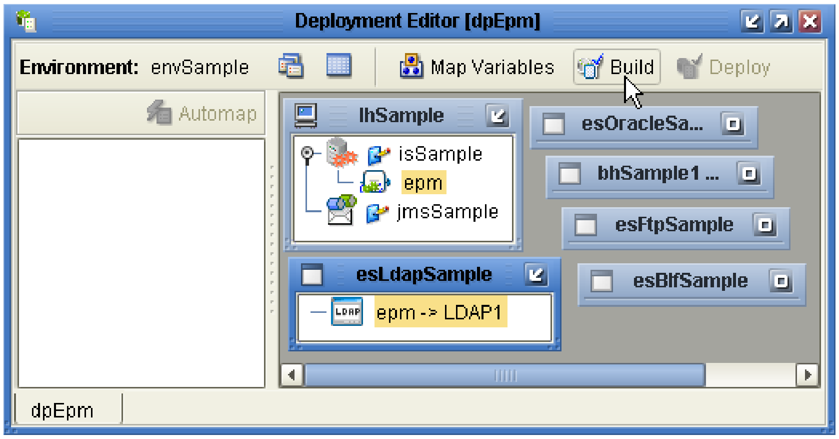
Environment: envSample

Connectivity Map:

Used	Connectivity Map
☑	cmEpm

4 In the Deployment Editor, click **Automap** to map the two components to their respective external systems, and then click **Build** (see Figure 49).

Figure 49 Deployment Profile dpEpm Being Built



Deployment Editor [dpEpm]

Environment: envSample

5 After the build step completes successfully (creating the EAR file), click **Deploy**.

Result: The ePM GUI project is built and deployed, and eXchange Partner Manager can now be started and used.

Message Tracker becomes available only after you build and deploy its Project.

To build and deploy the Deployment Profile for the Tracker Project

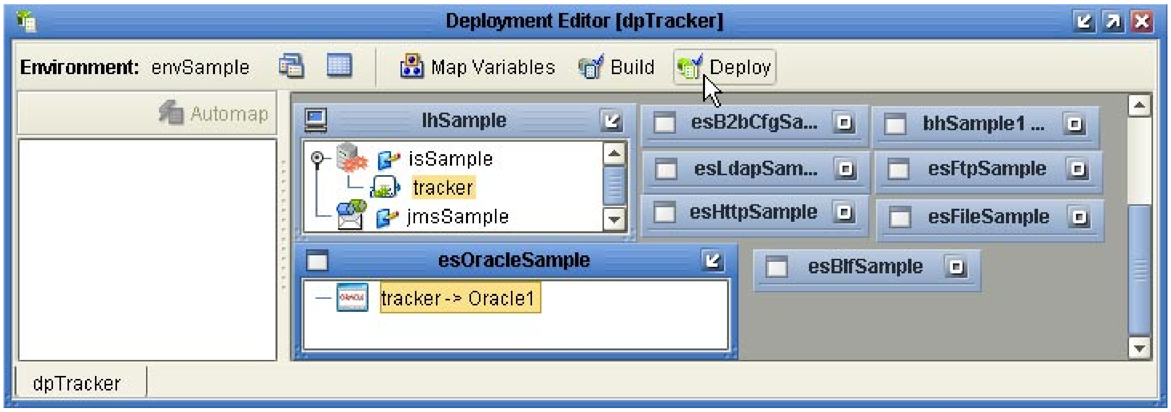
1 In the project tree, under **eXchange > GUI**, right-click the **Tracker** project and, on the pop-up context menu, point at **New** and click **Deployment Profile**.

2 In the **Create Deployment Profile** dialog box, name the Deployment Profile to **dpTracker**, point it at the correct Environment, be sure that it is referencing **cmTracker** (the Connectivity Map for the Message Tracker GUI), and then click OK.

3 In the Deployment Editor, click **Automap** to map the two components to their respective external systems, and then click **Build**.

4 After the build step completes successfully (creating the EAR file), click **Deploy**. See Figure 50.

Figure 50 Deployment Profile dpTracker Being Deployed

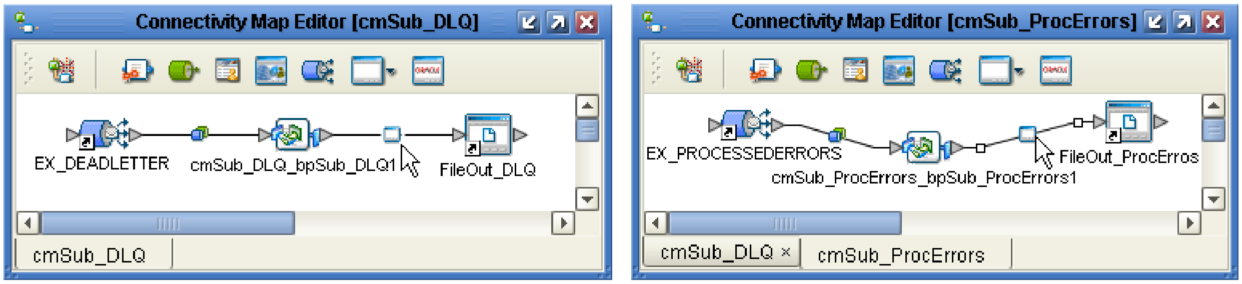


Result: The Message Tracker GUI project is built and deployed, and the Message Tracker application is now able to store and retrieve messages.

5.6 Building and Deploying the Error-Handling Projects

These procedures build and deploy Deployment Profiles that map the resources named in the **cmSub_DLQ** and **cmSub_ProcErrors** Connectivity Maps to the externals in an Environment. The **cmSub_DLQ** map uses a BP to receive messages from the JMS topic EX_DEADLETTER (sometimes called the "dead letter queue") and write them to a file; the **cmSub_ProcErrors** map uses a BP to receive messages from the JMS topic EX_PROCESSEDERRORS and write them to a file. See Figure 51.

Figure 51 Connectivity Maps for Sub_DLQ and Sub_ProcErrors



Before you begin: With the **Project Explorer** tab active, in the project tree, open the **eXchange > Errors** project. Also: If you have not already done so, start the domain associated with the Environment's Integration Server.

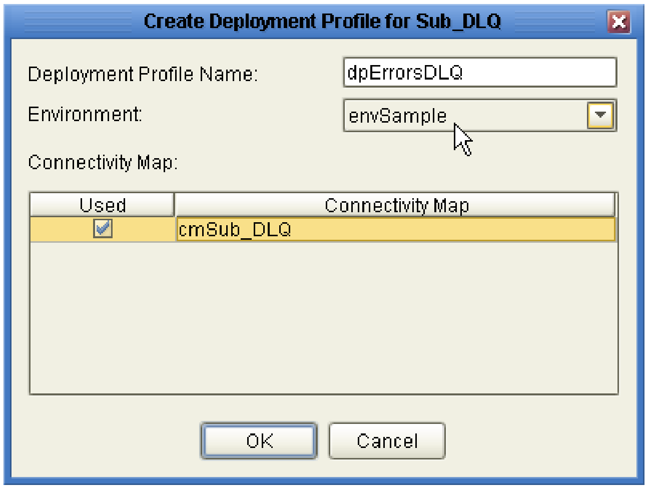
To build and deploy the Deployment Profile for the Sub_DLQ Project

1 In the project tree, under **eXchange > Errors > Sub_DLQ**, open **cmSub_DLQ** (see Figure 51) and ensure its File eWay is configured appropriately for your operating environment.

2 Right-click the **Sub_DLQ** project and, on the pop-up context menu, point at **New** and click **Deployment Profile**.

3 In the **Create Deployment Profile** dialog box, name the Deployment Profile to **dpErrorsDLQ**, point it at the correct Environment, be sure that it is referencing **cmSub_DLQ** (the Connectivity Map for the “dead letter” topic), and then click OK. See Figure 52.

Figure 52 Deployment Profile for Sub_DLQ Project


Create Deployment Profile for Sub_DLQ

Deployment Profile Name: dpErrorsDLQ

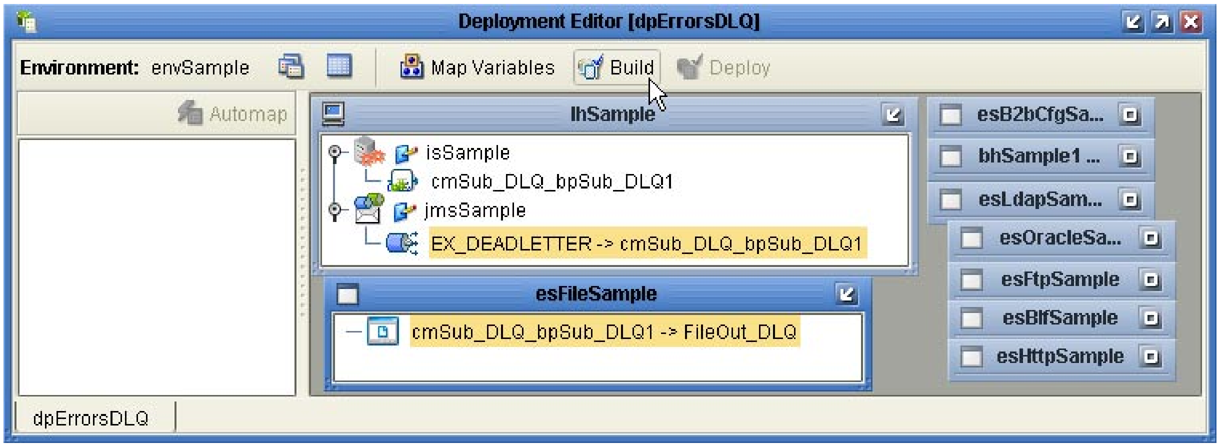
Environment: envSample

Connectivity Map:

Used	Connectivity Map
☑	cmSub_DLQ

4 In the Deployment Editor, click **Automap** to map the two components to their respective external systems, and then click **Build** (see Figure 53).

Figure 53 Deployment Profile dpErrorsDLQ Being Built


Deployment Editor [dpErrorsDLQ]

Environment: envSample

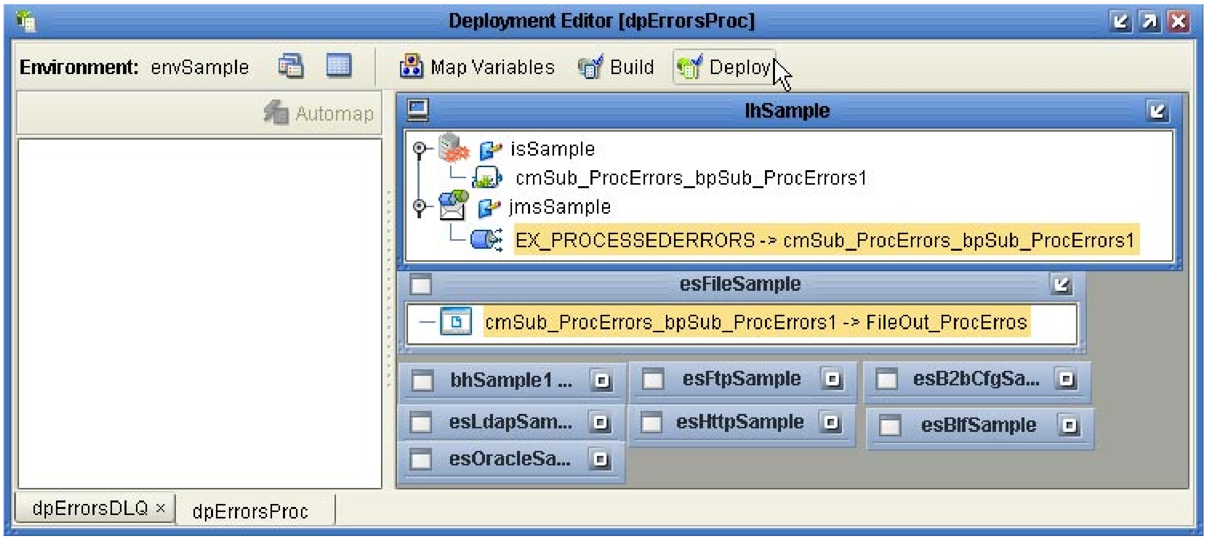
5 After the build step completes successfully (creating the EAR file), click **Deploy**.

Result: The **Sub_DLQ** project is built and deployed. At runtime, any undeliverable messages that cannot be processed will be copied and written to a file.

To build and deploy the Deployment Profile for the Sub_ProcErrors Project

1. In the project tree, under **eXchange > Errors > Sub_ProcErrors**, open **cmSub_ProcErrors** (see Figure 51) and ensure its File eWay is configured appropriately for your operating environment.
2. Right-click the **Sub_ProcErrors** project and, on the pop-up context menu, point at **New** and click **Deployment Profile**.
3. In the **Create Deployment Profile** dialog box, name the Deployment Profile to **dpErrorsProc**, point it at the correct Environment, be sure that it is referencing **cmSub_ProcErrors** (the Connectivity Map for the “processed errors” topic), and then click OK.
4. In the Deployment Editor, click **Automap** to map the two components to their respective external systems, and then click **Build**.
5. After the build step completes successfully (creating the EAR file), click **Deploy**. See Figure 54.

Figure 54 Deployment Profile dpErrorsProc Being Deployed



Result: The **Sub_ProcErrors** project is built and deployed. At runtime, undeliverable messages written to the processed-errors topic will be copied and written to a file.

Chapter 6

Using B2B Web Facilities

This chapter provides concepts, terminology, and step-by-step procedures for using the B2B Web-based facilities: eXchange Partner Manager (ePM) and Message Tracking.

Note: *Before you can use these facilities, you must have already built and deployed valid GUI projects in Enterprise Designer. If necessary, see* **"Building and Deploying the GUI Projects" on page 88**.

What's in This Chapter

General

Specific to ePM

Specific to Tracker

6.1 Providing Access to B2B Web Facilities

Access to ePM and Message Tracking is controlled by the Group List privileges set in SeeBeyond Integration Server Security Gateway. The applicable steps are provided below; for in-depth information, see the *eGate Integrator System Administration Guide*.

Before you begin: The associated GUI project or projects (for **epm** and/or **tracker**) must already be built and deployed, and the B2B Host's external servers—domain, LDAP, and Oracle—must already be running. (Note that you do ***not*** need to be running a Repository.)

To add or set up user access to B2B web facilities

1 Open a new browser window and point it a URL that has the following form:

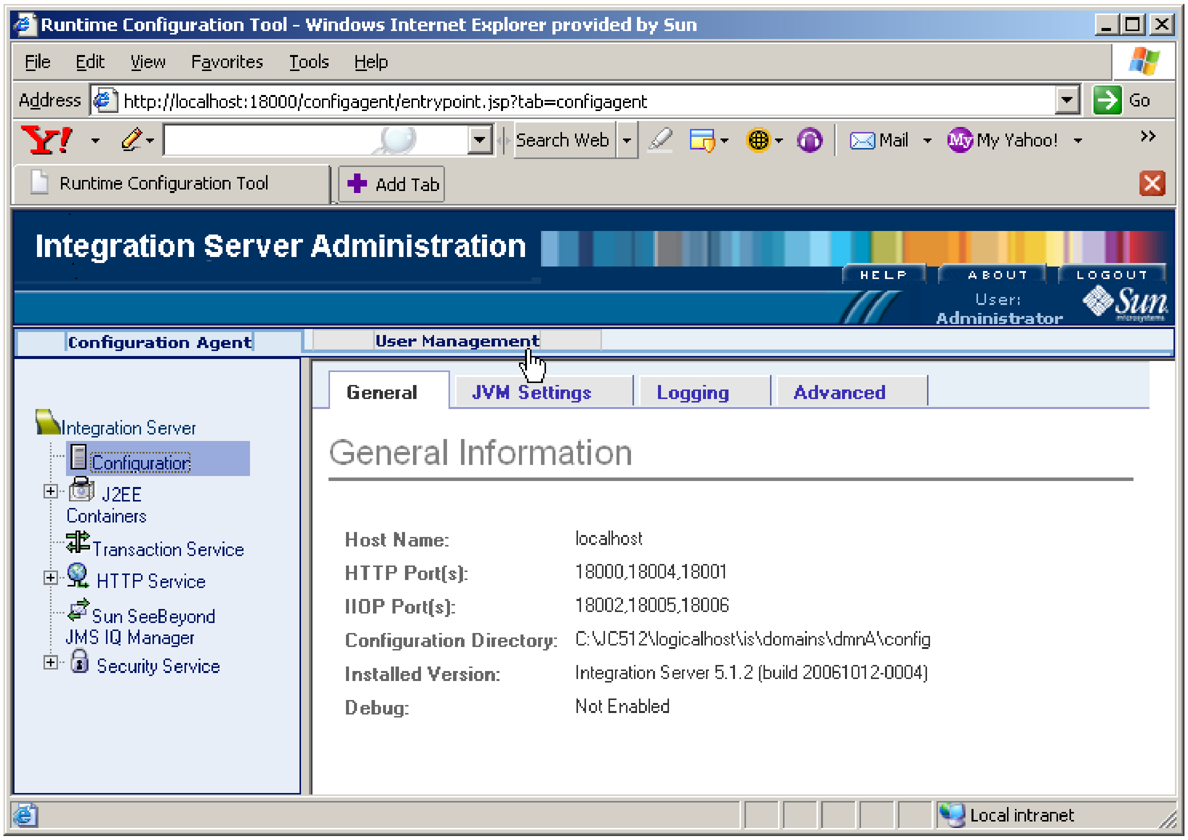
```
http://<machine-name>:<admin-port>
```

Thus, for a domain running on localhost port 18000, the URL would be as follows:

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

2 Log in with Administrator privileges and, in the Integration Server Administration panel, click the **User Management**. tab; see Figure 55.

Figure 55 Integration Server Administration: User Management


Runtime Configuration Tool - Windows Internet Explorer provided by Sun
File Edit View Favorites Tools Help
Address http://localhost:18000/configagent/entrypoint.jsp?tab=configagent
Go
Search Web
Mail
My Yahoo!
Runtime Configuration Tool
Add Tab
Integration Server Administration
HELP
ABOUT
LOGOUT
User: Administrator
Sun
Configuration Agent
User Management
Integration Server
Configuration
J2EE Containers
Transaction Service
HTTP Service
Sun SeeBeyond JMS IQ Manager
Security Service
General
JVM Settings
Logging
Advanced
Local intranet

General Information

Host Name: localhost
HTTP Port(s): 18000,18004,18001
IIOP Port(s): 18002,18005,18006
Configuration Directory: C:\JC512\logicalhost\is\domains\dmnA\config
Installed Version: Integration Server 5.1.2 (build 20061012-0004)
Debug: Not Enabled

3 In the **User Management** screen, either add a new user or edit an existing user.

4 In the **Add/Edit User** screen (see Figure 56), do the following:

A Supply a username.

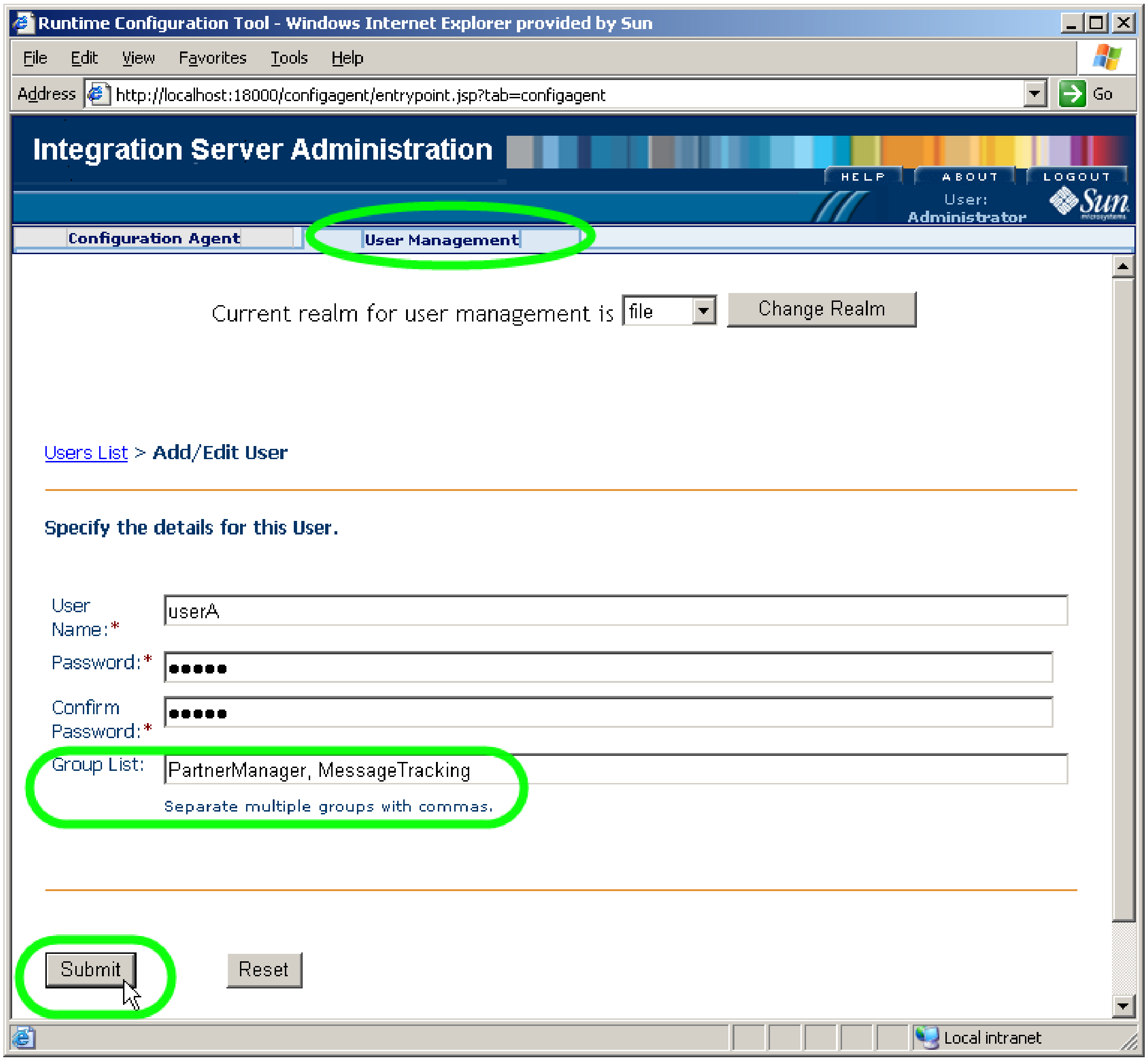
B Supply a valid password for the username.

C For **Group List**, enter either or both of the following (in addition to any other group privileges for this user):

- **PartnerManager, MessageTracking**

D When you are done, click **Submit**; see Figure 56.

Figure 56 Specifying PartnerManager and MessageTracking Privileges



Runtime Configuration Tool - Windows Internet Explorer provided by Sun
File Edit View Favorites Tools Help
Address
Go
Integration Server Administration
HELP
ABOUT
LOGOUT
User: Administrator
Configuration Agent
User Management
Change Realm
Users List > Add/Edit User
Submit
Reset
Local intranet

Address: http://localhost:18000/configagent/entrypoint.jsp?tab=configagent

Current realm for user management is file

Specify the details for this User.

User Name:* userA

Password:* ●●●●●

Confirm Password:* ●●●●●

Group List: PartnerManager, MessageTracking

Separate multiple groups with commas.

5 Repeat steps 3 and 4 as needed for any other users who need access to ePM or Message Tracker for this domain.

6.2 eXchange Partner Manager (ePM) Overview

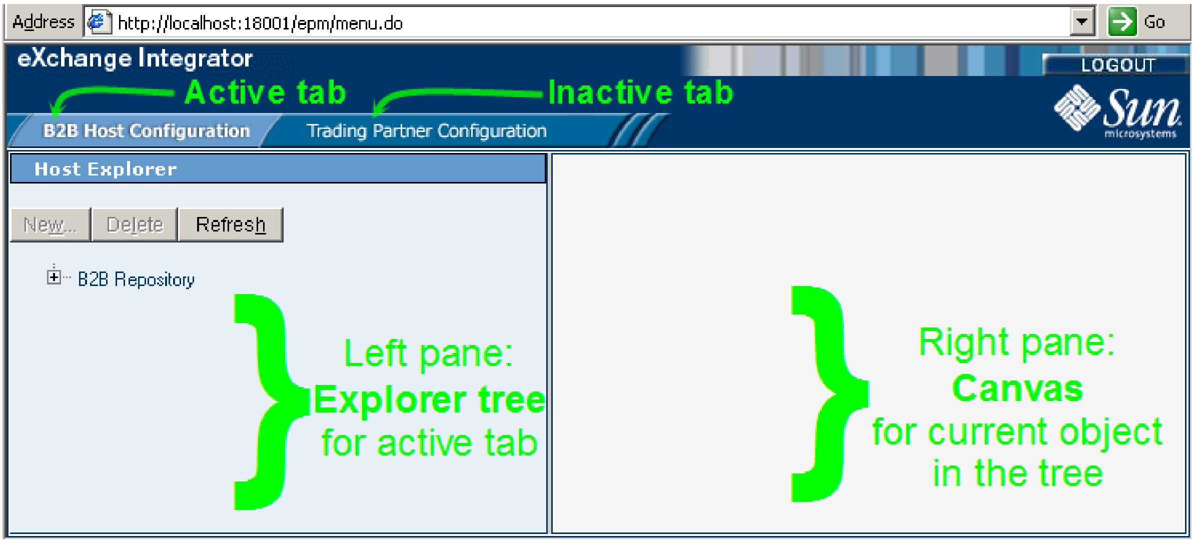
The purpose of ePM is to help you manage the parameters associated with your trading partners. Specific combinations of parameter setting are bound into a *transaction profile.* A transaction profile group consists of well-configured outbound and inbound business actions, associated with well-configured outbound and inbound delivery actions, such that each delivery action is tied to a well-configured transport.

The GUI provides a cascading series of defaults that can be inherited or overridden at various levels.

6.2.1 Operational Overview of the ePM GUI

After you sign in, ePM presents you with a layout comprising two tabs across the top, a tree view on the left, and a canvas on the right, where you view and modify settings. See Figure 57.

Figure 57 eXchange Partner Manager (ePM) — Initial State



- Tabs are modal: The display of the panes beneath depends on which tab is active.
- Tabs are not “sticky”: When you change modes to a new tab, all unsaved changes in the previous tab are discarded.
- The explorer pane on the left displays a hierarchical tree of containers and contents:
  - When the B2B Host Configuration tab is active, the Host Explorer tree displays B2B Hosts and their contents.
  - When the Trading Partner Configuration tab is active, the Trading Partner Explorer tree shows Trading Partners and their contents.
  - There is only topmost node (also called the *root* node). This is always named “B2B Repository” irrespective of the Repository name.

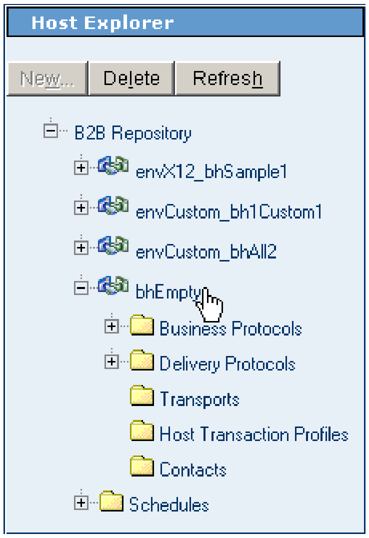
- In the explorer pane, the operation of the **New** button depends on the currently selected item in the tree: The button is available if and only if you can create a new child object of the currently selected item.
- The **Delete** button deletes the selected object and its children (if any).
- The **Refresh** button discards any unsaved changes and then displays an updated snapshot of the objects in the Repository.

Note: *Because many actions in ePM cause unsaved changes to be discarded, any time you add or change a value, it is good practice to verify the change and then click* ***Save****.*

6.2.2 Structural Overview of the ePM GUI

The hierarchy of the Host Explorer tree in the left pane has the following structure, illustrated in Figure 58.

Figure 58 ePM Structure of the Host Explorer



Root. There is exactly one root, always named **B2B Repository**. It cannot be deleted. The root contains a special **Schedules** folder as well as one or more B2B Hosts.

B2B Hosts are the second-highest object type. They can be created, imported, renamed, exported, and deleted. Each B2B Host contains:

- A **Business Protocols** folder, containing one or more business protocols. Each business protocol contains one or more *business action groups*. Each business action group contains at least two *business actions*—at least one *outbound* business action and at least one *inbound* business action.
- A **Delivery Protocols** folder, containing one or more delivery protocols. Each delivery protocol contains one or more *delivery action groups*. Each delivery action group contains at least two *delivery actions*—at least one outbound delivery action and at least one inbound delivery action.
- A **Transports** folder, containing one or more external transports.

- A **Host Transaction Profiles** folder, containing zero or more *transaction profile groups*. Each transaction profile group contains an outbound transaction profile and and an inbound transaction profile.
- A **Contacts** folder, containing zero or more contacts.

Schedules. The special **Schedules** folder contains zero or more schedules that can be referenced by other objects in the B2B Repository. A schedule can be regularly periodic, or it can be specified according to a daily, weekly, or monthly arrangement. Each schedule can be created, renamed, and deleted, and schedules can be imported or exported in aggregate.

Notes section. For each item that can be modified, the canvas provides a Notes area where you can provide free text. (For an illustration, see **Figure 61 on page 100**). Best practice is to use this area to the describe the item, and also to log every important change made to it.

6.3 Basic Operations in ePM

This section presents the following procedures:

- **To access ePM** on page 98
- **To locate a B2B Host in the explorer tree** on page 100
- **To rename a B2B Host or Trading Partner** on page 101
- **To export a B2B Host or Trading Partner** on page 101
- **To import/overwrite a B2B Host or Trading Partner** on page 101
- **To create, view, or configure a schedule** on page 102

Accessing ePM

Before you begin: In Enterprise Designer, you must have already set up at least one B2B Host and built it to create an eXchangeService (see **"Building a B2B Host" on page 80**), you must have already built and deployed the ePM GUI project (see **"Building and Deploying the GUI Projects" on page 88**). Furthermore, the Integration Server (domain) and the LDAP server for the associated eXchangeService must already be running. Note that the Repository does ***not*** need to be running.

To access ePM

1 Open a new browser window and point it a URL that has the following form:

```
http://<domain_machinename>:<baseport+1>/epm
```

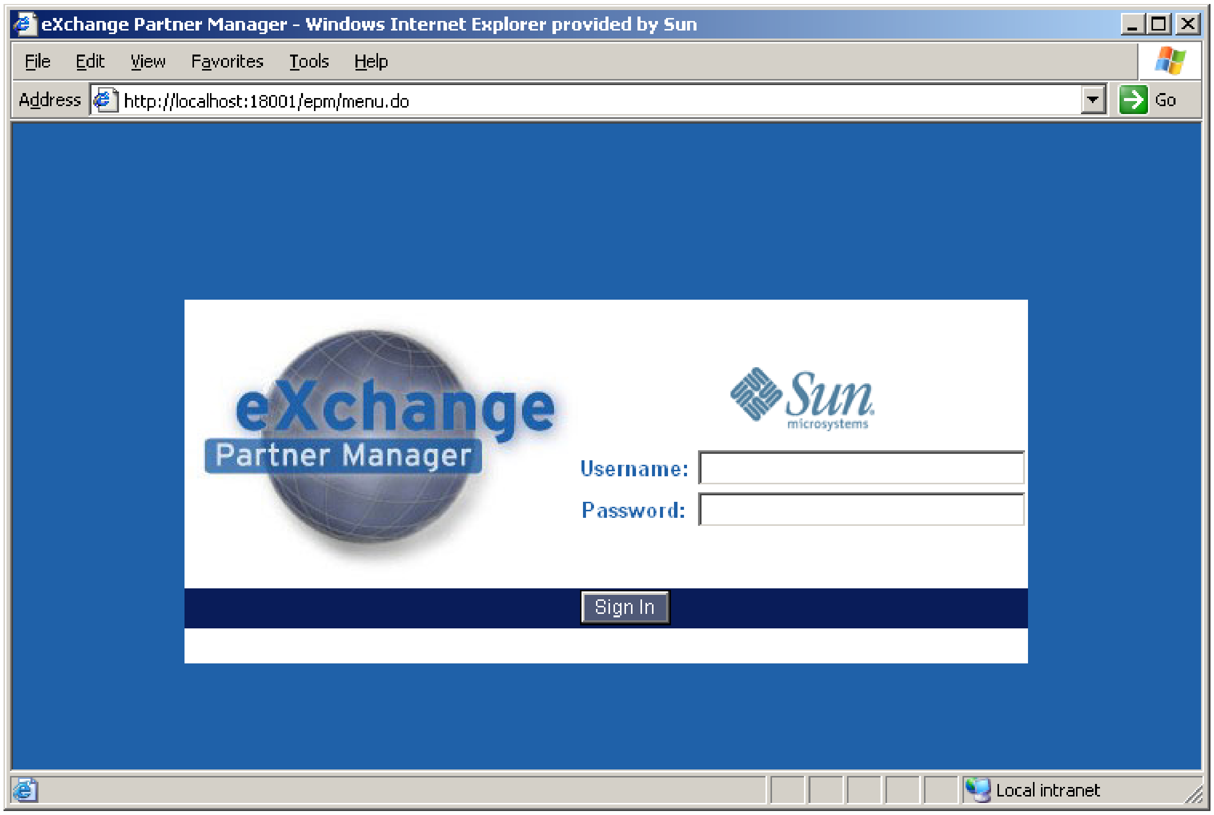
The http port is 1 greater than the base administrative port. Thus, for a domain running on localhost port 18000, the URL would be as follows:

```
http://localhost:18001/epm
```

Note: *This URL is case-sensitive. If the you receive a status 404 with the warning "Requested resource not available," ensure that in the final four characters of the URL—**/epm**—you entered the three letters "epm" in all-lowercase.*

Result: After a pause, the browser displays the eXchange Partner Manager sign-in screen. See Figure 59.

Figure 59 Sign-In Screen for eXchange Partner Manager (ePM)



eXchange Partner Manager - Windows Internet Explorer provided by Sun

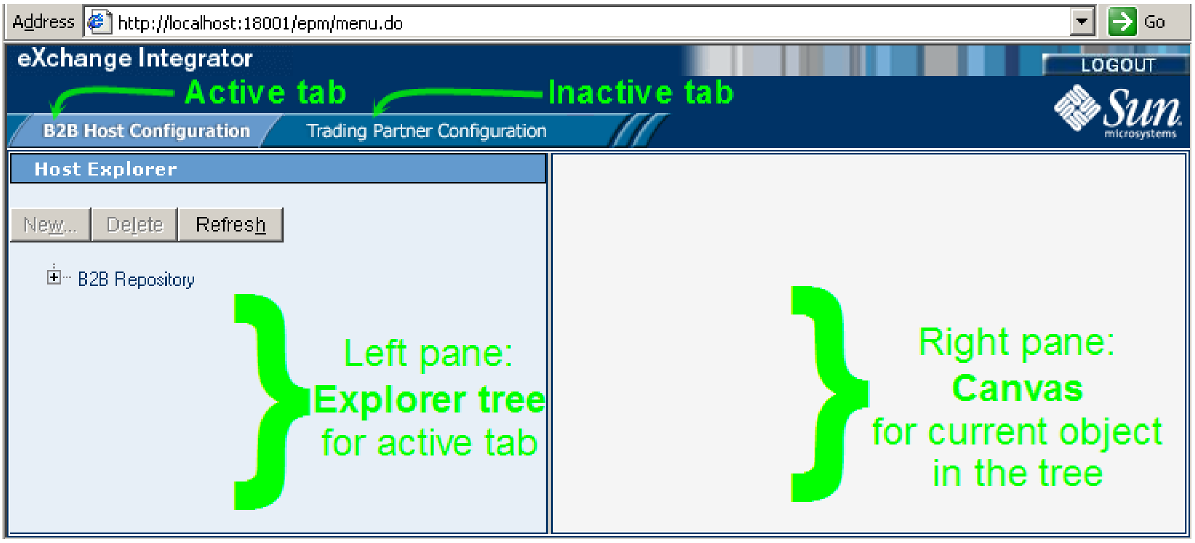
Address http://localhost:18001/epm/menu.do

2 Enter the appropriate user name and password and click **Sign In**.

Note: *Username/password combinations are set up in Integration Server Administration; see* **procedure on page 94**. *The only users authorized to access ePM are those that have a "PartnerManager" entry in their Group List.*

Result: After a pause, the window displays eXchange Partner Manager (ePM), a two-pane window with an explorer tree and a canvas. See Figure 60.

Figure 60 eXchange Partner Manager (ePM) — Initial State



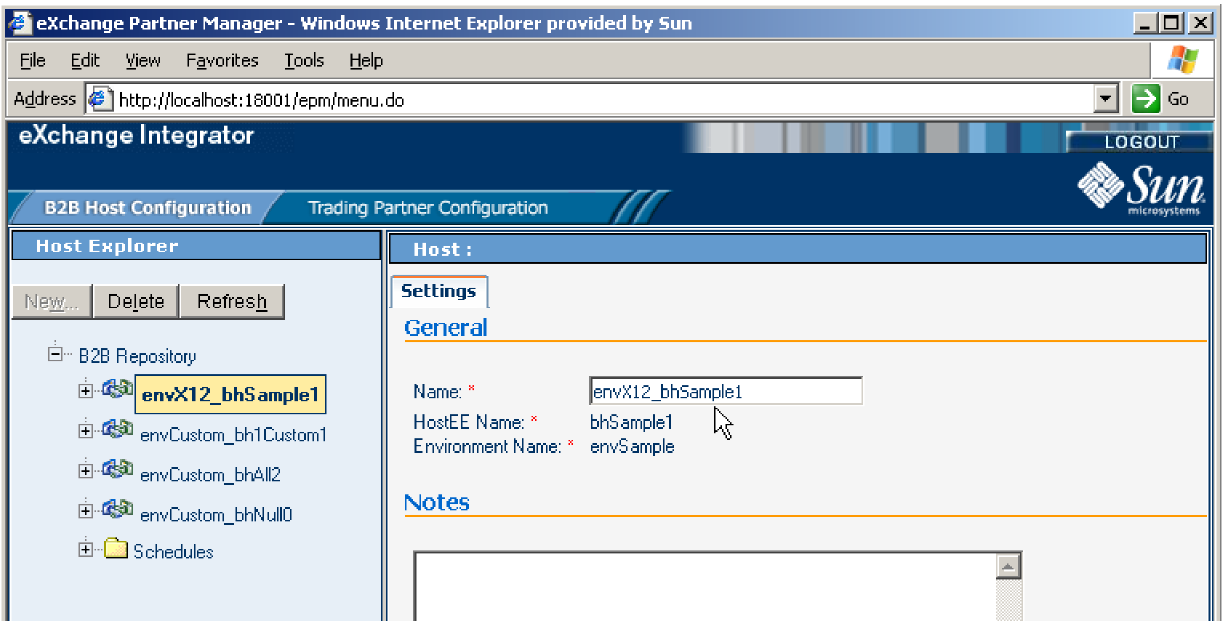
Basic B2B Host Operations in ePM

Various operations in ePM are not tied to any particular B2B Host or Trading Partner, such as: navigating the explorer tree; renaming a Host or TP; exporting and importing Hosts and TPs; and setting up schedules. This section contains procedures for each of these tasks.

To locate a B2B Host in the explorer tree

- Expand the root folder, **B2B Repository**, to see all the B2B Hosts it contains. Figure 61 provides an illustration.

Figure 61 ePM Repository Showing B2B Hosts



Name: * envX12_bhSample1
HostEE Name: * bhSample1
Environment Name: * envSample

You can rename any B2B Host from its default name (formed by concatenating the Environment with the B2B Host instance as it appears on the Connectivity Map) to a name that is meaningful to you. You can also rename any Trading Partner.

To rename a B2B Host or Trading Partner

1. Select the Host or TP and, in the right pane, under the **Settings** tab, in the **General** parameters section, edit the value of **Name**.
2. When you are finished, click **Save**.

Note: *Because many actions in ePM cause unsaved changes to be discarded, any time you add or change a value, it is good practice to verify the change and then click **Save**.*

To export a B2B Host or Trading Partner

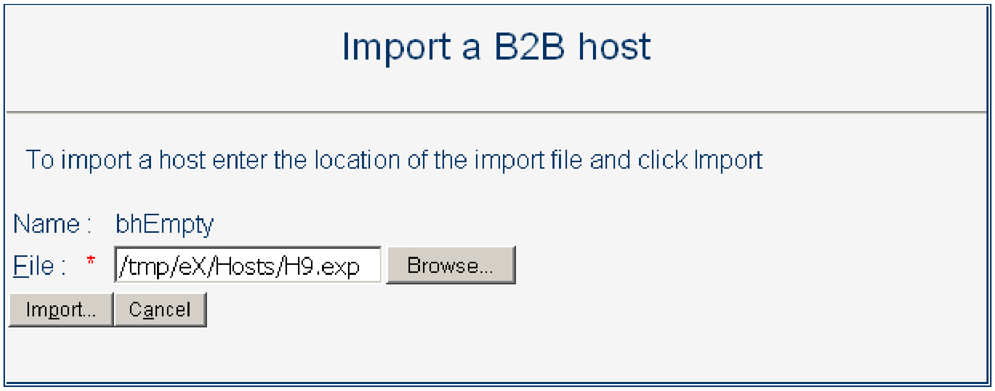
1. In the explorer tree on the left, select the Host or TP you want to export.
2. In the canvas on the right, at the bottom of the pane, click **Export**.
3. In the **File Download** dialog box, click **Save**.
4. In the **Save As** dialog box, provide a location and filename and then click **Save**.

Result: An XML file of filetype **.exp** is saved to the specified location.

To import/overwrite a B2B Host or Trading Partner

1. In the explorer tree on the left, select the Host or TP you want to overwrite.
2. In the canvas on the right, at the bottom of the pane, click **Import**.
3. In the **Import a B2B Host** [**Trading Partner**] pane (see Figure 62) click **Browse**, navigate to and open the **.exp** file you want to import, and then click **Import**.

Figure 62 Importing a B2B Host



Import a B2B host

To import a host enter the location of the import file and click Import

Name : bhEmpty

File : * /tmp/eX/Hosts/H9.exp

4. When you are finished, click **Save**.

Note: *Because many actions in ePM cause unsaved changes to be discarded, any time you add or change a value, it is good practice to verify the change and then click **Save**.*

To create, view, or configure a schedule

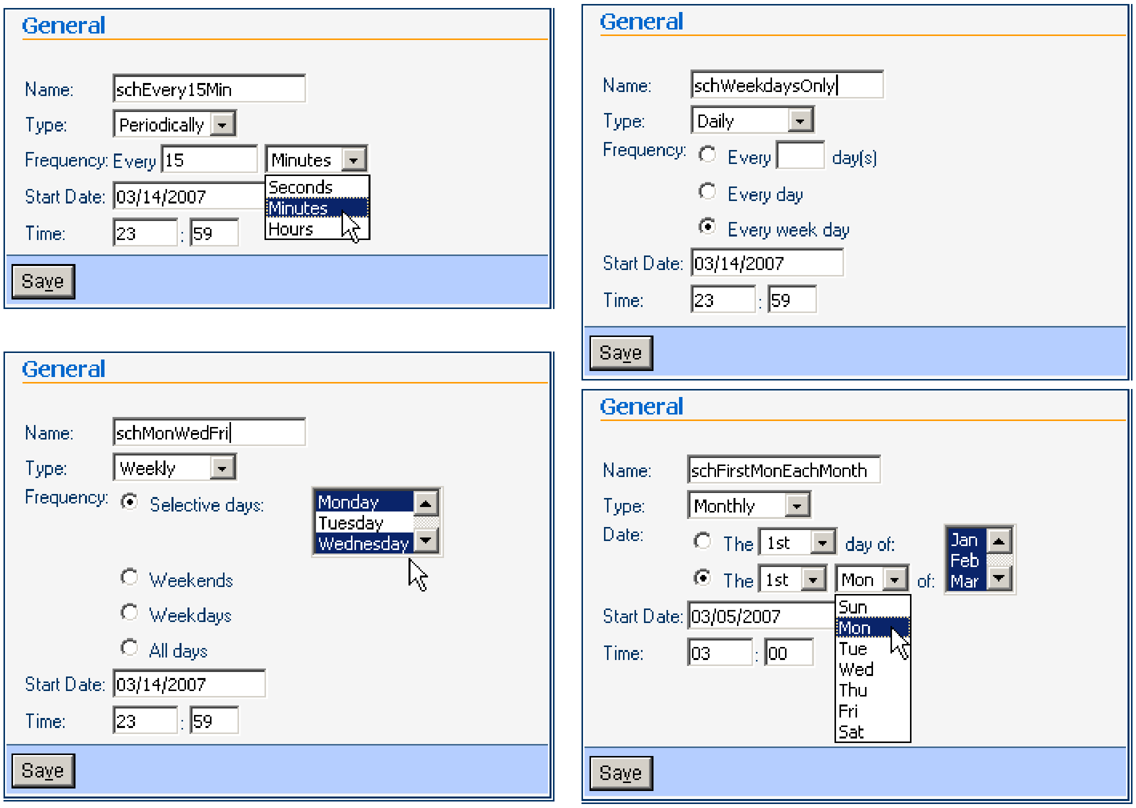
1 In the **B2B Host Configuration** tab, in the Host Explorer tree, select the **Schedules** folder at the bottom of the tree and do one of the following:

- To create a new schedule: Click **New** and, in the canvas on the right, enter a name and provide values for Type, Frequency, Start Date, and Time.
- To view or configure an existing schedule: Open the **Schedules** folder and click the schedule you want to display or modify.

2 Configure the schedule's type and frequency as follows (see Figure 63).

- For several-times-a-day schedule: Set **Type** to **Periodically** and, for **Frequency**, specify how many seconds, (or minutes, or hours) elapse between events.
- For a per-day schedule: Set **Type** to **Daily** and, for **Frequency**, specify whether the event occurs every few days, every day, or every weekday.
- To configure a per-week schedule: Set **Type** to **Weekly** and, for **Frequency**, specify either that the event occurs on the day(s) in the list (use [SHIFT]-click to select a range of days and [CTRL]-click to include or exclude a particular day); or, alternatively, that it occurs weekends only, weekdays only, or every day.
- To configure a per-month schedule: Set **Type** to **Monthly** and, for **Frequency**, specify whether the event occurs on a particular date or day within the month(s) specified in the list. In the list of months, use [SHIFT]-click to select a range of months and [CTRL]-click to include or exclude a particular month.

Figure 63 Configuring Schedules



General

Name: schEvery15Min
Type: Periodically
Frequency: Every 15 Minutes
Seconds
Minutes
Hours
Start Date: 03/14/2007
Time: 23 : 59
Save

General

Name: schWeekdaysOnly
Type: Daily
Frequency: ○ Every day(s)
○ Every day
● Every week day
Start Date: 03/14/2007
Time: 23 : 59
Save

General

Name: schMonWedFri
Type: Weekly
Frequency: ● Selective days: Monday Tuesday Wednesday
○ Weekends
○ Weekdays
○ All days
Start Date: 03/14/2007
Time: 23 : 59
Save

General

Name: schFirstMonEachMonth
Type: Monthly
Date: ○ The 1st day of: Jan Feb Mar
● The 1st Mon of:
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Start Date: 03/05/2007
Time: 03 : 00
Save

3. Specify an appropriate start date and time for the schedule.
4. When you are finished, click **Save**.

6.4 Viewing and Setting Business Protocol Parameters

Each B2B Host in ePM is automatically presupplied with the *PassThrough* business protocol. The PassThrough protocol provides no parameter settings beyond the minimum required to pass business data through without operating on it.

In addition to the Passthrough protocol, each B2B Host may contain one or more other business protocols, whose parameter types and possible settings depend on the way in which their business protocol attribute definitions (BPADs) were set up in Enterprise Designer:

- Parameters for Protocol Manager products, such as the X12 Protocol Manager or the HIPAA Protocol Manager, are described in the corresponding Protocol Manager User's Guide.
- Parameters for custom business protocols are determined by the custom BPADs defined in Enterprise Manager. For complete information on the relationship between the BPAD settings and ePM parameters, refer to **Appendix B on page 188**.

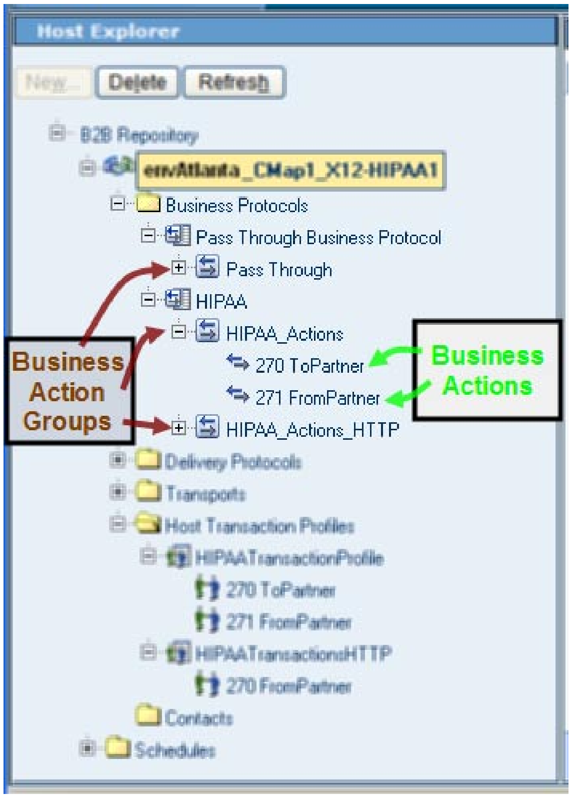
Presupplied business protocol parameter settings divide into the following groupings:

- Protocol-Outbound and ACK - Protocol-Outbound: Parameters determined by business attribute definitions (BADs) whose direction is ToPartner or Both. See **"Protocol-Outbound (ToPartner) Parameter and ACK Settings" on page 107**.
- Protocol-Inbound and ACK - Protocol-Inbound: Parameters determined by business attribute definitions (BADs) whose direction is FromPartner or Both. See **"Protocol-Inbound (FromPartner) Parameter and ACK Settings" on page 111**.
- Batch - Outbound and ACK - Batch-Outbound: Parameters determined by enveloping attribute definitions (EADs) whose direction is ToPartner or Both. See **"Batch-Outbound (ToPartner) Parameter and ACK Settings" on page 114**.
- Batch - Inbound and ACK - Batch-Inbound: Parameters determined by enveloping attribute definitions (EADs) whose direction is FromPartner or Both. Although the PassThrough business protocol does not supply any batch-inbound parameters, other business protocols might do so.

6.4.1 Configuring Business Action Groups and Business Actions

A *business action* is a particular business message type, such as an X12 850 (purchase order request) or a HIPAA 270 (eligibility coverage / benefit inquiry). A *business action group* is a collection of business actions that you want to choreograph. For example, one action group might contain simply an outbound notification message; another might contain a request message in both inbound and outbound directions, along with all possible forms of reply messages in both directions. See Figure 64.

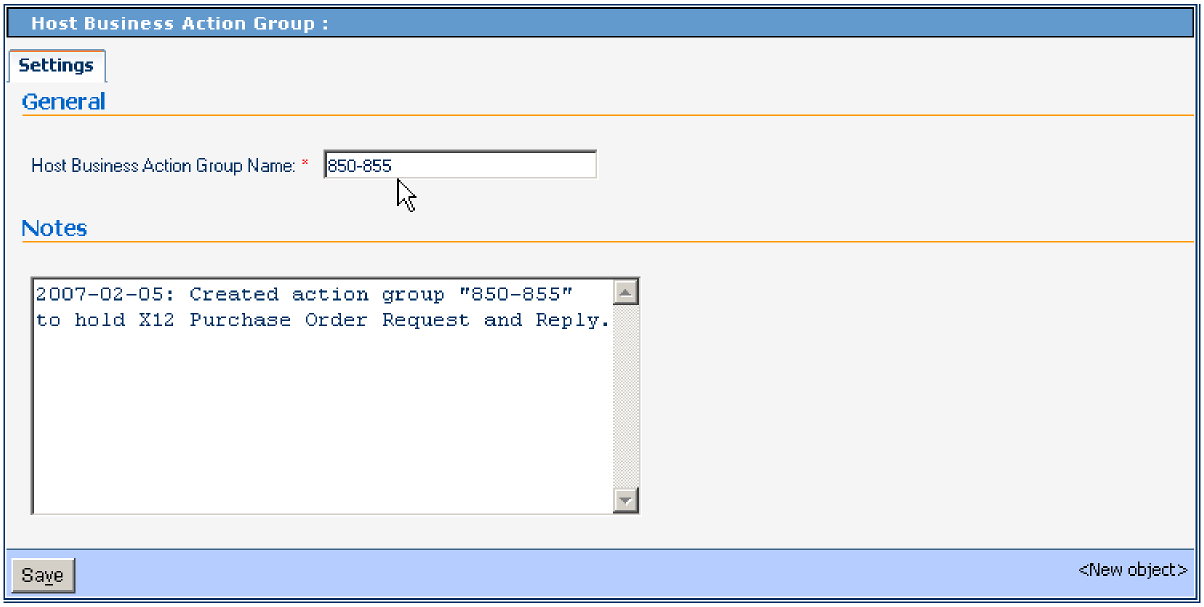
Figure 64 Business Action Groups and Business Actions



To create and configure a business action group

1. In the Host Explorer tree, click the business protocol under which you want to create an action group. This could be the **Pass Through Business Protocol**, a business protocol supplied with a Protocol Manager, or a custom business protocol, depending on the attribute definitions you want to use.
2. Click **New**.
3. In the Host Business Action Group pane on the right, enter a name for the action group. See Figure 65.

Figure 65 Creating a New Business Action Group



Host Business Action Group Name: * 850-855

2007-02-05: Created action group "850-855" to hold X12 Purchase Order Request and Reply.

4 Click **Save**.

Result: The new business action group appears in the explorer tree on the left and, in the right pane, a confirmation message is displayed in a new tab: **Overrides**.

5 As required by the nature of the business messages, modify parameter settings as needed by overriding the default values shown. Any modifications you make will be inherited by business actions and transaction profiles that use this action group.

6 Click **Save**.

7 Repeat as needed to create additional action groups for the same business protocol.

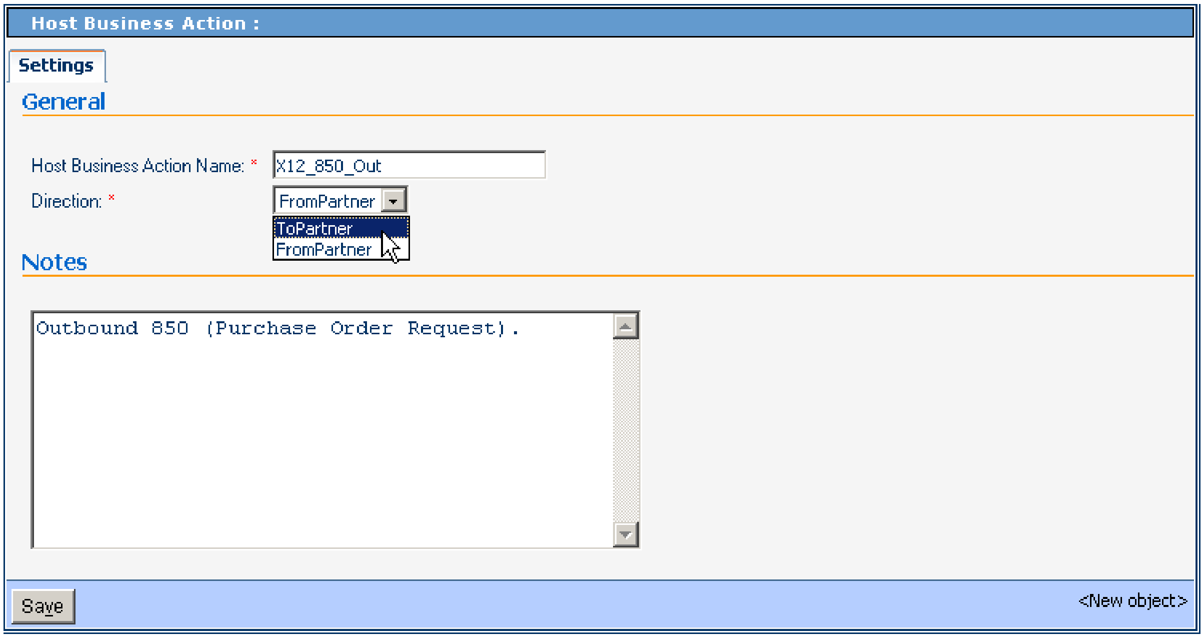
To create and configure a business action

1 In the Host Explorer tree, click the business action group under which you want to create a business action group.

2 Click **New**.

3 In the Host Business Action Group pane on the right, enter a name for the action group and, for Direction, choose either FromPartner or ToPartner. See Figure 66.

Figure 66 Creating a New Business Action



Host Business Action Name: * X12_850_Out

Direction: * FromPartner

Outbound 850 (Purchase Order Request).

4 Click **Save**.

Result: The new business action appears in the explorer tree on the left and, in the right pane, a confirmation message is displayed in a new tab: **Overrides**.

5 As required by the nature of the business message, modify parameter settings as needed by overriding the default values shown. Any modifications that you make will be inherited by transaction profiles that use this business action.

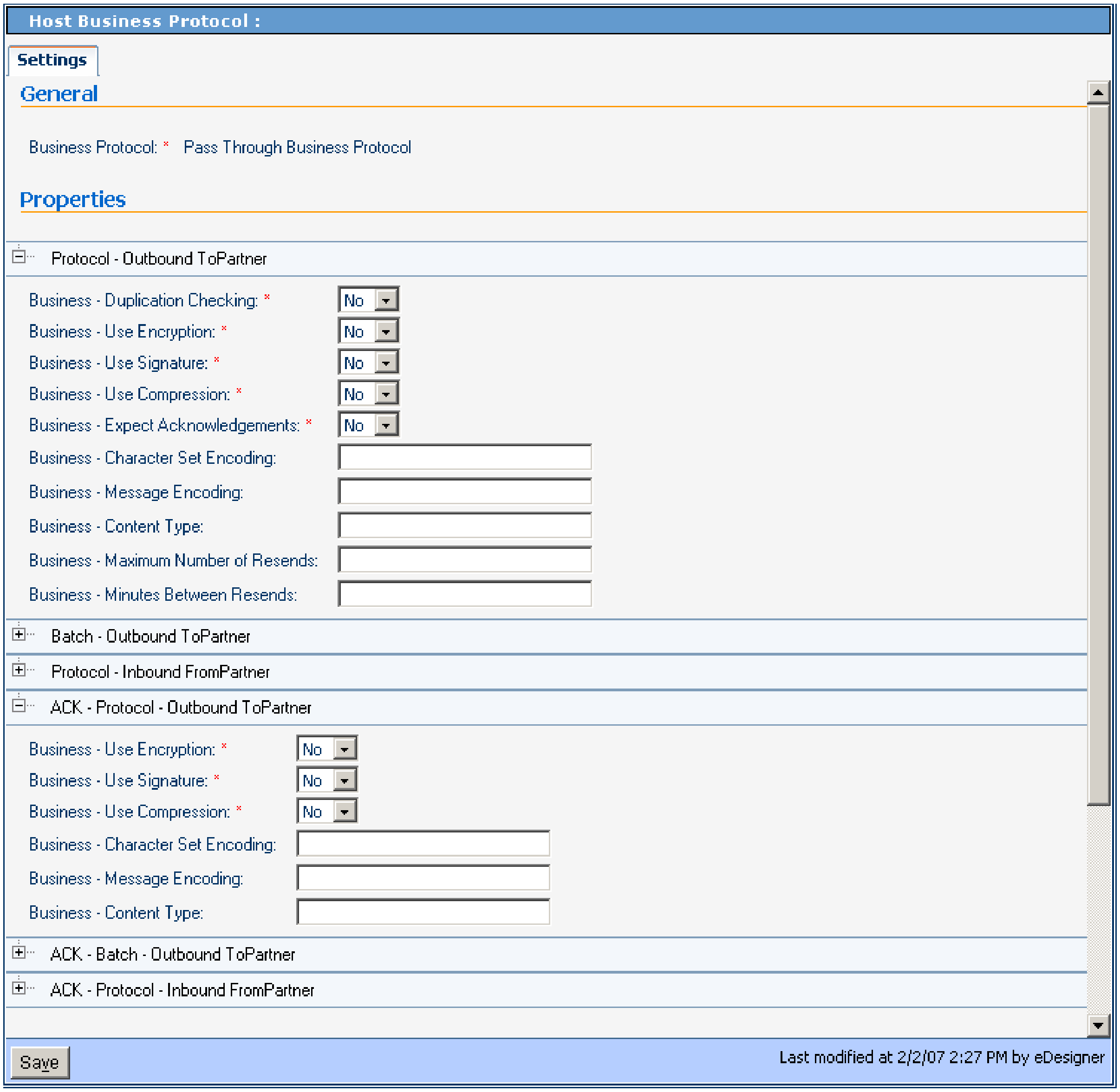
6 Click **Save**.

7 Repeat as needed to create additional business actions in the same action group.

6.4.2 Protocol-Outbound (ToPartner) Parameter and ACK Settings

This section lists and describes business protocol-outbound=ToPartner parameter settings and corresponding acknowledgment settings. For an illustration of how these appear in ePM, see Figure 67.

Figure 67 Protocol-Outbound (ToPartner) Parameter and ACK Settings



General

Business Protocol: * Pass Through Business Protocol

Properties

⊟ Protocol - Outbound ToPartner

Business - Duplication Checking: * No
Business - Use Encryption: * No
Business - Use Signature: * No
Business - Use Compression: * No
Business - Expect Acknowledgements: * No
Business - Character Set Encoding:
Business - Message Encoding:
Business - Content Type:
Business - Maximum Number of Resends:
Business - Minutes Between Resends:

⊞ Batch - Outbound ToPartner

⊞ Protocol - Inbound FromPartner

⊟ ACK - Protocol - Outbound ToPartner

Business - Use Encryption: * No
Business - Use Signature: * No
Business - Use Compression: * No
Business - Character Set Encoding:
Business - Message Encoding:
Business - Content Type:

⊞ ACK - Batch - Outbound ToPartner

⊞ ACK - Protocol - Inbound FromPartner

Last modified at 2/2/07 2:27 PM by eDesigner

Business - Duplication Checking

Description

Whether or not outbound messages are checked for duplication.

Settings

Yes or **No**

Default

No

Business - Use Encryption

Description

Whether or not outbound messages are encrypted.

Settings

Yes or **No**

Default

No

Business - Use Signature

Description

Whether or not outbound messages are electronically signed for authentication.

Settings

Yes or **No**

Default

No

Business - Use Compression

Description

Whether or not outbound messages are compressed.

Settings

Yes or **No**

Default

No

Business - Expect Acknowledgments

Description

Whether or not unacknowledged outbound messages are considered errors.

Settings

Yes or **No**

Default

No

Business - Character Set Encoding

Description

The character set used to encode the text of the outgoing message; optional.

Leave blank to use the default encoding.

Settings

A string representing a valid character set encoding (such as **ASCII** or **UTF-8**); or empty.

Default

(empty)

Business - Message Encoding

Description

The message encoding by which the native representation of the data can be mapped to a representation that can be used for exchange over 7-bit mail transport protocols; optional.

Leave blank to use the default encoding.

Settings

A string representing a valid message encoding (**binary**, **7bit**, **8bit base64**, **SMIME**, and so forth); or empty.

Default

(empty)

Business - Content Type

Description

(reserved)

Settings

(reserved)

Default

(empty)

Business - Maximum Number of Resends

Description

The maximum number of resend tries permitted; optional.

Does not apply to acknowledgments.

Settings

Nonnegative integer. However, if you are expecting a response or acknowledgment, set this to a positive value.

If this parameter and the following one are both set to **0**, then no correlation will be started on the message; consequently, in this case, an incoming response or acknowledgment will cause an exception to be thrown.

Default

(empty)

Business - Minutes Between Resends

Description

Number of minutes to wait between resends; optional.

Does not apply to acknowledgments.

Settings

Nonnegative integer. However, if you are expecting a response or acknowledgment, set this to a positive value.

If this parameter and the preceding one are both set to **0**, then no correlation will be started on the message; consequently, in this case, an response or acknowledgment will cause an exception to be thrown.

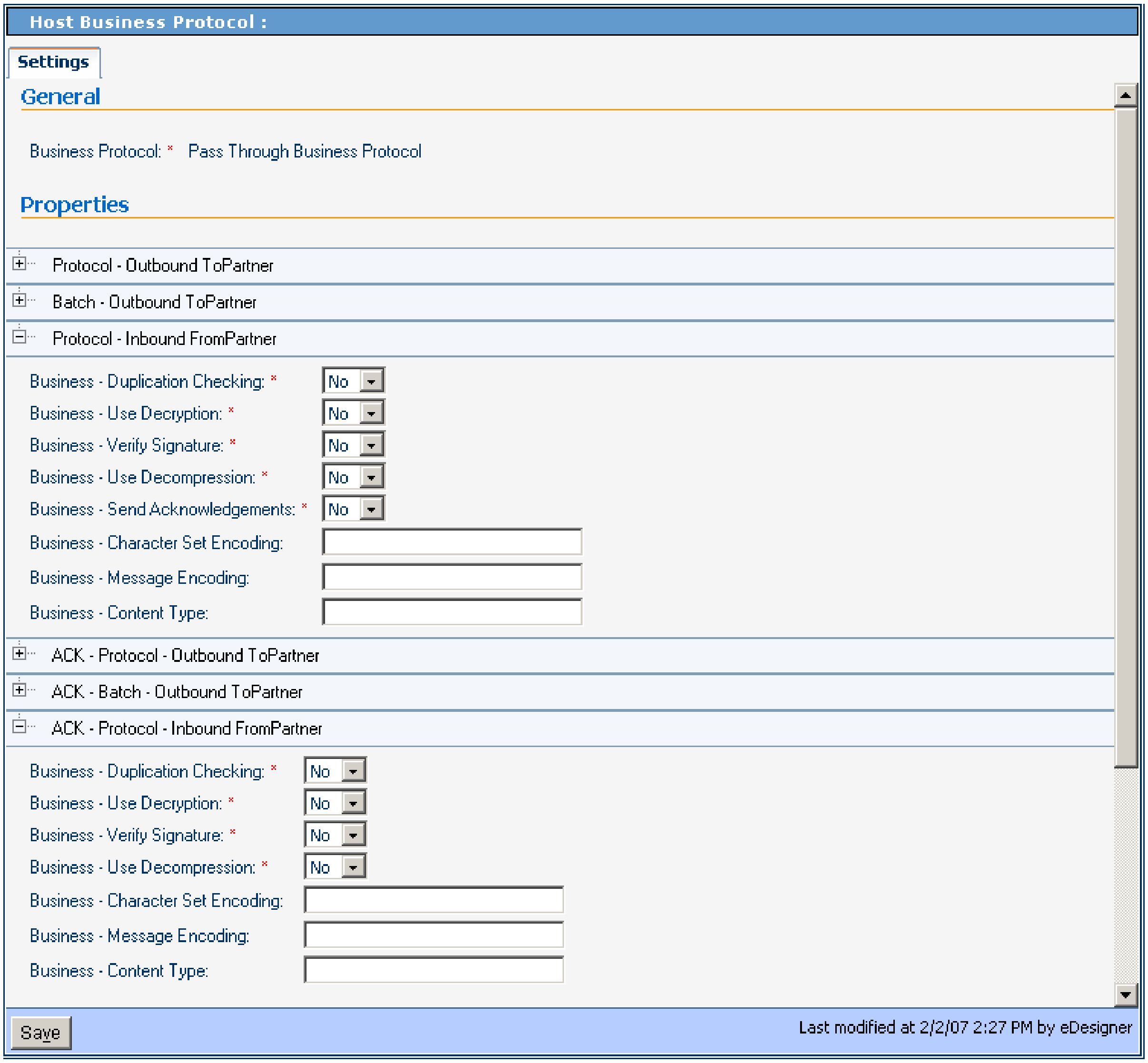
Default

(empty)

6.4.3 Protocol-Inbound (FromPartner) Parameter and ACK Settings

This section lists and describes business protocol-inbound=FromPartner parameter settings and corresponding acknowledgment settings. For an illustration of how these appear in ePM, see Figure 68.

Figure 68 Protocol-Inbound (FromPartner) Parameter and ACK Settings



Host Business Protocol :

Settings

General

Business Protocol: * Pass Through Business Protocol

Properties

Protocol - Outbound ToPartner

Batch - Outbound ToPartner

Protocol - Inbound FromPartner

Business - Duplication Checking: * No
Business - Use Decryption: * No
Business - Verify Signature: * No
Business - Use Decompression: * No
Business - Send Acknowledgements: * No
Business - Character Set Encoding:
Business - Message Encoding:
Business - Content Type:

ACK - Protocol - Outbound ToPartner

ACK - Batch - Outbound ToPartner

ACK - Protocol - Inbound FromPartner

Business - Duplication Checking: * No
Business - Use Decryption: * No
Business - Verify Signature: * No
Business - Use Decompression: * No
Business - Character Set Encoding:
Business - Message Encoding:
Business - Content Type:

Last modified at 2/2/07 2:27 PM by eDesigner

Business - Duplication Checking

Description

Whether or not inbound messages are checked for duplication.

Settings

Yes or **No**

Default

No

Business - Use Decryption

Description

Whether or not inbound messages are decrypted.

Settings

Yes or **No**

Default

No

Business - Verify Signature

Description

Whether or not inbound messages are authenticated for a valid electronic signature.

Settings

Yes or **No**

Default

No

Business - Use Decompression

Description

Whether or not outbound messages are decompressed.

Settings

Yes or **No**

Default

No

Business - Send Acknowledgments

Description

Whether or not inbound messages are to be acknowledged.

Settings

Yes or **No**

Default

No

Business - Character Set Encoding

Description

The character set used to encode the text of the incoming message; optional.

Leave blank to use the default message encoding.

Settings

A string representing a valid character set encoding (such as **ASCII** or **UTF-8**); or empty.

Default

(empty)

Business - Message Encoding

Description

The message encoding by which the native representation of the data can be mapped to a representation that can be used for exchange over 7-bit mail transport protocols; optional.

Leave blank to use the default message encoding.

Settings

A string representing a valid message encoding (such as **binary**, **7bit**, **8bit**, or **base64**); or empty.

Default

(empty)

Business - Content Type

Description

(reserved)

Settings

(reserved)

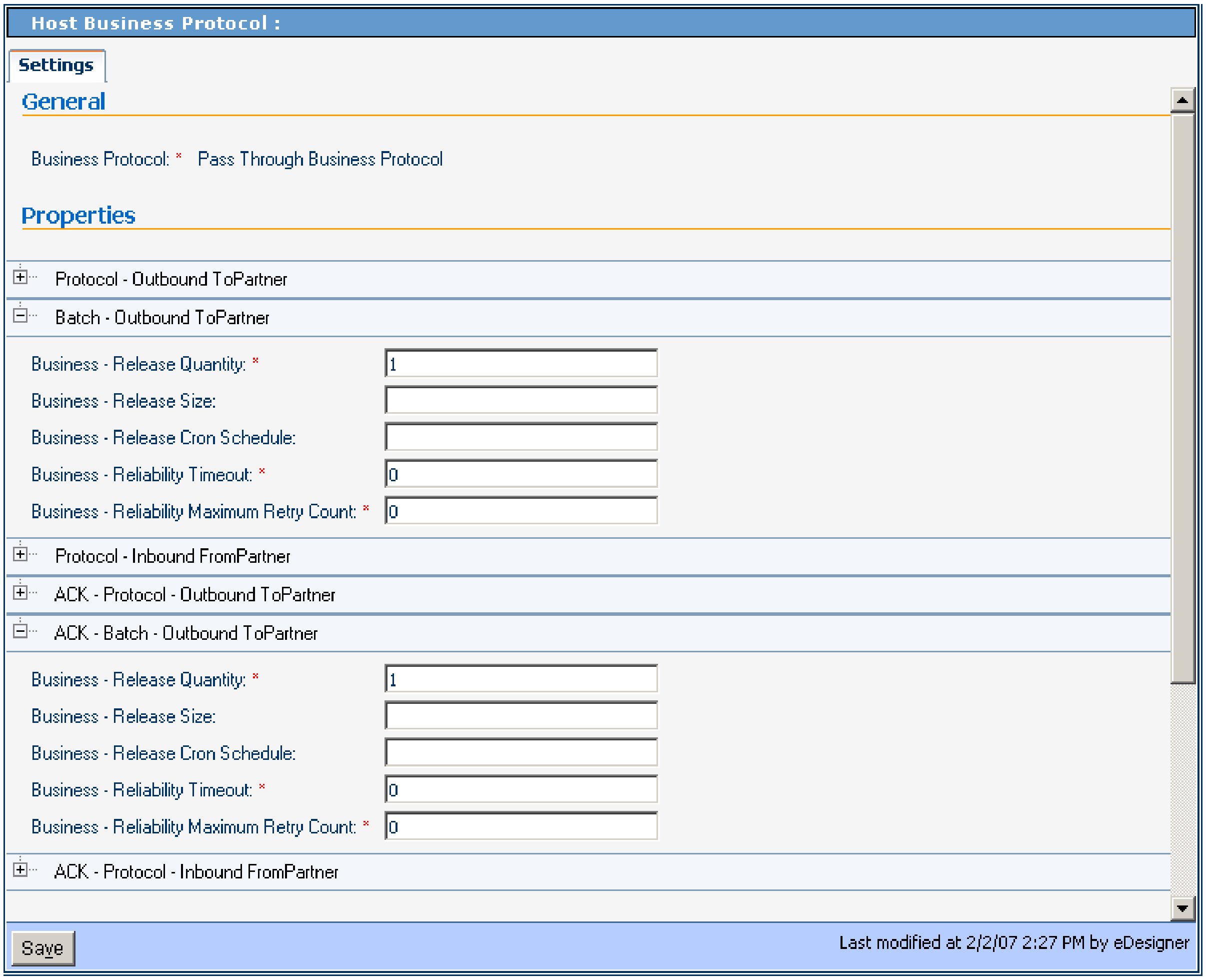
Default

(empty)

6.4.4 Batch-Outbound (ToPartner) Parameter and ACK Settings

This section lists and describes business batch-outbound=ToPartner parameter settings and corresponding acknowledgment settings. For an illustration of how these appear in ePM, see Figure 69.

Figure 69 Batch-Outbound (ToPartner) Parameter and ACK Settings



Business Protocol: * Pass Through Business Protocol

⊞ Protocol - Outbound ToPartner

⊟ Batch - Outbound ToPartner

Business - Release Quantity: *	1
Business - Release Size:	
Business - Release Cron Schedule:	
Business - Reliability Timeout: *	0
Business - Reliability Maximum Retry Count: *	0

⊞ Protocol - Inbound FromPartner

⊞ ACK - Protocol - Outbound ToPartner

⊟ ACK - Batch - Outbound ToPartner

Business - Release Quantity: *	1
Business - Release Size:	
Business - Release Cron Schedule:	
Business - Reliability Timeout: *	0
Business - Reliability Maximum Retry Count: *	0

⊞ ACK - Protocol - Inbound FromPartner

Last modified at 2/2/07 2:27 PM by eDesigner

Business - Release Quantity

Description

Specifies the threshold beyond which a batch send is triggered; required.

Set **Release Quantity** to a very high value (99999999) if you want to send messages on a schedule, using Release Scheduler String.

Settings

Nonnegative integer

Default

1

Business - Release Size

Description

(reserved)

Settings

(reserved)

Default

(empty)

Business - Cron Schedule

Description

A expression specifying when and how often to run; optional.

The expression uses cron syntax and consists of six (or optionally seven) arguments, separated by spaces, to specify: second, minute, hour, day-of-month, month, day-of-week (and optionally year). See **"About cron expressions" on page 115**.

When a trigger time occurs, a batch is sent even if its count has not reached Release Quantity.

Settings

Either a valid cron expression (see below), or empty.

Default

(empty)

About cron expressions

A cron expression specifies when and how often to run. It consists of six (or optionally seven) arguments, separated by spaces:

- Arg#1: **Second.** An integer in the range 0-59.
- Arg#2: **Minute.** An integer in the range 0-59.
- Arg#3: **Hour.** An integer in the range 0-23.
- Arg#4: **Day-of-month.** An integer in the range 1-31.
- Arg#5: **Month.** Either an integer in the range 1-12, or a case-insensitive three-character string, from: {Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec}.
- Arg#6: **Day-of-week.** Either an integer in the range 1-7, or a case-insensitive three-character string, from: {Sun, Mon, Tue, Wed, Thu, Fri, Sat}.
- Arg#7 (optional): **Year.** Either empty, or an integer in the range 1970-2099.

Be careful not to supply real numbers (such as 0.5) if an integer is required; doing so causes misparsing and misprocessing to occur, but does not issue an error. Although the "." (period) character is not permitted, the following special characters are allowed:

- ***** (asterisk): Can be used in any of the seven fields to signify "all values". For example, "*" as the fourth argument means "every day."
- **,** (comma): Can be used in any of the seven fields to separate items in a discrete list. For example, "MON,THU" in the sixth field means "on Monday and Thursday".
- **-** (hyphen): Can be used in any of the seven fields to indicate minimum-maximum of a range. For example, "Nov-Feb" in the fifth field means "in November, December, January, and February".
- **/** (slash): Can be used in any of the seven fields to specify increments. For example, "7/20" in the first field means "on seconds 7, 27, and 47".
- **?** (query): Can be used in field 4 or field 6 (but no other field, and not simultaneously) to avoid collision or ambiguity.
- **L** can be used in field 4 or field 6 or both (but not in combination with lists or ranges) to specify "the last". For example, "L" in the fourth field means "the last day of the month" and "1L" in the sixth field means "the last Sunday of the month".

Examples of valid cron expressions

- "**0 0 12 * * ?"** means "At noon every day".
- "**0 30 21 ? * ***" means "At 9:30pm every day".
- "**0 0 2 ? * Mon-Fri**" means "Weekdays at 2:00am".
- "**0 10/30 1,22 L * ?**" means "At 1:10 and 1:40am and 10:10 and 10:40pm the last day of every month".

Business - Reliability Timeout

Description

Specifies the maximum number of minutes to wait for a reply before attempting a re-send; required.

Settings

Nonnegative integer

Default

0

Business - Reliability Maximum Retry Count

Description

Specifies maximum number of times to retry sending before giving up; required.

Settings

Nonnegative integer. The special setting **0** (the default) carries the following meaning:

(1) "I do not re-send messages to this partner."

(2)"If I expect a TA1 but do not receive one, I shall nonetheless issue no error."

Default

0

6.5 Viewing and Setting Delivery Protocol Parameters

Each B2B Host in ePM is automatically presupplied with the *PassThrough* delivery protocol. The PassThrough protocol provides no parameter settings beyond the minimum required to deliver and receive message packets without operating on them.

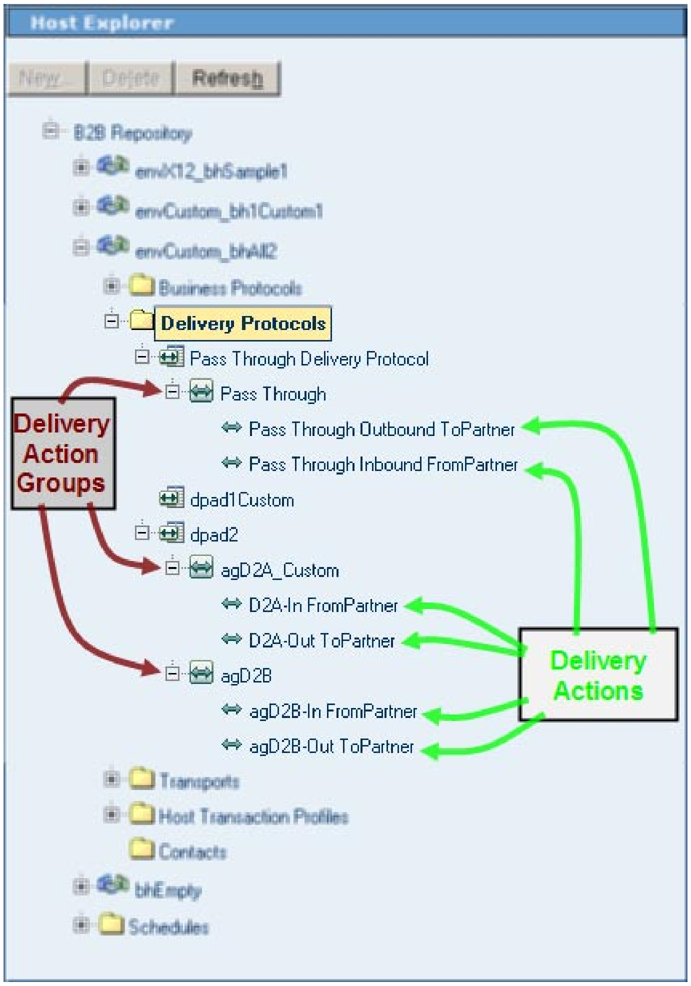
In addition to the Passthrough protocol, each B2B Host may contain one or more other delivery protocols, whose parameter types and possible settings depend on the way in which their delivery protocol attribute definitions (DPADs) were set up in Enterprise Designer:

- Parameters for each Protocol Manager product are described in the corresponding Protocol Manager User's Guide.
- Parameters for custom delivery protocols are determined by the custom DPADs defined in Enterprise Manager. For complete information on the relationship between the DPAD settings and ePM parameters, refer to **Appendix B on page 188**.
- Other parameters and their possible settings are determined the delivery protocol's messaging attribute definitions (MADs) and packaging attribute definitions (PADs). See **"Parameter and ACK Settings for Delivery Protocols" on page 120**.

6.5.1 Configuring Delivery Action Groups and Delivery Actions

A *delivery action* is a particular messaging format specification, such as the type of encryption used for message payload and signature. A *delivery action group* is a collection of delivery actions that logically belong in the same group. See Figure 70.

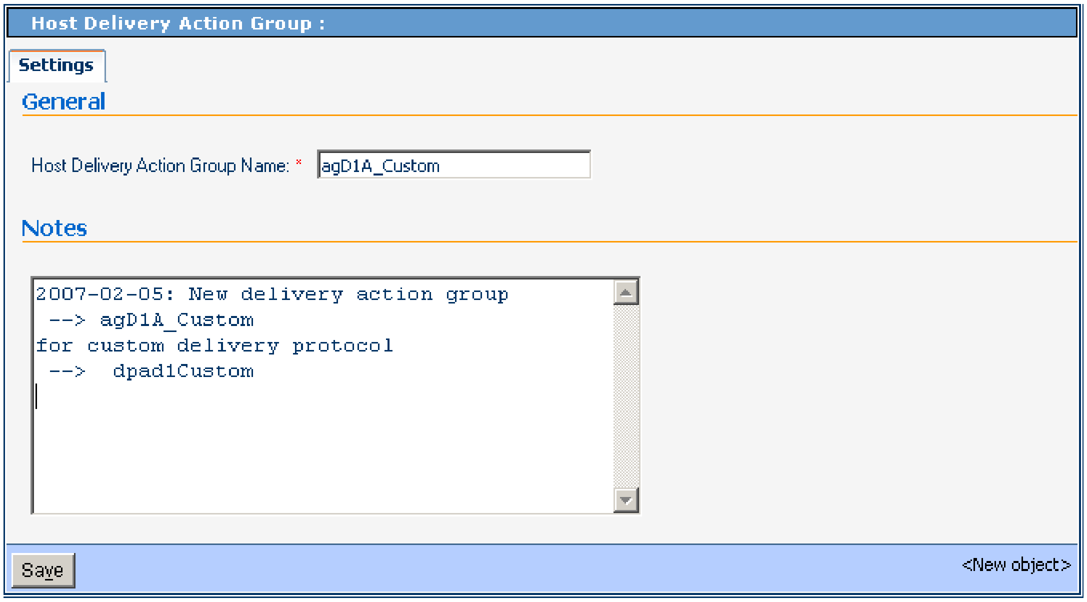
Figure 70 Delivery Action Groups and Delivery Actions



To create and configure a delivery action group

1. In the Host Explorer tree, click the delivery protocol under which you want to create an action group. This could be the **Pass Through Delivery Protocol**, a business protocol supplied with a Protocol Manager, or a custom delivery protocol, depending on the attribute definitions you want to use.
2. Click **New**.
3. In the Host Delivery Action Group pane on the right, enter a name for the action group. See Figure 71.

Figure 71 Creating a New Delivery Action Group



Host Delivery Action Group Name: * agD1A_Custom

```
2007-02-05: New delivery action group
 --> agD1A_Custom
for custom delivery protocol
 -->  dpad1Custom
```

4 Click **Save**.

Result: The new delivery action group appears in the explorer tree on the left and, in the right pane, a confirmation message is displayed in a new tab: **Overrides**.

5 As required by the nature of the delivery formats, modify parameter settings as needed by overriding the default values shown. Any modifications you make will be inherited by delivery actions and transaction profiles that use this action group.

6 Click **Save**.

7 Repeat as needed to create additional action groups for the same delivery protocol.

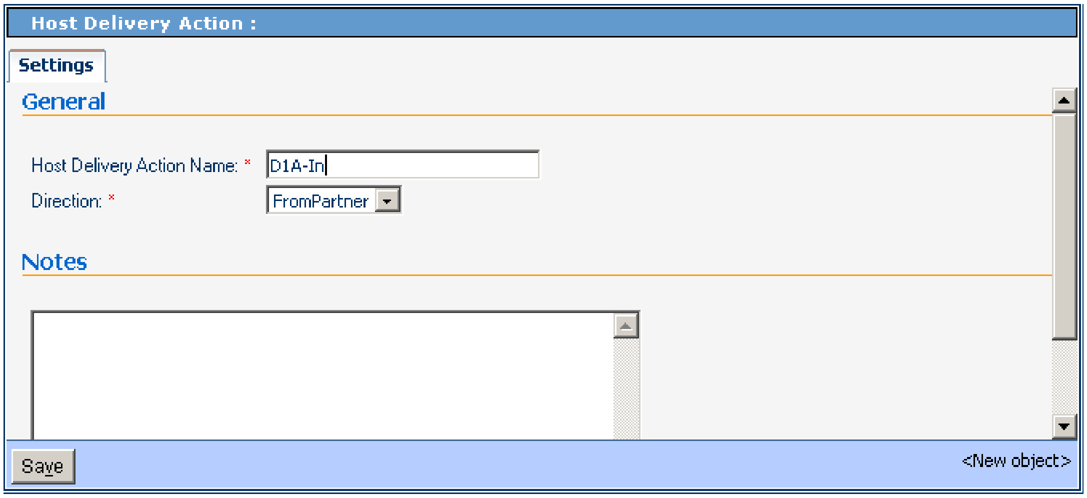
To create and configure a delivery action

1 In the Host Explorer tree, click the delivery action group under which you want to create a delivery action group.

2 Click **New**.

3 In the Host Delivery Action Group pane on the right, enter a name for the action group and, for Direction, choose either FromPartner or ToPartner. See Figure 72.

Figure 72 Creating a New Delivery Action



Host Delivery Action Name: * D1A-In

Direction: * FromPartner

4 Click **Save**.

Result: The new delivery action appears in the explorer tree on the left and, in the right pane, a confirmation message is displayed in a new tab: **Overrides**.

5 As required by the nature of the delivery format, modify parameter settings as needed by overriding the default values shown. Any modifications that you make will be inherited by transaction profiles that use this delivery action.

6 Click **Save**.

7 Repeat as needed to create additional delivery actions in the same action group.

6.5.2 Parameter and ACK Settings for Delivery Protocols

With only a few exceptions, the parameter settings and acknowledgment settings for delivery protocols are the same as for business protocols. For details, see:

- **"Protocol-Outbound (ToPartner) Parameter and ACK Settings" on page 107**.
- **"Protocol-Inbound (FromPartner) Parameter and ACK Settings" on page 111**.
- **"Batch-Outbound (ToPartner) Parameter and ACK Settings" on page 114** and **"Delivery - Batch Tracking Code"**.

Delivery - Batch Tracking Code

Description

List of options for tracking message batches. To assure tracking of outbound and inbound batches, retain the default setting

Settings

No Tracking or **Header Only** or **Header and Message**

Default

Header and Message

6.6 Creating and Configuring Transaction Profiles

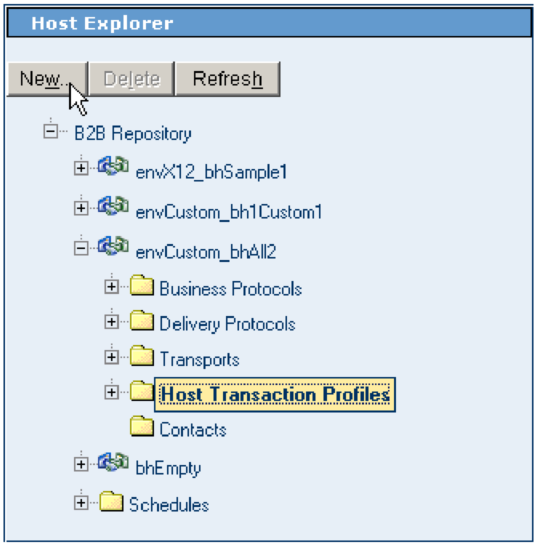
A Transaction Profile consists of well-configured outbound and inbound business actions, associated with well-configured outbound and inbound delivery actions, such that each delivery action is tied to a well-configured transport. Actions from the PassThrough action group can be used for either or both business actions and/or either or both delivery actions.

Transaction profiles are created in groups containing individual outbound (to-partner) and inbound (from-partner) profiles.

To create a new transaction profile group

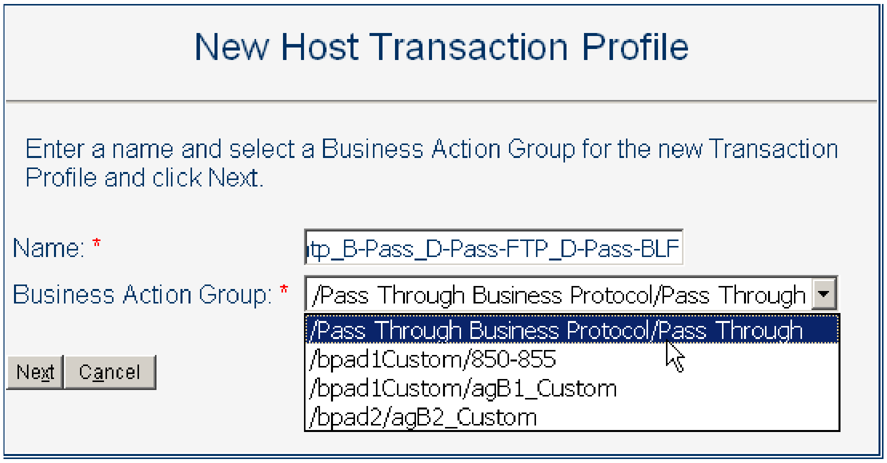
1 In the Host Explorer tree, open the Host for the transaction profile, click the **Host Transaction Profiles** folder and click **New**. See Figure 73.

Figure 73 Creating a New Transaction Profile Group



2 In the **New Host Business Transaction Profile** pane on the right, enter a name for the transaction profile group and select a business action group from the pull-down list. See Figure 65.

Figure 74 New Transaction Profile Group - Name and Business Action Group



New Host Transaction Profile

Enter a name and select a Business Action Group for the new Transaction Profile and click Next.

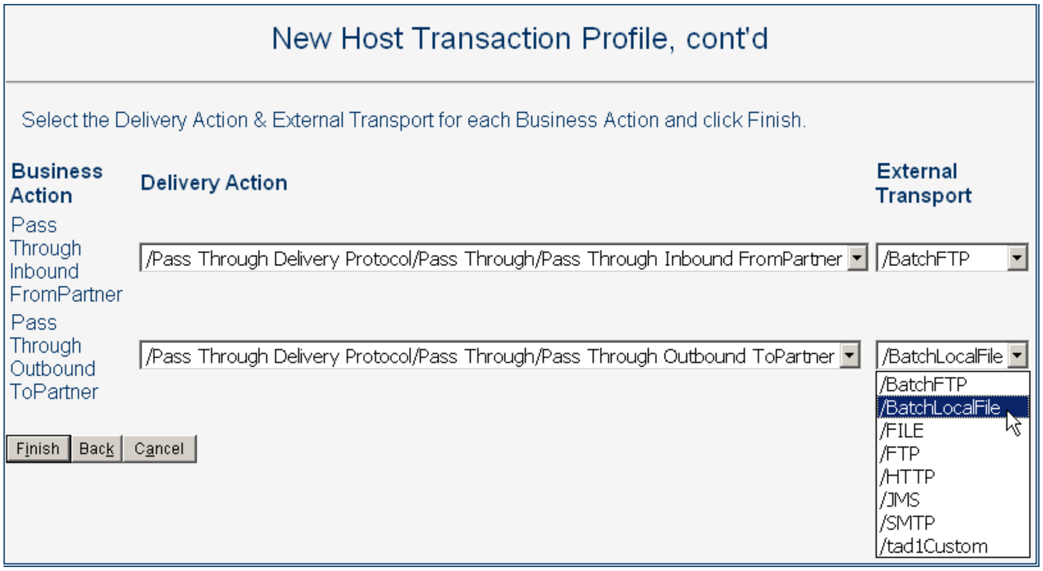
Name: * tp_B-Pass_D-Pass-FTP_D-Pass-BLF

Business Action Group: * /Pass Through Business Protocol/Pass Through

3 Click **Next**.

4 For the inbound business action, and then again for the outbound business action, select a corresponding inbound delivery action and external transport from the pull-down lists. See Figure 75.

Figure 75 New Transaction Profile Group - Delivery Actions and Transports



New Host Transaction Profile, cont'd

Select the Delivery Action & External Transport for each Business Action and click Finish.

Business Action	Delivery Action	External Transport
Pass Through Inbound FromPartner	/Pass Through Delivery Protocol/Pass Through/Pass Through Inbound FromPartner	/BatchFTP
Pass Through Outbound ToPartner	/Pass Through Delivery Protocol/Pass Through/Pass Through Outbound ToPartner	/BatchLocalFile

5 Optionally, you can click **Back** to change any choices previously made.

6 When you are satisfied with your choices, click **Finish**.

Result: After a pause, the new transaction profile group appears in the explorer tree.

7 Click **Save**.

8 Optionally, you can open the group to see the inbound and outbound transaction profiles that were created within the group, and you can use the **Overrides** tab to

modify default values of any settings. Any modifications made to a Host transaction profile are inherited by all trading partner transaction profiles that reference it.

9 Repeat as needed to create additional transaction profile groups for the same host.

6.7 Inheriting and Overriding Parameter Settings

Parameter settings can be overridden at several levels both for the B2B Host and, below the level of B2B Host, for each individual trading partner.

At all levels below the Host Protocol/Transport, (including all trading partners levels):

- A parameter can have its value set explicitly. If it has an explicit setting at this level, the setting overrides any default values that were set for this parameter higher in the inheritance chain.
- A parameter can be set to inherit from the next-higher level. If so, the parameter's current value at any time is inherited from the first explicit setting higher in its inheritance chain.

To explicitly override a parameter setting at a particular level, access the **Overrides** tab for that level, locate the parameter, check its **Override** box, explicitly set the parameter to the value you want, and click **Save**. When you do this, all of the same parameters lower in the inheritance chain (except for those affected by an explicit setting lower in the chain) are updated to the new value.

Quick Summary of Inheritance

Inheritance of ePM parameter settings follows these rules ("<" means "inherits from"):

- Host transaction profile < host action < host action group < host protocol
(< Enterprise Designer attribute definitions)
- TP transaction profile < TP action < TP action group < TP protocol;
but if not explicitly overridden anywhere in the TP chain: < host transaction profile
- TP action < TP action group < TP protocol;
but if not explicitly overridden anywhere in the TP chain: < host action
- TP action group < TP protocol;
but if not explicitly overridden anywhere in the TP chain: < host action group
- TP protocol < host protocol

6.7.1 Inheritance and Overrides in B2B Host Configuration

On the Host side, inheritance and overrides are involved but straightforward:

- If a **host transaction profile** parameter setting is not explicitly overridden, then:
settings for its business parameters are inherited from the *host business action*;

settings for its delivery parameters are inherited from the *host delivery action*; and settings for its transport parameters are inherited directly from the *host transport*.

- If a **host** business or delivery **action** setting is not explicitly overridden, then settings for its parameters are inherited from its *host action group*.
- If a **host** business or delivery **action group** setting is not explicitly overridden, then settings for its parameters are inherited from its *host business or delivery protocol*.
- If a **host** transport/business/delivery **protocol** setting is not explicitly overridden, then settings for its parameters are determined by the *attribute definitions* built into B2B Host in Enterprise Designer.
- Host transport settings inherited from the B2B Host built in Enterprise Designer are also affected by eWay configuration settings. If an eWay configuration parameter is not set, its value is determined by the configuration settings for the corresponding external system in the Environment.

6.7.2 Inheritance and Overrides in Trading Partner Configuration

On the TP side, inheritance and overrides can be affected by Host settings:

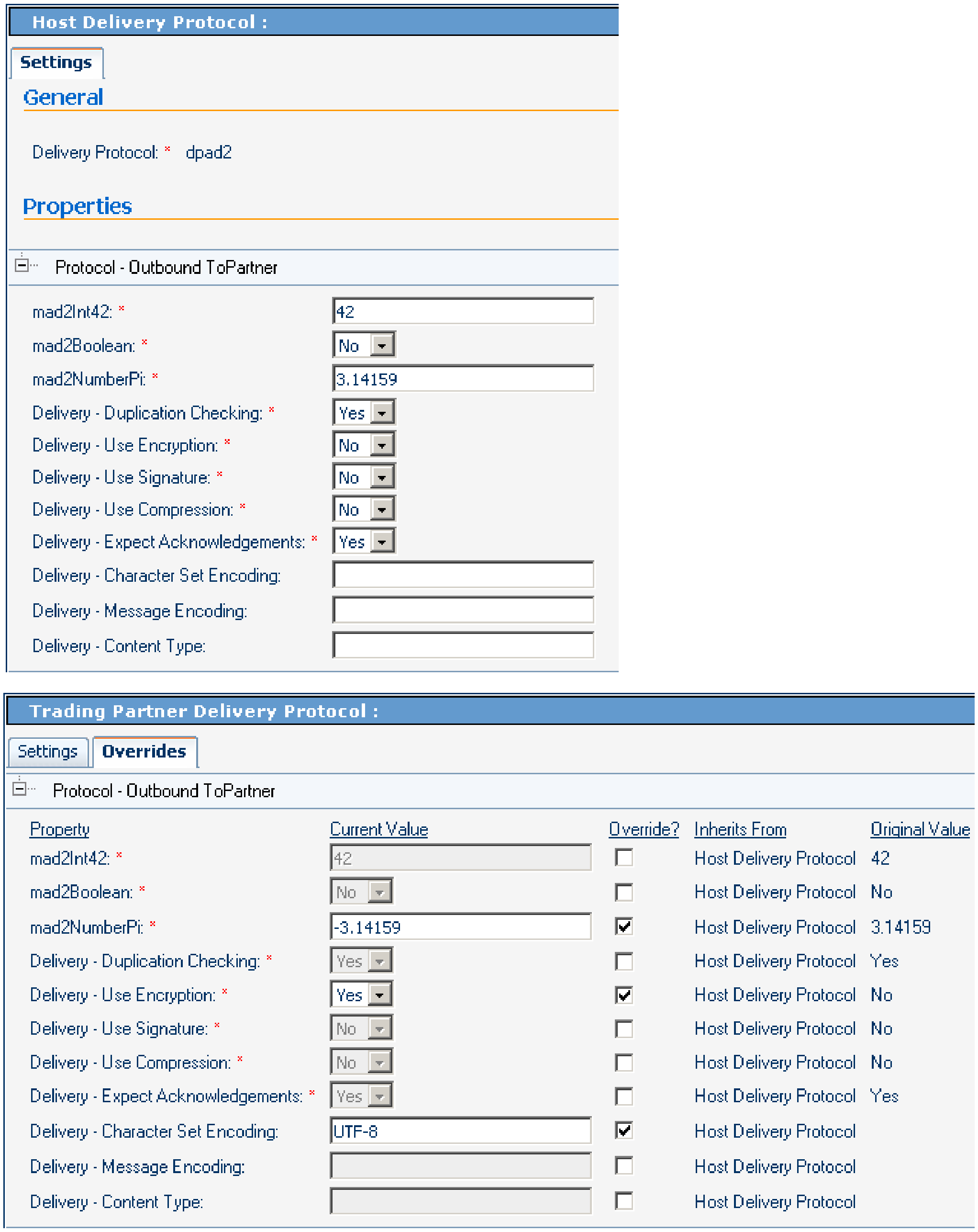
- If a **TP transaction profile** parameter setting is not explicitly overridden, then: settings for its business parameters are inherited from the *TP business action*; settings for its delivery parameters are inherited from the *TP delivery action*; and settings for its transport parameters are inherited directly from the *TP transport*; however, if there are no explicit overrides in the TP chain, it inherits from the corresponding *host transaction profile*.
- If a **TP** business or delivery **action** setting is not explicitly overridden, then it inherits any explicit overrides in its *TP action group* and above; however, if there are no explicit overrides in the TP chain, it inherits from the corresponding *host action*.
- If a **TP** business or delivery **action group** setting is not explicitly overridden, then it inherits any explicit overrides in its *TP business/delivery protocol*; however, if there are no explicit overrides in the TP protocol, it inherits from the corresponding *host action group*.
- If a **TP** transport or business/delivery **protocol** setting is not explicitly overridden, then it inherits from any explicit overrides in the corresponding *host protocol*.

6.7.3 Example of Multiple Inheritance and Overrides

In this example, a B2B Host was created that included a custom delivery protocol named dpad2 that included three custom messaging attribute definitions (MADs). In ePM, various of the outbound parameter settings were modified at every possible level. (This is poor practice, and employed here simply to illustrate the principles.)

Figure 76 below shows parameters at the highest level in ePM—the protocol level itself. In the example, three parameters in the host protocol are overridden by the TP protocol.

Figure 76 Parameters at the Protocol Level: Host and TP



Host Delivery Protocol :

General

Delivery Protocol: * dpad2

Properties

Protocol - Outbound ToPartner

mad2Int42: *	42
mad2Boolean: *	No
mad2NumberPi: *	3.14159
Delivery - Duplication Checking: *	Yes
Delivery - Use Encryption: *	No
Delivery - Use Signature: *	No
Delivery - Use Compression: *	No
Delivery - Expect Acknowledgements: *	Yes
Delivery - Character Set Encoding:	
Delivery - Message Encoding:	
Delivery - Content Type:	

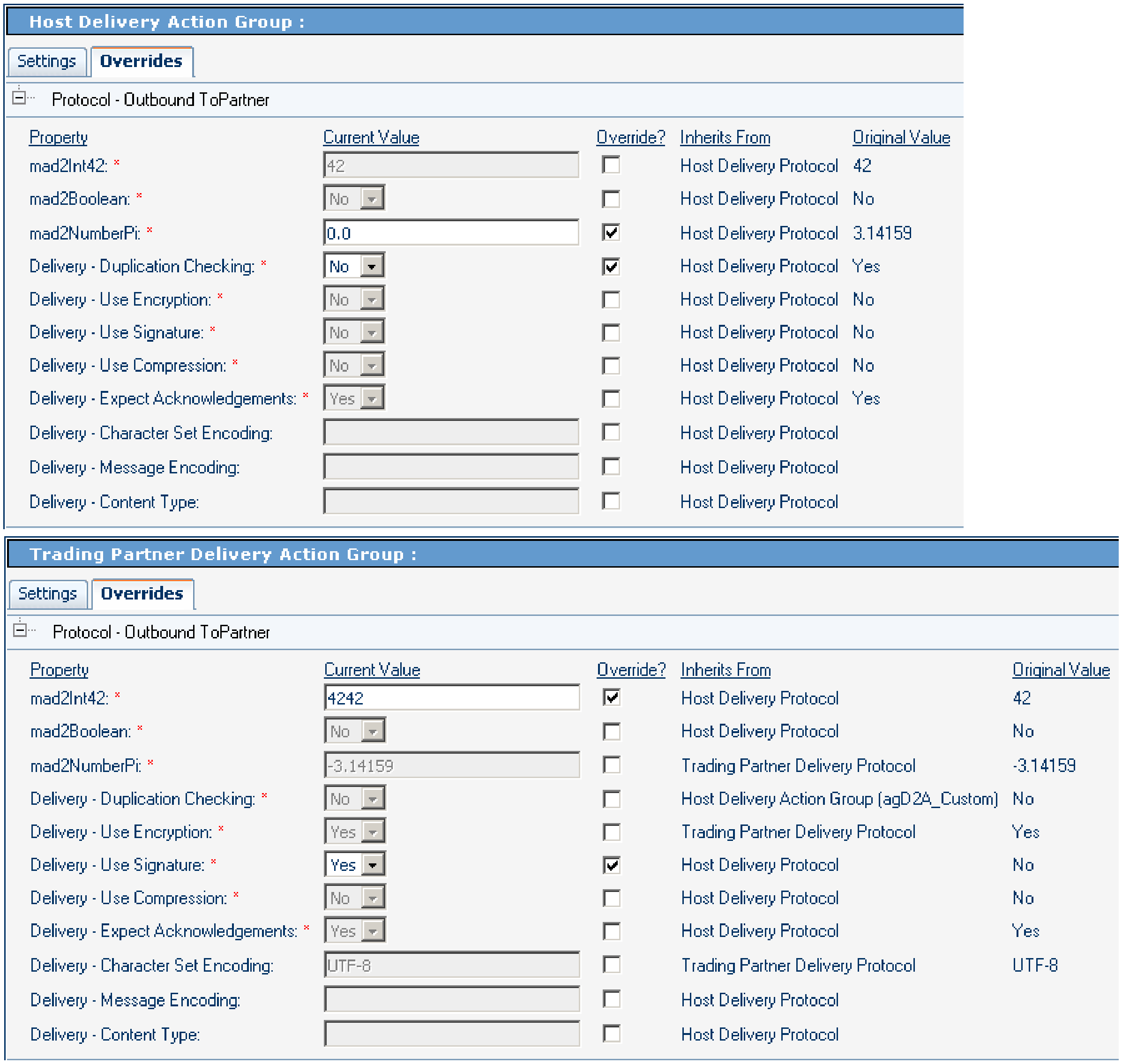
Trading Partner Delivery Protocol :

Protocol - Outbound ToPartner

Property	Current Value	Override?	Inherits From	Original Value
mad2Int42: *	42	☐	Host Delivery Protocol	42
mad2Boolean: *	No	☐	Host Delivery Protocol	No
mad2NumberPi: *	-3.14159	☑	Host Delivery Protocol	3.14159
Delivery - Duplication Checking: *	Yes	☐	Host Delivery Protocol	Yes
Delivery - Use Encryption: *	Yes	☑	Host Delivery Protocol	No
Delivery - Use Signature: *	No	☐	Host Delivery Protocol	No
Delivery - Use Compression: *	No	☐	Host Delivery Protocol	No
Delivery - Expect Acknowledgements: *	Yes	☐	Host Delivery Protocol	Yes
Delivery - Character Set Encoding:	UTF-8	☑	Host Delivery Protocol	
Delivery - Message Encoding:		☐	Host Delivery Protocol	
Delivery - Content Type:		☐	Host Delivery Protocol	

Figure 77 below shows parameters at the second-highest ePM level—the action group level. In the example, the host action group overrides two parameter settings inherited from the host protocol, and the TP action group overrides two different parameter settings inherited from the host action group.

Figure 77 Parameters at the Action Group Level: Host and TP



Host Delivery Action Group :

⊟ Protocol - Outbound ToPartner

Property	Current Value	Override?	Inherits From	Original Value
mad2Int42: *	42	☐	Host Delivery Protocol	42
mad2Boolean: *	No	☐	Host Delivery Protocol	No
mad2NumberPi: *	0.0	☑	Host Delivery Protocol	3.14159
Delivery - Duplication Checking: *	No	☑	Host Delivery Protocol	Yes
Delivery - Use Encryption: *	No	☐	Host Delivery Protocol	No
Delivery - Use Signature: *	No	☐	Host Delivery Protocol	No
Delivery - Use Compression: *	No	☐	Host Delivery Protocol	No
Delivery - Expect Acknowledgements: *	Yes	☐	Host Delivery Protocol	Yes
Delivery - Character Set Encoding:		☐	Host Delivery Protocol	
Delivery - Message Encoding:		☐	Host Delivery Protocol	
Delivery - Content Type:		☐	Host Delivery Protocol	

Trading Partner Delivery Action Group :

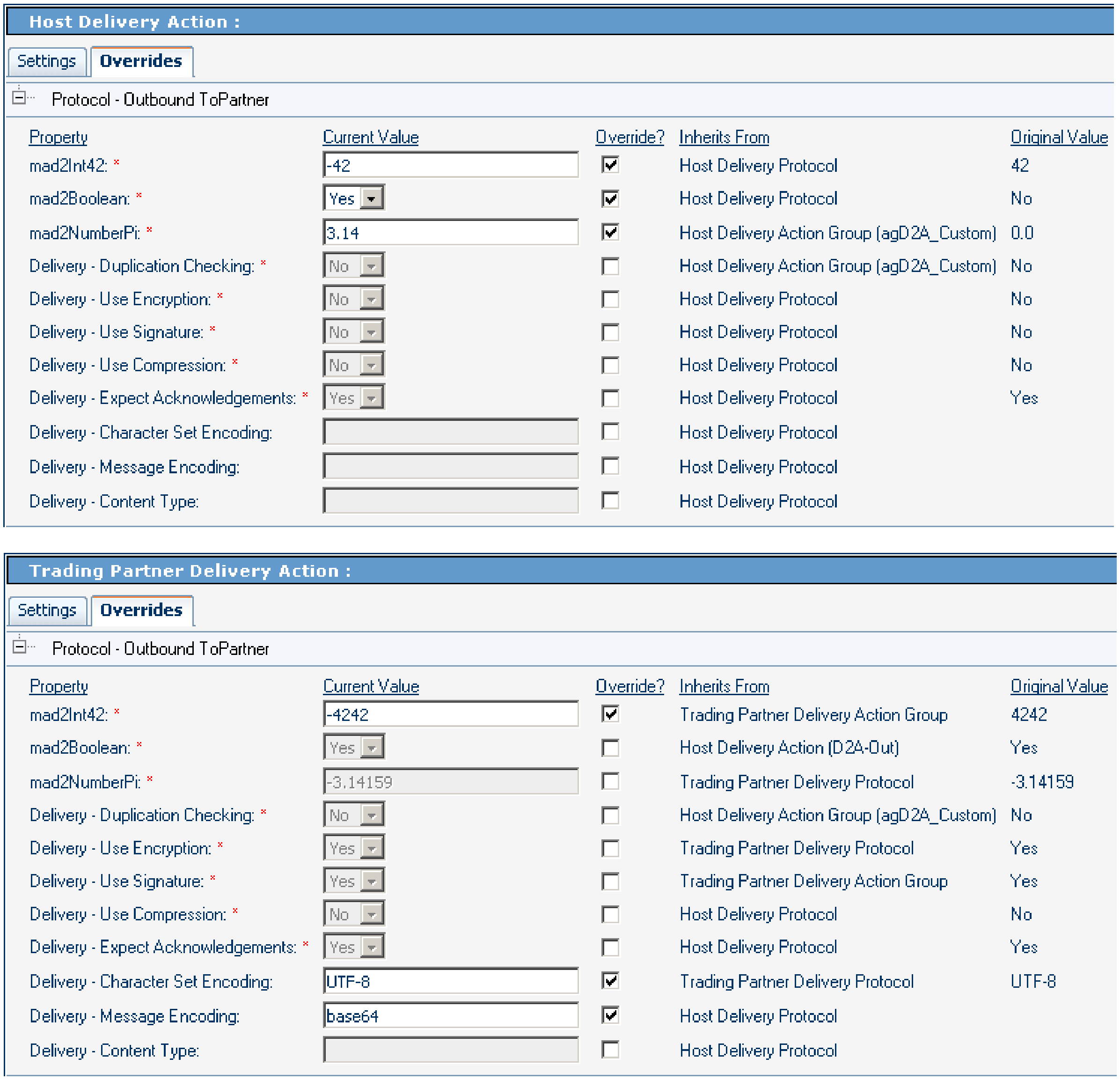
⊟ Protocol - Outbound ToPartner

Property	Current Value	Override?	Inherits From	Original Value
mad2Int42: *	4242	☑	Host Delivery Protocol	42
mad2Boolean: *	No	☐	Host Delivery Protocol	No
mad2NumberPi: *	-3.14159	☐	Trading Partner Delivery Protocol	-3.14159
Delivery - Duplication Checking: *	No	☐	Host Delivery Action Group (agD2A_Custom)	No
Delivery - Use Encryption: *	Yes	☐	Trading Partner Delivery Protocol	Yes
Delivery - Use Signature: *	Yes	☑	Host Delivery Protocol	No
Delivery - Use Compression: *	No	☐	Host Delivery Protocol	No
Delivery - Expect Acknowledgements: *	Yes	☐	Host Delivery Protocol	Yes
Delivery - Character Set Encoding:	UTF-8	☐	Trading Partner Delivery Protocol	UTF-8
Delivery - Message Encoding:		☐	Host Delivery Protocol	
Delivery - Content Type:		☐	Host Delivery Protocol	

In Figure 77 above, notice that the TP action group setting for Duplication Checking is inherited from its parallel host action group (because there is no explicit override at TP protocol level), and that the settings for mad2NumberPi, Encryption, and Character Set Encoding are inherited from its immediate parent, the TP protocol level.

Figure 78 shows parameters at the third-highest level in ePM—the action level. In the example, the host action overrides two parameter settings inherited from the protocol and one inherited from the action group. The TP action overrides one parameter setting inherited from the TP action group, one inherited from the TP protocol, and one inherited from the host protocol.

Figure 78 Parameters at the Action Level: Host and TP



Host Delivery Action :

Protocol - Outbound ToPartner

Property	Current Value	Override?	Inherits From	Original Value
mad2Int42: *	-42	☑	Host Delivery Protocol	42
mad2Boolean: *	Yes	☑	Host Delivery Protocol	No
mad2NumberPi: *	3.14	☑	Host Delivery Action Group (agD2A_Custom)	0.0
Delivery - Duplication Checking: *	No	□	Host Delivery Action Group (agD2A_Custom)	No
Delivery - Use Encryption: *	No	□	Host Delivery Protocol	No
Delivery - Use Signature: *	No	□	Host Delivery Protocol	No
Delivery - Use Compression: *	No	□	Host Delivery Protocol	No
Delivery - Expect Acknowledgements: *	Yes	□	Host Delivery Protocol	Yes
Delivery - Character Set Encoding:		□	Host Delivery Protocol	
Delivery - Message Encoding:		□	Host Delivery Protocol	
Delivery - Content Type:		□	Host Delivery Protocol	

Trading Partner Delivery Action :

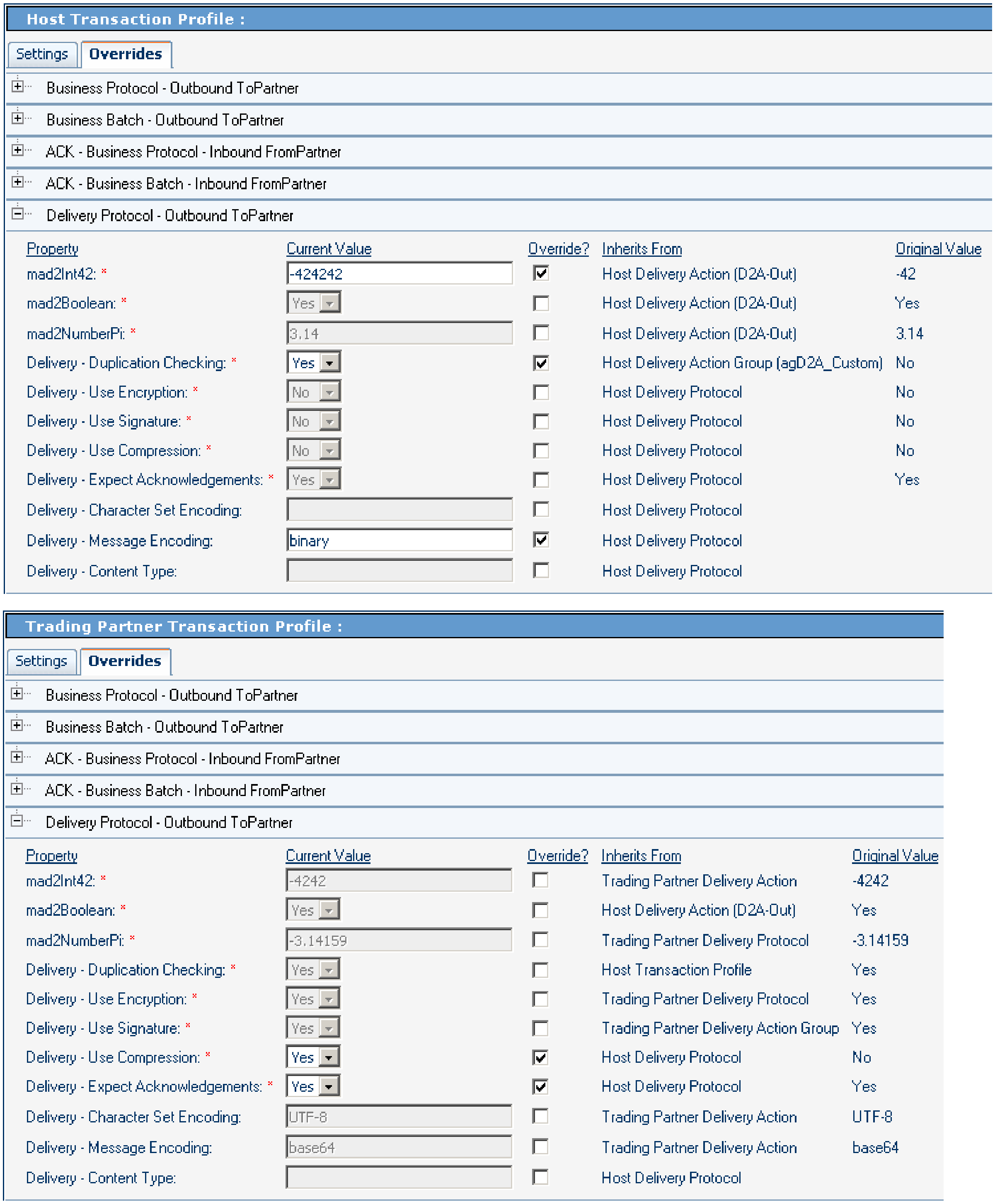
Protocol - Outbound ToPartner

Property	Current Value	Override?	Inherits From	Original Value
mad2Int42: *	-4242	☑	Trading Partner Delivery Action Group	4242
mad2Boolean: *	Yes	□	Host Delivery Action (D2A-Out)	Yes
mad2NumberPi: *	-3.14159	□	Trading Partner Delivery Protocol	-3.14159
Delivery - Duplication Checking: *	No	□	Host Delivery Action Group (agD2A_Custom)	No
Delivery - Use Encryption: *	Yes	□	Trading Partner Delivery Protocol	Yes
Delivery - Use Signature: *	Yes	□	Trading Partner Delivery Action Group	Yes
Delivery - Use Compression: *	No	□	Host Delivery Protocol	No
Delivery - Expect Acknowledgements: *	Yes	□	Host Delivery Protocol	Yes
Delivery - Character Set Encoding:	UTF-8	☑	Trading Partner Delivery Protocol	UTF-8
Delivery - Message Encoding:	base64	☑	Host Delivery Protocol	
Delivery - Content Type:		□	Host Delivery Protocol	

In Figure 78 above, notice that the TP action setting for mad2Boolean is inherited from its parallel host action (because there is no explicit override in the TP inheritance chain, either at the TP protocol or the TP action group level), and that the setting for Use Signature is inherited from its immediate parent, the TP action group level.

Figure 79 shows parameters at the bottom level in ePM—the transaction profile level. In the example, the host transaction profile overrides one parameter settings inherited from the protocol, one inherited from the action group, and one inherited from the action. The TP transaction profile overrides two parameter settings, both inherited from the host protocol, but one of them deserves special attention: The parameter setting for Expect Acknowledgments, which was set to Yes in the host protocol (see Figure 76), is overridden and set to Yes—the same value. The effect of this is to break the inheritance but keep the value: Even if the setting in the host protocol should change in the future, the parameter setting in the TP transaction profile will not change.

Figure 79 Parameters at the Transaction Profile Level: Host and TP



Host Transaction Profile :

⊞ Business Protocol - Outbound ToPartner
⊞ Business Batch - Outbound ToPartner
⊞ ACK - Business Protocol - Inbound FromPartner
⊞ ACK - Business Batch - Inbound FromPartner
⊟ Delivery Protocol - Outbound ToPartner

Property	Current Value	Override?	Inherits From	Original Value
mad2Int42: *	-424242	☑	Host Delivery Action (D2A-Out)	-42
mad2Boolean: *	Yes	☐	Host Delivery Action (D2A-Out)	Yes
mad2NumberPi: *	3.14	☐	Host Delivery Action (D2A-Out)	3.14
Delivery - Duplication Checking: *	Yes	☑	Host Delivery Action Group (agD2A_Custom)	No
Delivery - Use Encryption: *	No	☐	Host Delivery Protocol	No
Delivery - Use Signature: *	No	☐	Host Delivery Protocol	No
Delivery - Use Compression: *	No	☐	Host Delivery Protocol	No
Delivery - Expect Acknowledgements: *	Yes	☐	Host Delivery Protocol	Yes
Delivery - Character Set Encoding:		☐	Host Delivery Protocol	
Delivery - Message Encoding:	binary	☑	Host Delivery Protocol	
Delivery - Content Type:		☐	Host Delivery Protocol	

Trading Partner Transaction Profile :

⊞ Business Protocol - Outbound ToPartner
⊞ Business Batch - Outbound ToPartner
⊞ ACK - Business Protocol - Inbound FromPartner
⊞ ACK - Business Batch - Inbound FromPartner
⊟ Delivery Protocol - Outbound ToPartner

Property	Current Value	Override?	Inherits From	Original Value
mad2Int42: *	-4242	☐	Trading Partner Delivery Action	-4242
mad2Boolean: *	Yes	☐	Host Delivery Action (D2A-Out)	Yes
mad2NumberPi: *	-3.14159	☐	Trading Partner Delivery Protocol	-3.14159
Delivery - Duplication Checking: *	Yes	☐	Host Transaction Profile	Yes
Delivery - Use Encryption: *	Yes	☐	Trading Partner Delivery Protocol	Yes
Delivery - Use Signature: *	Yes	☐	Trading Partner Delivery Action Group	Yes
Delivery - Use Compression: *	Yes	☑	Host Delivery Protocol	No
Delivery - Expect Acknowledgements: *	Yes	☑	Host Delivery Protocol	Yes
Delivery - Character Set Encoding:	UTF-8	☐	Trading Partner Delivery Action	UTF-8
Delivery - Message Encoding:	base64	☐	Trading Partner Delivery Action	base64
Delivery - Content Type:		☐	Host Delivery Protocol	

]

6.8 Creating and Configuring Trading Partners

Ultimately, the parameters settings that are actually used when business messages are delivered to/from trading partners are determined by a trading partner transaction profile.

- For outgoing messages, the choice of transaction profile is determined by the name of the trading partner, action group, and action.
- For incoming messages, the choice of transaction profile is determined by the endpoint where the message is picked up.

A particular trading partner can contain many transaction profiles based on several different B2B Hosts, but each TP transaction profile is based on exactly one host transaction profile. However, individual parameters settings in the host transaction profile can be overridden—either within the TP transaction itself, or anywhere in the TP inheritance chain: TP action, TP action group, and TP business/delivery protocol. For details, see **Inheriting and Overriding Parameter Settings** on page 123.

6.8.1 Basic Operations for Trading Partners

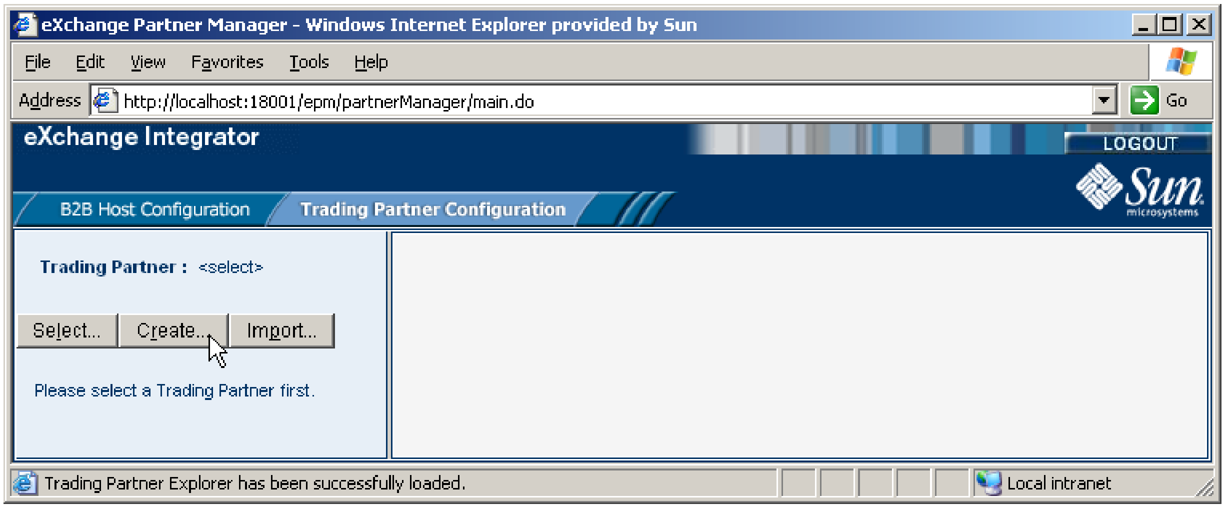
This section presents the following procedures:

- **To create a new trading partner** on page 130
- **To find and select an existing trading partner** on page 131
- **To add a B2B Host to a trading partner** on page 132

To create a new trading partner

1. With the **Trading Partner Configuration** tab active, in the explorer pane, click **Create**. See Figure 80.

Figure 80 Creating a New Trading Partner



eXchange Partner Manager - Windows Internet Explorer provided by Sun

Address http://localhost:18001/epm/partnerManager/main.do

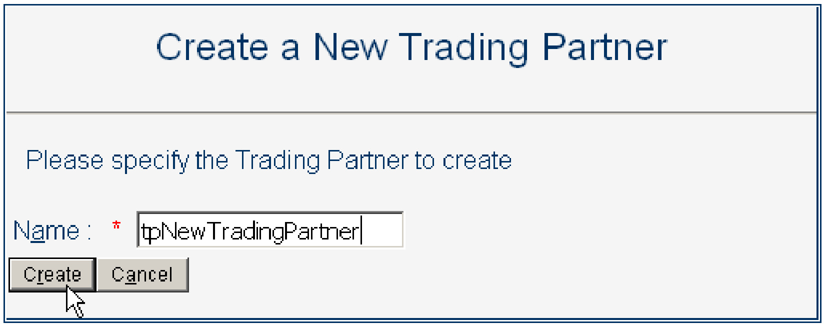
Trading Partner : <select>

Please select a Trading Partner first.

Trading Partner Explorer has been successfully loaded.

2 In the canvas (Create a New Trading Partner), enter a name for the new trading partner. See Figure 81.

Figure 81 Naming a New Trading Partner



Create a New Trading Partner

Please specify the Trading Partner to create

Name : * tpNewTradingPartner

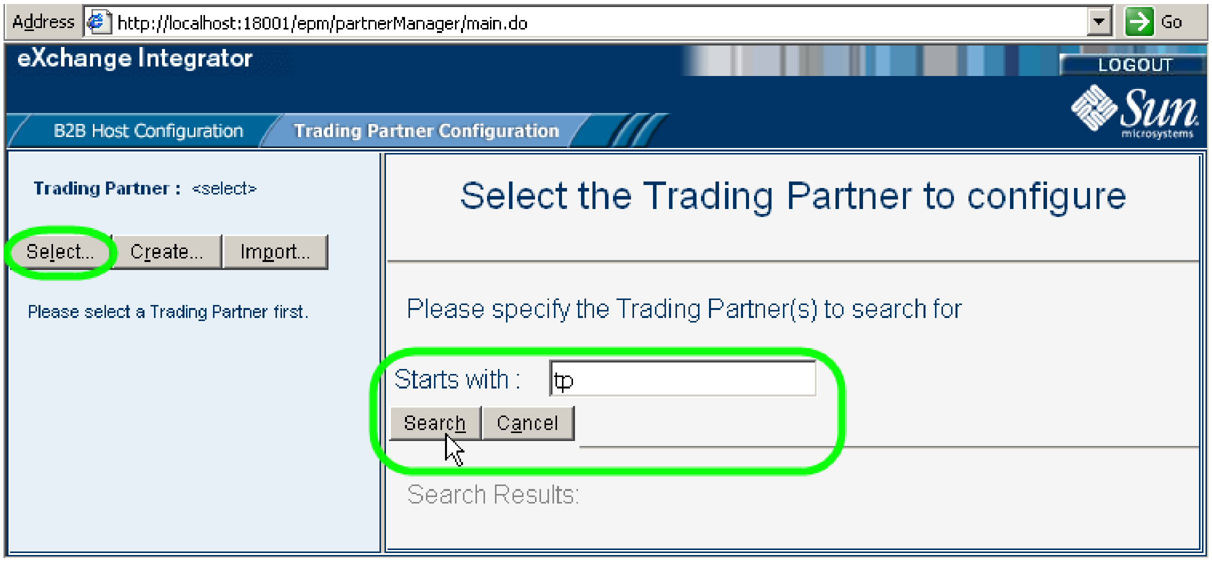
3 Click the **Create** button at the bottom of the canvas.

Result: The new trading partner appears in the explorer tree in the left pane.

To find and select an existing trading partner

1 With the **Trading Partner Configuration** tab active, in the explorer pane, click **Select**. Then, in the canvas, optionally after entering the first few characters of a possible name, click **Search**. See Figure 82.

Figure 82 Searching for a Trading Partner



Address http://localhost:18001/epm/partnerManager/main.do

Trading Partner : <select>

Please select a Trading Partner first.

Select the Trading Partner to configure

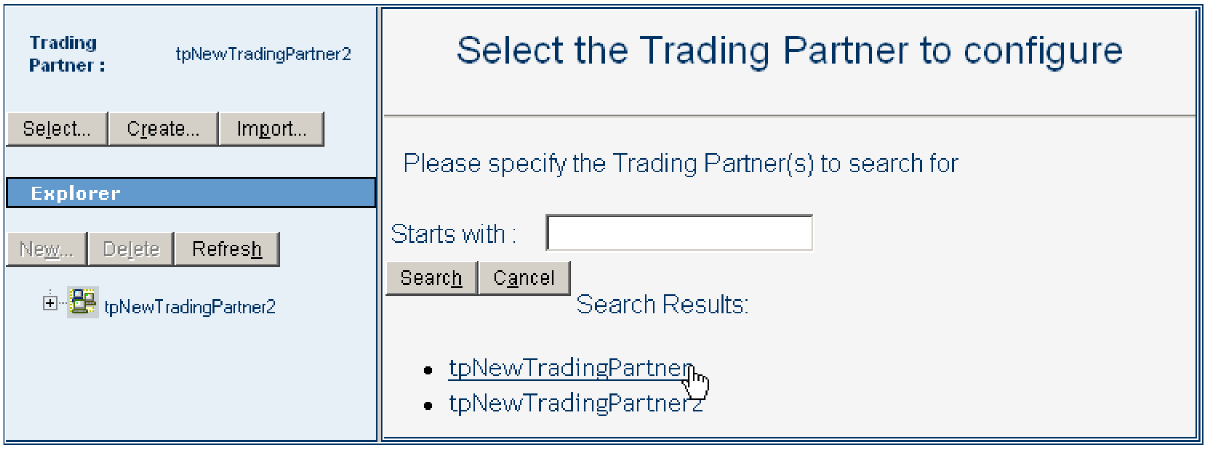
Please specify the Trading Partner(s) to search for

Starts with : tp

Search Results:

2 From the **Search Results** list, select the trading partner you want. See Figure 83.

Figure 83 Selecting a Trading Partner



Select the Trading Partner to configure

Please specify the Trading Partner(s) to search for

Starts with :

Search Results:

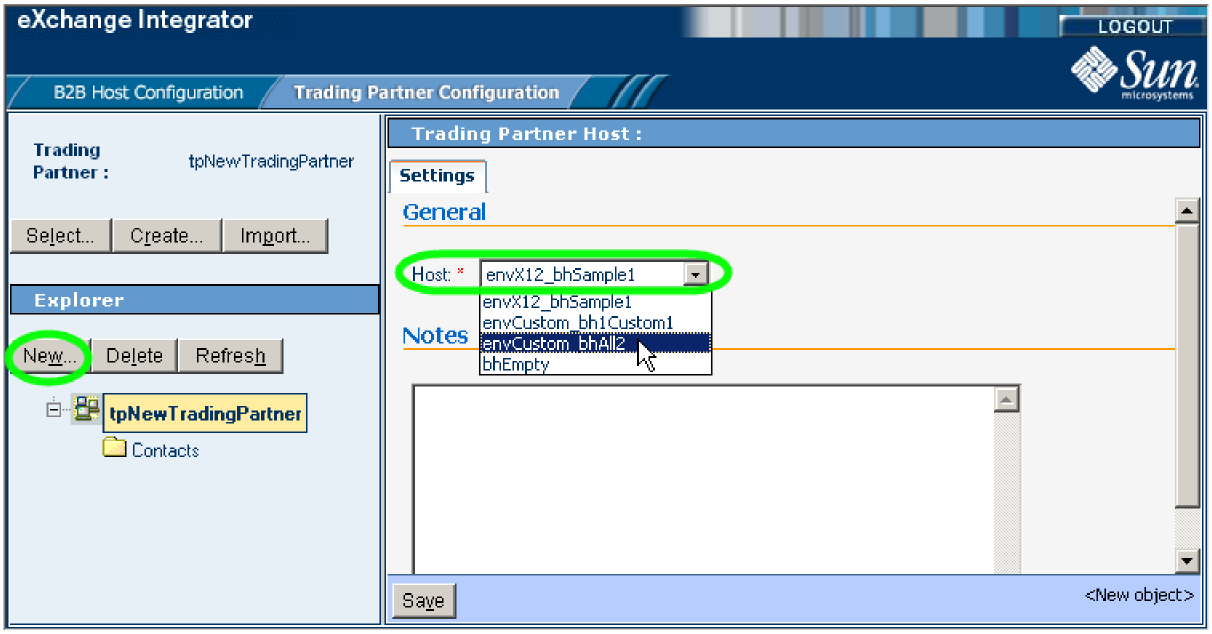
- tpNewTradingPartner
- tpNewTradingPartner2

Result: The new trading partner appears in the TP explorer tree in the left pane.

To add a B2B Host to a trading partner

1 In the TP explorer tree, select the trading partner and click **New**. Then, in the canvas (under the **Settings** tab), select a B2B Host from the pull-down list. See Figure 84.

Figure 84 Configuring a Trading Partner With a New B2B Host



Host: * envX12_bhSample1

2 Click **Save**.

Result: The new TP host appears in the TP tree in the left pane. All of its parameter settings apply to the trading partner unless overridden.

6.8.2 Configuring Trading Partners

Configuring a trading partner consists mainly of creating/configuring TP transaction profiles; each TP transaction profile inherits from the host transaction profile. You can then, optionally, override parameter settings at any level of the TP inheritance chain. You can also add one or more contacts that are specific to the TP.

This section presents the following procedures:

- **To create and configure a TP transaction profile** on page 133
- **To configure TP protocols, action groups, actions, transports, and contacts** on page 133
- **To create and configure a TP-specific contact** on page 134

To create and configure a TP transaction profile

1. With the **Trading Partner Configuration** tab active, in the TP explorer tree, open the TP and the host to display its five folders: Business Protocols, Delivery Protocols, Transports, Transaction Profiles, and Contacts.
2. Click the host's **Transaction Profiles** folder, make a selection from the B2B Host Transaction Profiles pull-down list, and click **Save**.

   Result: The new TP transaction profile group appears in the TP tree in the left pane. If the TP tree does not display the new TP transaction profile group, click **Refresh**.
3. Optionally, you can configure one or more TP transaction profiles by following the steps below, repeating this step as needed for additional TP transaction profile groups.

   A In the TP explorer tree, open the TP transaction profile group and select the inbound TP transaction profile; then use the canvas to override TP transaction profile parameter settings as needed, and click **Save**.

   B Within the same TP transaction profile group, select the outbound TP transaction profile; then use the canvas to override TP transaction profile parameter settings as needed, and click **Save**.
4. Repeat steps 2 and 3 as needed for additional TP transaction profile groups.
5. Optionally, if you want to override settings at a higher level than the TP transaction profile, you can configure one or more of the TP protocols, action groups, actions, and transports.

Note: *Although you must create at least TP transaction profile group for every trading partner, you need not create TP protocols, action groups, actions, or transports; all these are created automatically when you create a TP transaction profile. As needed, you can configure any of the items in the TP inheritance chain.*

To configure TP protocols, action groups, actions, transports, and contacts

1. Optionally, you can create a new TP business protocol: Click the host's **Business Protocols** folder, make a selection from the Host Business Protocol pull-down list, and click **Save**.
2. If you want to configure a TP business protocol, follow these steps:

A In the TP explorer tree, select the TP business protocol; then use the canvas to override TP business protocol parameter settings as needed, and click **Save**.

B With the TP business protocol selected, click **New** and create a new TP business action group by selecting a host business action group and clicking **Save**.

C In the TP explorer tree, select the TP business action group; then use the canvas to override TP action group parameter settings as needed, and click **Save**.

D With the TP business action group selected, click **New** and create a new TP business action by selecting a host business action and clicking **Save**.

E In the TP explorer tree, select the TP business action; then use the canvas to override TP business action parameter settings as needed, and click **Save**.

3 Optionally, you can create a new TP delivery protocol: Click the host's **Delivery Protocols** folder, make a selection from the Host Business Protocol pull-down list, and click **Save**.

4 If you want to configure a TP delivery protocol, follow these steps:

A In the TP explorer tree, select the TP delivery protocol; then use the canvas to override TP delivery protocol parameter settings as needed, and click **Save**.

B With the TP delivery protocol selected, click **New** and create a new TP delivery action group by selecting a host delivery action group and clicking **Save**.

C In the TP explorer tree, select the TP delivery action group; then use the canvas to override TP action group parameter settings as needed, and click **Save**.

D With the TP delivery action group selected, click **New** and create a new TP delivery action by selecting a host delivery action and clicking **Save**.

E In the TP explorer tree, select the TP delivery action; then use the canvas to override TP delivery action parameter settings as needed, and click **Save**.

5 Optionally, you can create a new TP transport: Click the host's **Transports** folder, make a selection from the Delivery Channel pull-down list, and click **Save**. If the TP tree does not display the new TP Transport, click **Refresh**.

6 If you want to configure a TP transport: Select the transport in the TP explorer tree, use the canvas to override TP transport parameter settings as needed, and click **Save**.

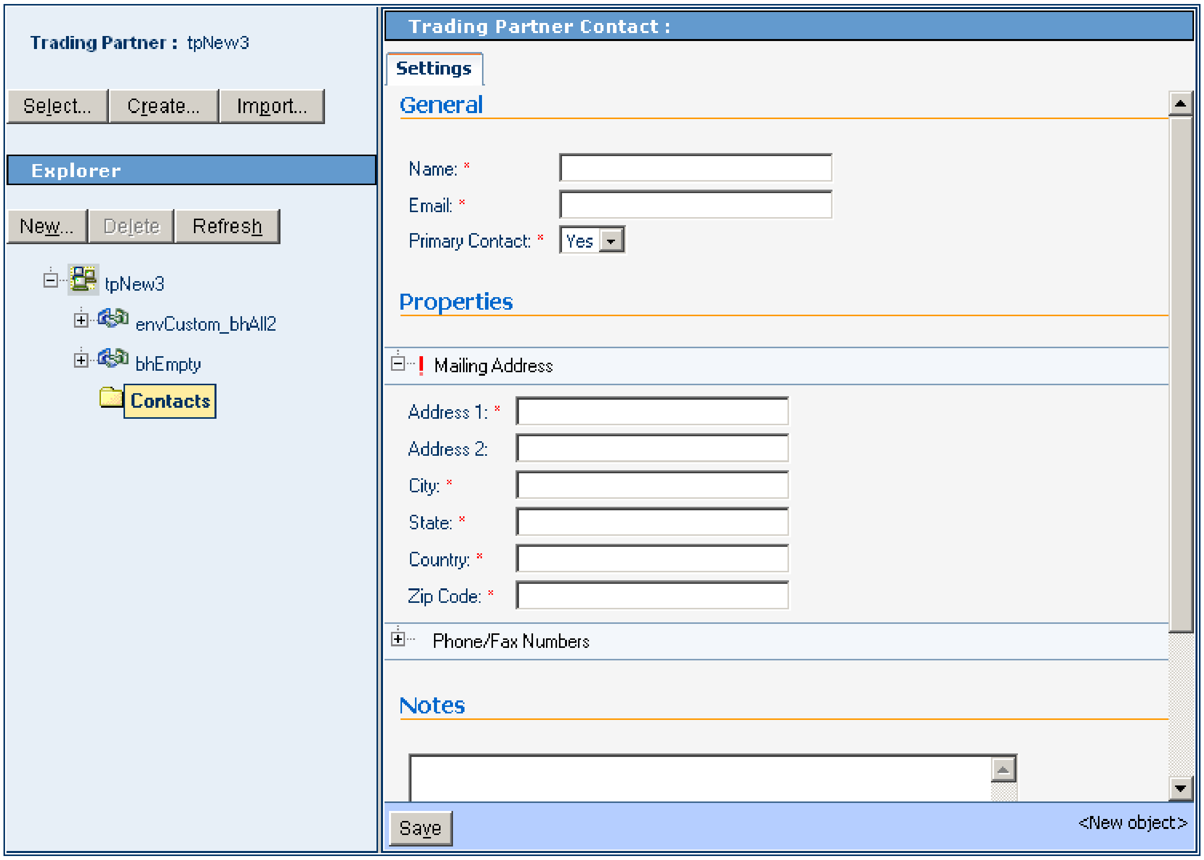
7 Optionally, you can create a new TP contact based on a host contact: Click the host's **Contacts** folder, make a selection from the Trading Partner Host Contact pull-down list, and click **Save**. Then select the contact in the TP explorer tree, use the canvas to override TP contact parameter settings as needed, and click **Save**.

To create and configure a TP-specific contact

1 With the **Trading Partner Configuration** tab active, in the TP explorer tree, open the TP to display its host(s) and its **Contacts** folder.

2 Click the TP-level **Contacts** folder, and then click **New**. See Figure 85.

Figure 85 Creating a New TP-Specific Contact



Trading Partner : tpNew3

Primary Contact: * Yes

3 Supply information for all required fields and, as needed, any optional fields, and then click **Save**.

Result: The name of the new contact appears in the TP tree, under the TP's **Contacts** folder.

6.9 Message Tracking

Included with eXchange is a special application named **tracker**. After it is connected to an Oracle database and deployed to a domain, it keeps track of all messages sent and received. The corresponding Web-based tool is named **Message Tracking**. It allows you to monitor the status of messages as they are received and processed through the B2B Suite.

This section explains how to access and use Message Tracking.

6.9.1 Before You Begin

- Your domain (Logical Host) must already be running, as well as your Oracle and LDAP external systems.
- You must already have deployed the appropriate Projects.
- For Message Tracking to be useful, there must be one or more messages that have already been picked up by the current domain's Integration Server.

6.9.2 Accessing Message Tracking

To access Message Tracking

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

   `http://``<logicalhost>`**`:`**`<port+1>`**`/`**`<objname>`**`/msgTrack/PkgTrack.do`**

   Where:

   - *`logicalhost`*—The hostname or IP address of the machine where your Project is deployed; the machine name of the Logical Host.
   - *`port+1`*—Typically 18001, 19001, 28001, or the like. The Web server connector port configured in your Integration Server. To discover this information: In Environment Explorer, right-click the current Logical Host and view its properties. Open **IS Configuration > Sections > Web Container > Web Server > Default Web Server**; `port` is the value set for **Connector Port**. If you have several Web server configurations, check them also.
   - *`objname`*: The name of the Message Tracking instance as it appears on the Connectivity Map. By default, the name that appears in cmTracker is `tracker`.

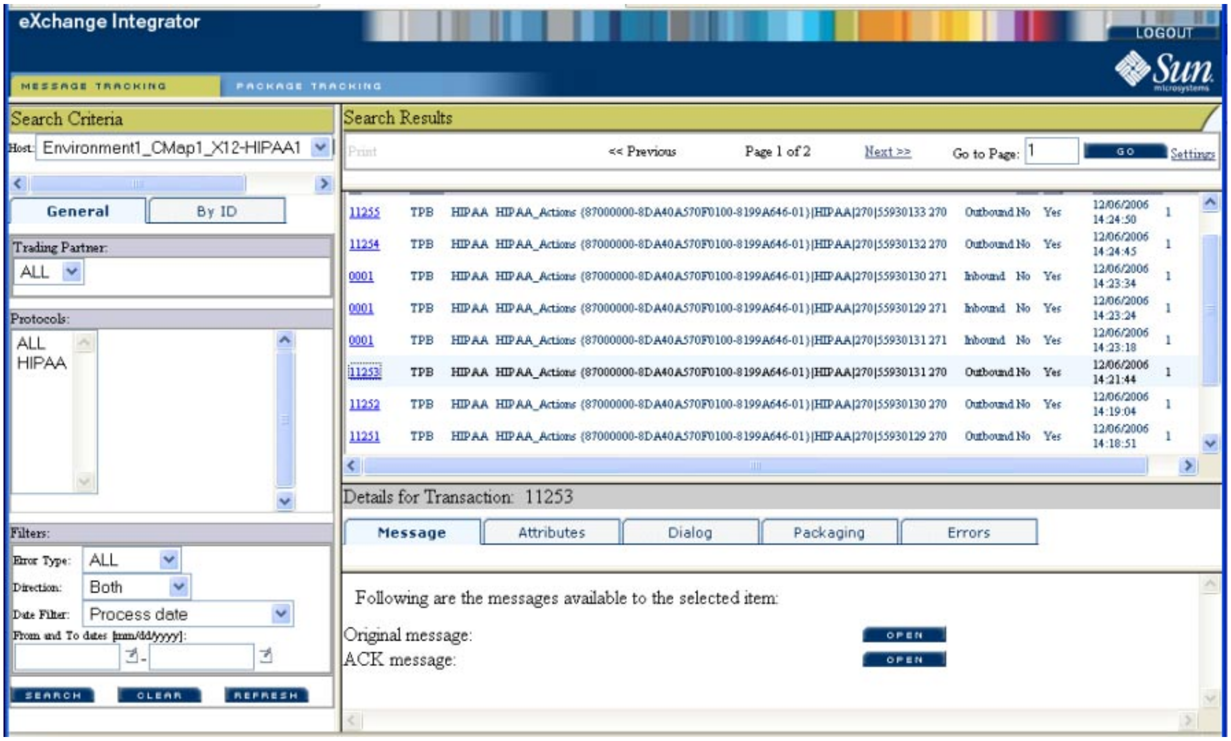
Example: To access Message Tracking for a typical scenario, use the following URL:

- `http://localhost:18001/tracker/msgTrack/PkgTrack.do`

6.9.3 Using Message Tracking

When you first run Message Tracking, the **Message Tracking** window appears, with no data. The controls in the left pane allow you to search and filter data. As messages are found that meet the search criteria, results are populated in the right pane. See Figure 86.

Figure 86 Message Tracking Window



Host: Environment1_CMap1_X12-HIPAA1

Trading Partner: ALL

Error Type: ALL

Direction: Both

Date Filter: Process date

From and To dates [mm/dd/yyyy]:

Page 1 of 2

Go to Page: 1

11255	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930133	270	Outbound	No	Yes	12/06/2006 14:24:50	1
11254	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930132	270	Outbound	No	Yes	12/06/2006 14:24:45	1
0001	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930130	271	Inbound	No	Yes	12/06/2006 14:23:34	1
0001	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930129	271	Inbound	No	Yes	12/06/2006 14:23:24	1
0001	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930131	271	Inbound	No	Yes	12/06/2006 14:23:18	1
11253	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930131	270	Outbound	No	Yes	12/06/2006 14:21:44	1
11252	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930130	270	Outbound	No	Yes	12/06/2006 14:19:04	1
11251	TPB	HIPAA	HIPAA_Actions	{87000000-8DA40A570F0100-8199A646-01}|HIPAA|270|55930129	270	Outbound	No	Yes	12/06/2006 14:18:51	1

Details for Transaction: 11253

Following are the messages available to the selected item:

Original message:

ACK message:

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, and then click GO.
2. Under Trading Partner, either click ALL or choose a particular trading partner from the drop-down list.
3. Under Protocols, either click ALL or choose a protocol 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. The navigation links (Previous, Next, and Go to Page) allow you to see other pages of ten results each.

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, and then click GO.
2. For Protocols, either click ALL or choose a particular protocol from the list.

3. For Package Type, either click ALL or choose a particular packaging protocol from the drop-down list.
4. For ID, enter a string for matching the message ID.
5. 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.

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

Purpose: After performing a search, or after setting up a search using either of the two previous procedures, 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.

To obtain details of a specified package

Purpose: On a package-by-package basis, you can examine the message text.

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 the original message.

Result: The text is displayed. You can use cut/copy/paste on any text in the window.

Chapter 7

Designing Business Processes

You can use eXchange to configure the components depicted by each activity in your Business Processes. This chapter provides the background information you need to create and understand eXchange BPs; for detailed technical information, also see the *eXchange Developer's Guide*.

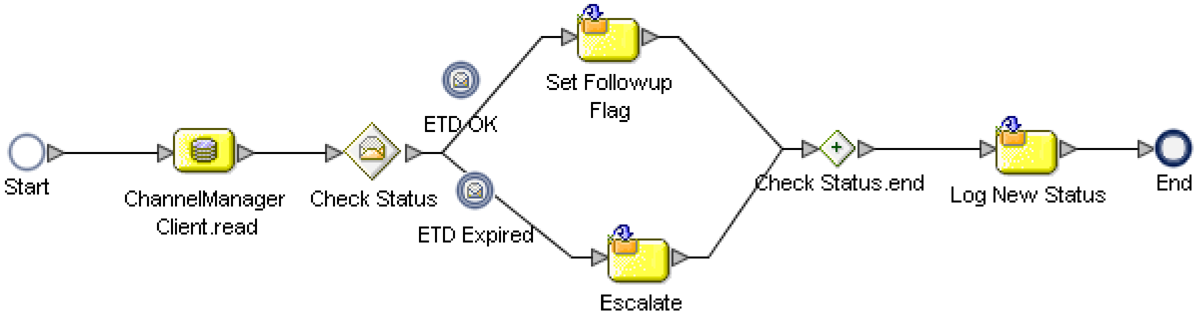
What's in This Chapter

7.1 Building a Business Process

A business process is a collection of actions that take place in your company, revolving around a specific business practice. These processes can involve a variety of participants and may include internal and external computer systems or employees. In eInsight, you create a graphical representation called a *Business Process*, or *BP*.

A business process modeled in the GUI look something like Figure 87.

Figure 87 Sample Business Process



Add a BP to your Project

Adding a BP to your project provides an empty modeling canvas where you add and manipulate items on the canvas, called *activities*. Before you can model your business process, you must add a new Business Process to your project.

Note: *The eGate User's Guide has detailed information on creating a project.*

1. In Enterprise Designer, in Project Explorer, right-click the project and, on the popup context menu, point at **New** and click **Business Process**.
2. Enter a new name for your BP.

7.1.1 Modeling a Business Process in eXchange

To model a business process in eXchange, drag and drop modeling elements on the eXchange Protocol Designer, and then link these components to reflect the logical flow of the business process. eXchange provides the tools you need to quickly develop BP models, including graphical editing tools to help you adjust, size, and align model components.

eXchange Protocol Designer

Once you create a new BP, you will build your model in the eXchange Protocol Designer (as shown in Figure 88). The eXchange Protocol Designer is the area in the Enterprise Designer where you view, create, and edit your BP.

To create a Business Process

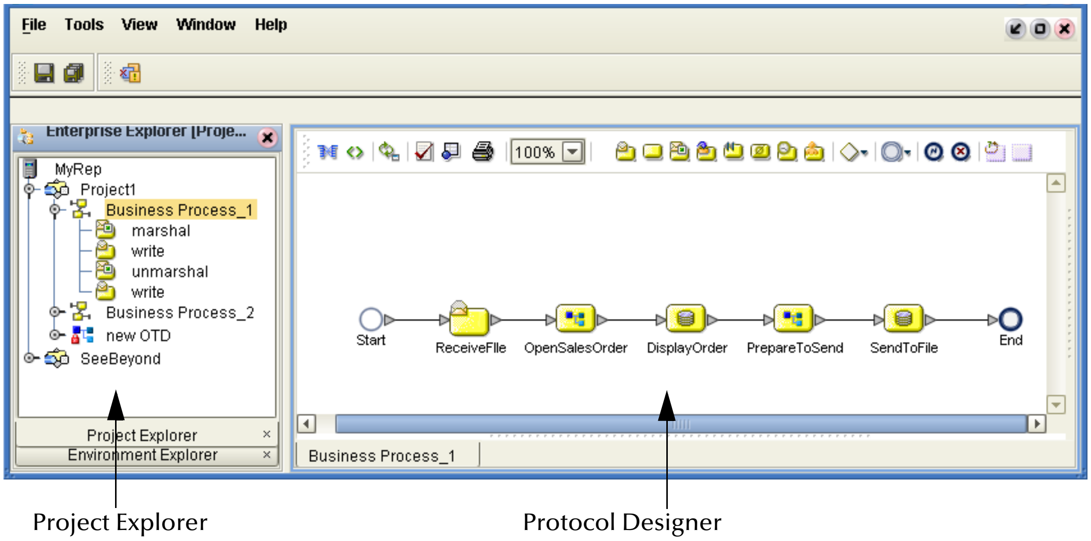
Begin designing your BP by dragging and dropping modeling elements from the toolbar onto the eXchange Protocol Designer canvas.

The **Start** and **End** activity appear on the blank canvas by default. There is only one starting point for any BP. (There can be multiple end points.)

1. Drag the appropriate modeling elements to your blank BP to the eXchange Protocol Designer canvas. See Figure 88.

Note: *See* **Appendix B** *for a complete list of modeling element options.*

Figure 88 Business Process Being Created in Protocol Designer



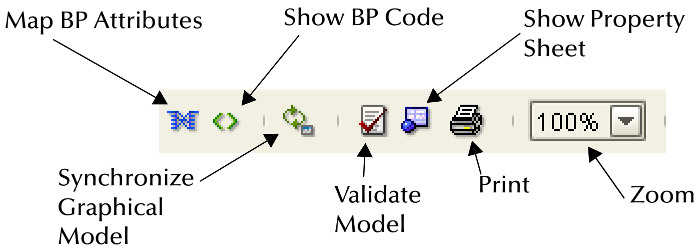
2 Draw links between the modeling elements to show the process flow (Figure 88)

3 On the main toolbar, click **Save** to save your changes to the Repository.

This will validate your BP, generate the code to run it, and save your changes to the Repository.

7.2 Using the eXchange Protocol Designer GUI

Figure 89 Toolbar Options



- Map Business Process Attributes—Opens the Business Rule Designer.
- Show BP Code—Toggles display of underlying BPEL code.
- Synchronize Graphical Model and BP Code—Causes the graphic model, the Business Rules, and the underlying BPEL code to match.
- Validate BP Model—Runs application to check syntactic validity.
- Show Property Sheet—Toggles display of property list and graphical overview.
- **Print**—Prints the BP graphic. Options allow you to control the scale.
- **Zoom**—Enlarges or shrinks the displayed graphic in the canvas.

7.3 Modeling Elements

The eXchange Protocol Designer is where the user creates the BP flow. It provides a palette of modeling elements for designing your BP. Like other logical components in a project, BPs appear in the Project Explorer tree.

Elements from the Enterprise Explorer can either be dropped onto empty canvas or onto an Activity. Many elements provide custom settings so that you can model every detail of your process. Each BP you create consists of basic elements as described in the following sections:

- **Activity Elements** on page 142
- **Branching Activities** on page 144
- **Intermediate Events** on page 145
- **Scope** on page 146
- **While** on page 146

Activity Elements

You can include several different kinds of activities and subprocesses in a BP. For examples of each of the different kinds of activities, see Table 21.

To add an activity

1. Either drag a modeling element from the toolbar or drag a web service operation from the Project Explorer, and then drop it where you want it on the canvas.
2. Click the activity name and begin typing to rename it from the default.

Note: *Every activity name must contain at least one character (A-Z, a-z, or 0-9); it must start with a letter or an underscore (_), and it may contain spaces.*

The activity appears on the modeling canvas.

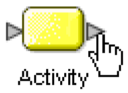
Link modeling elements

eXchange supports orthogonal and diagonal link styles – this setting applies to all links in a model and is an automated application of the style.

To link modeling elements, do the following:

1. Move your cursor over the connector portion of your modeling element.
2. Hold the cursor over the outside edge of the modeling element until it changes from the arrow pointer to a hand (see Figure 90).

Figure 90 Selected Activity



3 Click down, and drag a line from the first activity to the connector of the second activity. When the link attaches, release the mouse button.

Table 21 Activity Elements

Button	Command	Function
○▷ Start	Start Node	The Start Node is a modeling element indicating the start of the process. This element appears in the eXchange Protocol Designer by default, when you create a new BP. A Start Node can only link to an activity that has a **receive** or **read** capability, signaled by a subicon in the upper left resembling an opened envelope (see **Receive Activity** just below).
▷——▷ ▷—⋈—▷	Link Link with Business Rule	Links indicate the flow of the BP by connecting elements together. When you select a link, a context menu allows you to configure how data is going to be passed to and from the underlying component or web service operation using BP attributes. eXchange ensures the model is being properly linked because it does not allow invalid links to connect. Links can also accept mapped values. A link with mapped values is highlighted in blue.
▷○ End	End Node	The End modeling element indicates the completed state of a BP. This element appears in the Business Process Designer by default, when you create a new BP.
✉	Receive Activity	The Receive activity indicates the invocation of a BP or a wait state pending the arrival of an inbound message. The Receive activity represents the actual method by which a BP is initiated. For example: ▪ An eWay polls a file system or database and retrieves data that is passed to the engine, along with the indication that a BP instance has started. ▪ A user types a URL into a browser and a servlet initiates a BP by sending a message to eGate or eInsight.
▭	Activity	An activity is a step in the BP in which the engine will invoke a web service operation or an eGate component. Depending upon the configuration of the component, a response may or may not be required. One example would be a synchronous extraction process from a database to return the credit status of a trading partner.

Table 21 Activity Elements

Button	Command	Function
	Reply Activity	The Reply activity allows a BP to respond to the external system or user that originally invoked the BP. The original receive at the beginning of the BP is paired with the Reply at the end of the process. In cases where a message must be sent back to the caller of the BP, the Reply uses information that correlates the message in the calling system. A Reply acts as the last step in a BP in which the BP is acting as a web service operation or subprocess. A Reply correlates the outbound message back to the calling process; for example, it can reply to an external system as a web service operation.
	Business Rule Activity	The Business Rule activity sets data values, including task assignments. It is used when imported models have multiple data mappings between the invocation of human tasks or automated systems.
	Compensate	The Compensate element invokes compensation on an inner scope that has already completed normally. This construct can be invoked only from within a fault handler or another compensation handler.
	Empty Activity	The Empty activity allows data to pass through without any changes.
	Wait Activity	The Wait activity acts as a timer. The user will build a model in which there are two simultaneous paths within a set scope, one for the BP and one for the timer. If the timer condition takes place first, an exception will be thrown and handled, and the BP path will then be abandoned.
	User Activity	The User activity is used only by eInsight, and should not be placed on a canvas unless your site is licensed for eInsight was well as eXchange. It is used when assigning, escalating, or otherwise using human intervention to complete eInsight business process tasks.

Branching Activities

Branching activities are objects you add to your BPs to specify the logical flow of information. eXchange provides three different kinds of branching activities: Decisions, Event Based Decisions, and Flow.

Add a Branching activity

To add a Branching activity to the modeling canvas:

1. On the toolbar, click the **Branching Activities** drop-down icon, and then release the mouse button.
2. Point at the type of Branching activity you want to add, click, and then drag the activity from the toolbar to the eXchange Protocol Designer canvas.

   The selected Branching activity appears on the modeling canvas.

Table 22 Branching Activities

◇	Decision	A Decision allows one of several possible paths to execute, based on expression logic. This element is used to create complex expressions that determine the path of the BP. It also contains the expression and connection names. Decisions allow you to define expressions that are evaluated to determine the proper BP flow. Expressions are built using the mapping interface and BP attributes.
◈	Event Based Decision	Multiple possible messages can be juxtaposed against a timeout condition to allow the type of message received to determine the appropriate BP path.
◈	Flow	Allows you to specify one or more activities to be performed concurrently.

Intermediate Events

Intermediate events are those activities that can interrupt the flow of a BP. Some intermediate events handle exceptions that may occur during your BP or compensate for exceptions that occur.

Add an Intermediate event

To add an **Intermediate event** to the modeling canvas:

1. On the toolbar, click the **Intermediate Events** drop-down icon, and then release the mouse button.
2. Point at the type of Intermediate event you want to add, click, and then drag the activity from the toolbar to the eXchange Protocol Designer canvas.

   The selected Intermediate event appears on the modeling canvas.

Table 23 Intermediate Events

⏪	Compensation Handler	Used when something in a BP fails and requires a rollback or upstream activities (like money has to be returned to the customer). On an automatic basis in the BP, upstream steps in the BP are notified that the failure has occurred and certain transactions need to be reversed, sometimes in a sequential order. The compensation handler allows you to design the process and circumstances in which the compensation takes place.
Ⓝ	Catch Named Exception	Each automated system (back-end system) or web service operation can publish their possible error codes (for instance, fault 15 is “bad data”). Those codes can be mapped to exception handlers. Each exception handler is connected to the scope that surrounds one or more steps in a BP. The components within that scope will throw the exceptions when things go wrong and the exception handler will automatically initiate the appropriate process to handle the problem.
Ⓝ*	Catch All Exceptions	This exception handler is configured to handle all exceptions that occur in a scope.

Table 23 Intermediate Events

	Message Event	This is similar to a Receive Activity, but it occurs only in the middle of a BP. Each of these elements can be a different message.
	Timer Event	A timeout condition is set upon Activities, sets of Activities, or a BP as a whole, to ensure that processes complete within given amount of time. Timeout conditions also allow you to design the BP branch to take after a timeout condition takes place.

Scope

The behavior for one or more activities can be defined by a scope. A scope can provide exception handlers, event handlers, a compensation handler, and data variables. The exception handlers for the scope can be used to catch the faults caused by the possible exception responses.

	Scope	The Scope element allows you to apply exception handlers, compensation, and transactionality to a set of sequential or simultaneous steps in a BP.

While

	While	This allows you to create a looping process within a BP (for instance, a negotiation process may take several weeks, but the manager wants to review the daily status). The loop continues until the negotiation is complete, and then the BP continues.

7.3.1 Validating a Business Process

After creating a BP, you can check to see if there are any problems such as activities that are not connected or an incorrect number of output links from an activity.

To check the BP for errors

- On the **toolbar, click Validate BP Model.**

  If an error is encountered, a message box displays information about the error. If there are no errors, a message appears stating that there were no errors**.**

Note: *If an error message displays, see* **"Saving an Unfinished Business Process"** *for information on repairing errors. Repairing the error may entail such items as adding logic to Decisions or adding attributes to activities.*

7.3.2 Saving an Unfinished Business Process

Even if a BP is not complete and/or contains errors, you can save it as a work in progress and return to it later by doing any of the following:

- On the **File** menu, choose **Save**
- On the main toolbar, click **Save**

- On the keyboard, press **Ctrl+S**

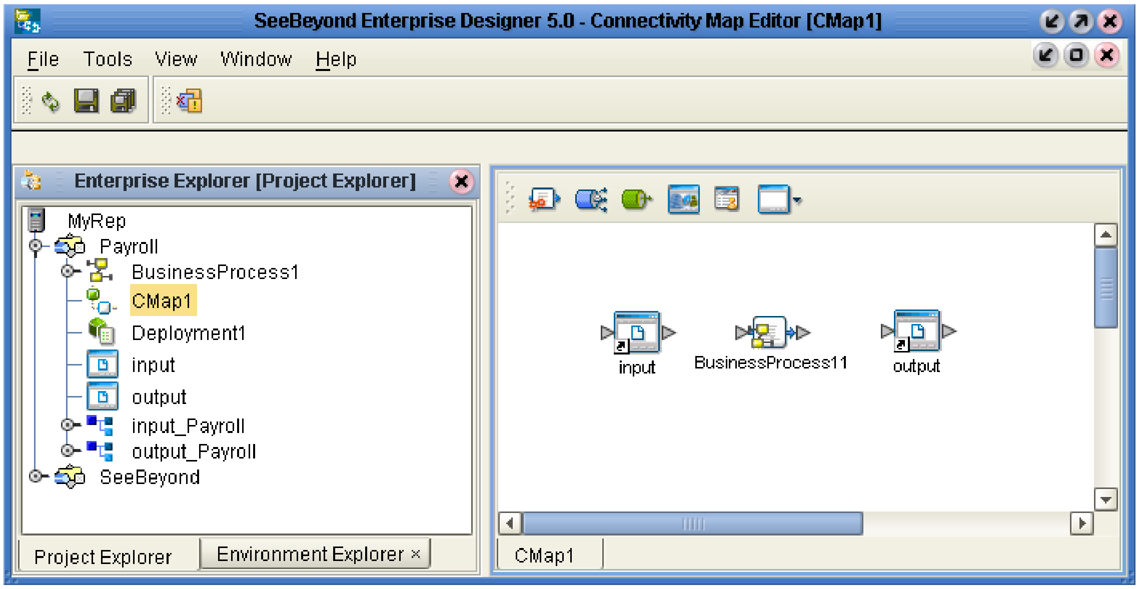
7.4 Using Business Processes in a Connectivity Map

The connectivity map represents connection information in Java CAPS. The flow is represented at a higher level than in the BP. eXchange also uses the information in the connectivity map to establish and maintain connections to systems for the correct step in a BP.

To include a BP as a service on a connectivity map

1. In the Connectivity Map Editor, drag a BP onto the canvas.
2. Add and connect other components and external systems as needed. See Figure 91.

Figure 91 Connectivity Map with BP

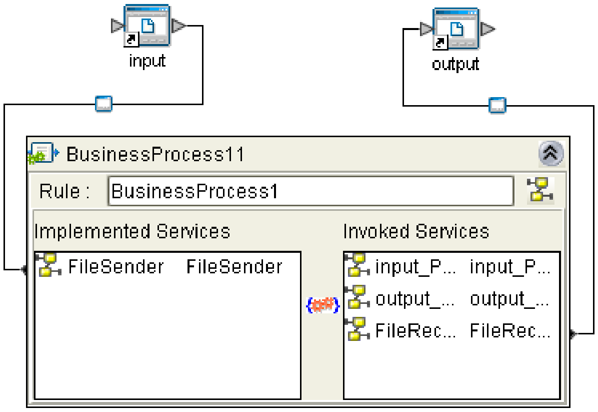


To connect the BP activities to the externals

1. In the map, double-click the BP to open the Binding Dialog.
2. Connect the appropriate activities to the corresponding external.

   Note that **Receive** activities appear in the left pane, and **Invoke** and **Reply** activities appear in the right pane. See Figure 92.

Figure 92 Connectivity Map: B2B Protocol Binding



Chapter 8

Configuring Exception Handling

This chapter explains the concept of exception handling and how to configure various methods of handling errors.

What's in This Chapter

8.1 Overview

Exception handling is the identification of failed components or systems. In eXchange, exception handling allows one or more components to throw an exception that is caught by eXchange within a *scope*. Using the **scope** element, you can configure eXchange to catch all exceptions or certain exceptions that you specify. The elements that you use to configure exception handling in your model are:

- **Catch Named Exceptions**
- **Catch All Exceptions**

Exception handling in B2B protocols relies heavily on the concept of *compensation*. Compensation is an application-specific activity that reverse the effects of a previous activity that was carried out as part of a larger unit of work that is being abandoned.

B2B protocols are often of long duration and use asynchronous messages for communication. They also manipulate sensitive business data in back-end databases and line-of-business applications. As a result, the overall business transaction may fail or be cancelled after many transactions have been committed during its progress. In these cases, the partial work may need to be reversed.

8.1.1 Exception Handling Configuration

Exception handlers are configured to catch errors that are thrown by eGate components and/or Web Services. These systems can be configured to publish one or more exceptions.

- **Manual Exception Handling**: The model can contain B2B protocol logic designed to handle the exception.
- **Automatic Exception Handling**: Pre-packaged functionality guides the user to create multiple types of catches for thrown exceptions.

Each exception can be handled differently. This is one example:

1. Build the exception handling logic as a B2B protocol.
2. Select the exception handler to configure which exception triggers the exception handling process.
3. Drag the **Scope** element onto the eXchange Protocol Designer canvas.
4. Drag the **Exception** modeling element into the scope for which it should take effect.
5. Define a B2B protocol that appropriately handles each exception.
6. Model manual exceptions in a B2B protocol.
7. Configure the exception handler to take place when one of the components within the **Scope** throws the appropriate exception.

Identifying Component or System Failures

Exception management allows users to quickly identify and correct problems with components or systems.

Users can filter the list of displayed instances to quickly identify exceptions.

Users can easily navigate to particular versions of a B2B protocol to monitor the progress of instances.

A Web-based interface allows users to securely access the monitoring environment over the Internet.

Identification of troubled instances, such as time-outs or bad messages.

Failed components/systems create visual alerts via the B2B protocol monitoring interface. The integrated monitoring environment allows you to identify the problem, assign a resource to fix the problem, and if necessary, restart the affected instances.

Users can quickly identify troubled instances from a large number of instances, repair and restart that instance for continued processing.

8.2 Scope

Scope allows you to define a range

- For handling of exceptions
- For creating compensation logic

The range of the scope can span one or more activities in the B2B protocol or even the entire B2B protocol.

Scope or Process-level exceptions

Either **Catch Named Exception** or **Catch All Exceptions** can be used at the B2B protocol level.

Catch Named Exception

1. Drag the **Catch Named Exception** element into the scope for which the exception handler applies.
2. In the Exception Handler properties, configure the following:
   - Fault Container—The output Attribute that will be containing the run-time name of the thrown fault.
   - Fault Name—The run-time value for the exception that will be passed from the component to the engine at run time.

Note: *The fault name is auto-populated with values based on the components dragged to the editor.*

3. Select the configuration control for the Exception Handler – the properties pane will appear to select the Fault name and container.
4. Drag the **Catch Named Exception** into the associated scope.

Catch All Exceptions

No configuration of the Catch All Exceptions element is required; any thrown exception not previously caught is caught with the Catch All Exceptions element.

8.3 Compensation

Compensation allows the modeler to create the process flow for executing complex compensations. Exception Handlers for parent scopes invoke the correct Compensation Handlers in the appropriate order.

Using Scope and Exceptions to Trigger Compensation

- Compensation Activity—In an exception handler, initiates the compensation process. It models the compensation as a B2B protocol, and indicates the Compensation for “DB Insert” should be initiated.
- Compensation Handler— This is dropped within a scope to create the compensation logic for a given scope.

8.4 Validating the B2B Protocol

After generating the business process code (BPEL), you can click the **Validation** button on the toolbar to identify any issues with the model. The validation results now appear in a wizard, listing any issues one by one with clear and understandable descriptions for the issues. You can fix each issue, regenerate the business process code, and again view the validation results until each of the issues has been fixed, and the model validates as correct.

Chapter 9

Persistence and Monitoring

This chapter discusses setting up design-time components for persistence and monitoring. These features are optional; if used, they require a different database, called an *eInsight engine* database, to collect and persist data from your business processes (BPs).

For eInsight engine database requirements, see **"Database for Persistence and Monitoring via eInsight Engine" on page 29**.

What's in This Chapter

Also See

9.1 Overview

eXchange provides scripts to create the eInsight database schema, which can be used to collect and persist data from your BPs. (The eInsight database schema is optional, and is independent of the eXchange database schema required to collect and persist data about your trading partner profiles.) As needed, you can set up several different eInsight database instances.

After the eInsight engine database schema has been created and configured for use by a particular database username, the instance must be referenced by the BPs whose activities are to be persisted by it. It must also be referenced by all Integration Servers that host the BPs.

Once a specific BP has been set up to so that its data is persisted by an instance of an eInsight engine database, you can additionally configure the BP so that its activities can be logged and monitored by Enterprise Manager.

Procedures

The following steps are required for eInsight database and engine setup:

- **Setting Up the Database Schema for eInsight Engine** on page 154
- **Configuring the eInsight Engine for Runtime** on page 155

After the schema and engine(s) have been set up, follow these steps for each BP:

- **Configuring Specific BPs to Use Persistence** on page 158
- **Turning Monitoring On or Off for Specific BPs** on page 159

For instructions on using Enterprise Manager to monitor BP activity, see:

- **Overview of Business Process Monitoring** on page 160

9.2 Setting Up the Database Schema for eInsight Engine

To create the run-time recoverability database schema, you must extract and run the eInsight database scripts that are automatically installed with eXchange.

9.2.1 Exporting and Extracting Database Scripts for eInsight

eXchange supplies eInsight engine database scripts as compressed files. These files cannot be run in place, and must instead be exported and extracted.

To export and extract the eInsight engine database schema scripts

1. In Enterprise Explorer, in the project tree, expand the following folders: **Sun SeeBeyond > eInsight > Download Database Scripts**
2. Right-click the **.zip** file associated with the appropriate database (oracle.zip, db2.zip, sybase.zip, or sqlserver.zip) and, on the popup context menu, click **Export**.
3. Save the **.zip** file to a local folder.
4. Extract the **.zip** file contents to a local folder, which will contain:
    - **install_db.bat** or **install.sh**—Creates the tablespace, users, tables, stored procedures, and initial values.
    - **uninstall_db.bat** or **uninstall_db.sh**—Reverses what the install_db script creates; that is, it drops tables and users, and deletes stored procedures.
    - **clear_db.bat** or **clear_db.sh**—Truncates all tables without performing any other uninstall actions.
    - *database-specific* **.sql** *scripts*—Called by the install_db and uninstall_db scripts. For example:, create_tables.sql, drop_tables.sql.
    - **Readme.txt**—Additional instructions specific to your database application.
5. For additional information, read the material in the Readme.txt file.

9.2.2 Running Database Scripts for eInsight

To create, modify, or delete the run-time recoverability database schema, run the corresponding eInsight database script that you extracted in the previous procedure. The database user that executes this script must have permission to create tables.

To install, truncate, or uninstall the eInsight engine database schema

1. Open a command prompt (on Windows) or (on UNIX) shell.
2. Change to the directory where you extracted the eInsight engine database scripts.
3. Enter one of the following commands.
   - To create the tablespace, users, tables, stored procedures, and initial values:

     ```
     install_db <user> <password> <tnsname>
     ```
   - To truncate the database tables created by the install_db script:

     ```
     clear_db <user> <password> <tnsname>
     ```
   - To undo the effects of the install_db script (in other words, to drop the tables, users, and stored procedures that were created):

     ```
     uninstall_db <user> <password> <tnsname>
     ```

   In each case:

   - *<user>* is the database username
   - *<password>* is the password for this database user
   - *<tnsname>* is the database or TNS name

Note: *The default user and password created from these scripts is: **einsight**. You can modify the user, password, and disk space allocated for tables and user permissions. Do not modify the table and column definitions.*

9.3 Configuring the eInsight Engine for Runtime

The eInsight engine provides many configuration options. This section describes the most important ones used by the B2B Suite; other B2B settings and recommendations for scalability, reliability, and performance can be found in **Appendix C**, **Appendix D**, and **Appendix E**. For a complete discussion of the eInsight engine options, refer to the *Java CAPS Deployment Guide* and the *eInsight Business Process Manager User's Guide*.

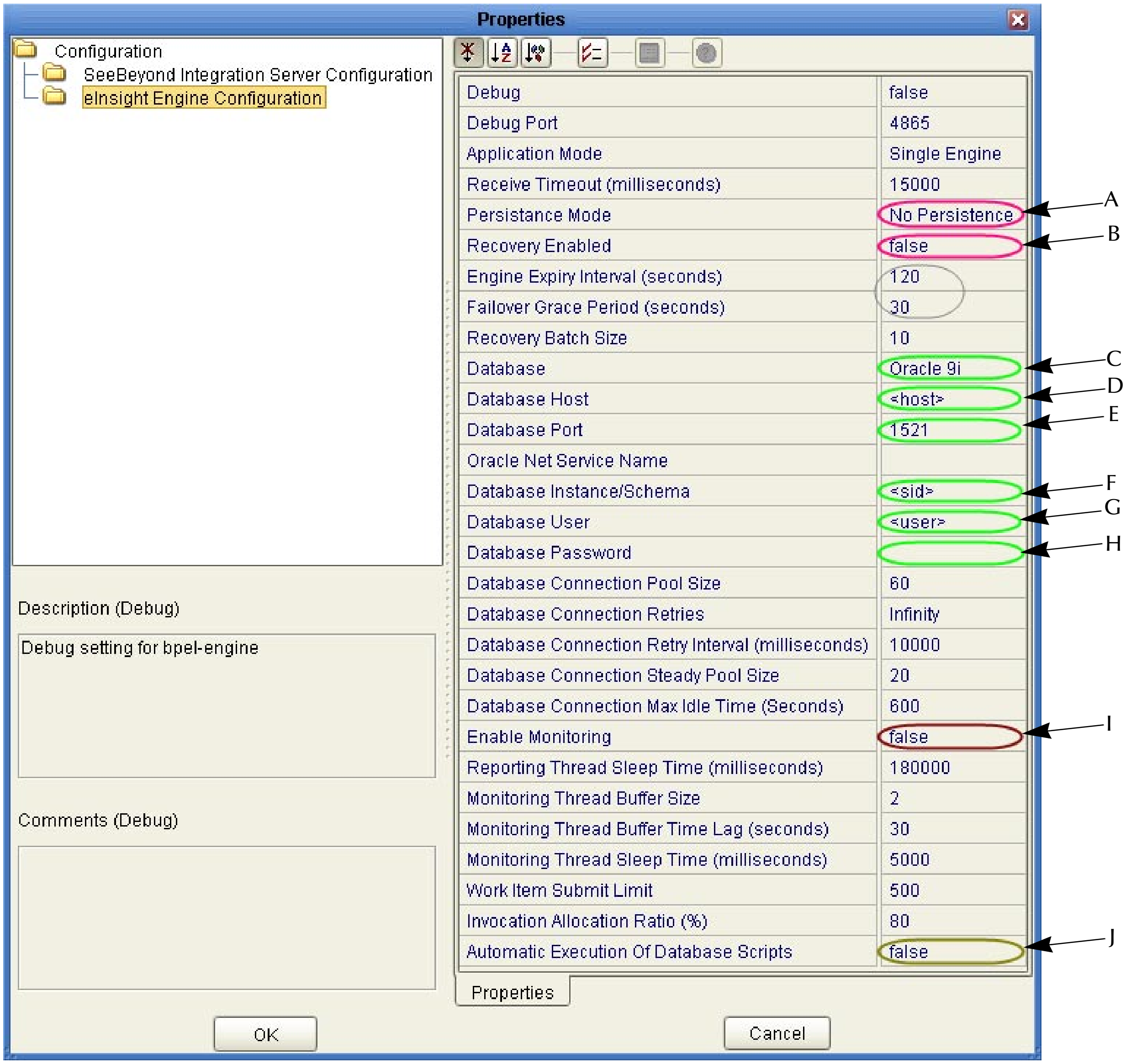
9.3.1 Basic Configuration Options for the eInsight Engine

To configure the eInsight engine in an Integration Server

1. In Environment Explorer, open the environment and Logical Host (checking it out, if necessary), right-click the **Integration Server** (IS), and, on the pop-up context menu, click **Properties**.

2 In the Properties dialog, open the Configuration > IS Configuration > **Sections** tree and click **eInsight Engine Configuration**. See Figure 93.

Figure 93 eInsight Engine Configuration


Properties
Configuration
SeeBeyond Integration Server Configuration
eInsight Engine Configuration
Properties
OK
Cancel
A
B
C
D
E
F
G
H
I
J

Description (Debug)

Debug setting for bpel-engine

Comments (Debug)

Debug	false
Debug Port	4865
Application Mode	Single Engine
Receive Timeout (milliseconds)	15000
Persistance Mode	No Persistence
Recovery Enabled	false
Engine Expiry Interval (seconds)	120
Failover Grace Period (seconds)	30
Recovery Batch Size	10
Database	Oracle 9i
Database Host	<host>
Database Port	1521
Oracle Net Service Name	
Database Instance/Schema	<sid>
Database User	<user>
Database Password	
Database Connection Pool Size	60
Database Connection Retries	Infinity
Database Connection Retry Interval (milliseconds)	10000
Database Connection Steady Pool Size	20
Database Connection Max Idle Time (Seconds)	600
Enable Monitoring	false
Reporting Thread Sleep Time (milliseconds)	180000
Monitoring Thread Buffer Size	2
Monitoring Thread Buffer Time Lag (seconds)	30
Monitoring Thread Sleep Time (milliseconds)	5000
Work Item Submit Limit	500
Invocation Allocation Ratio (%)	80
Automatic Execution Of Database Scripts	false

3 To enable persistence and recovery *without* load balancing and failover, configure the following:

A **Persistence Mode**—For simple persistence, change this from the default to: **Persist to Database - Single Engine (Recovery)**. For load-balancing, see **“Configuring eInsight Engines for Load Balancing” on page 157**.

B **Recovery Enabled**—For persistence, change this from the default to: **true**.

4 To connect to the eInsight database, configure the following:

C **Database**—Select from the list: **Sybase 12.5**, **Oracle 8.1.7**, **Oracle 9*i*, Oracle 10*g*, DB2 8.1** or **SQL Server 2000**.

D **Database Host**—Enter the name of the machine where your database resides.

E **Database Port**—Enter your database connection port number (default=1521).

F **Database Instance/Schema**—Enter the database instance/schema or SID.

G **Database User**—Enter the user name for your database.

H **Database Password**—Enter the password for this database user.

5 To enable monitoring via Enterprise Manager, configure the following:

I **Enable Monitoring**—For monitoring, change this from the default to: **true**.

6 To enable the automatic execution of the database scripts, configure the following:

J **Automatic Execution of Database Scripts**—Select **true**.

Note: *Until you are ready to optimize your eInsight engine(s) for performance, scalability, and reliability, do not change the default settings for other configuration properties.*

7 Click **OK**.

9.3.2 Configuring eInsight Engines for Load Balancing

When a Business Process needs to be scaled to meet heavier processing needs, you can distribute the Business Process across multiple engines to increase throughput. eInsight's load balancing algorithm automatically distributes processing across multiple engines; however, eInsight cannot load balance correlated messages.

To configure eInsight engines for load balancing

1 In the Properties dialog, open the Configuration > IS Configuration > **Sections** tree and click **eInsight Engine Configuration**. See Figure 93.

2 Change the setting for **Persistence Mode** to **Persist to Database - Multiple Engines** (in Figure 93, see item A).

3 Ensure that **Recovery Enabled** is set to: **true** (in Figure 93, see item B).

4 Configure all eInsight engines to share the same database.

Also see: **Monitoring Load-Balanced Business Process Instances** on page 168.

9.3.3 Configuring eInsight Engines for Failover

The eInsight engine failover capabilities can ensure throughput of running Business Process instances. When Business Process instances encounter an engine failure, eInsight load balances those instances across all available engines. As with load balancing, eInsight's failover capabilities are limited to non-correlated messages.

To configure eInsight engines for failover

Before you begin: Ensure that your eInsight engines are configured for load balancing (see **procedure on page 157**).

1 In the Properties dialog, open the Configuration > IS Configuration > **Sections** tree and click **eInsight Engine Configuration**. See Figure 93.

2. Change the setting for **Engine Expiry Interval** (in Figure 93, below item B) so that it registers itself as alive frequently enough to meet the demands of your system.

   Optimizing this setting might require some testing. This property also applies to the interval for the recovery of dangling instances. The default setting is 120.

3. Change the setting for **Failover Grace Period** to the optimal elapsed time period before moving running Business Process instances from an unavailable engine to an available engine. Optimizing this property setting might require some testing.

9.4 Setting Persistence and Monitoring for Individual BPs

Both persistence and monitoring require configuration for each BP individually. Monitoring can be turned on only for BPs that have been configured to use persistence.

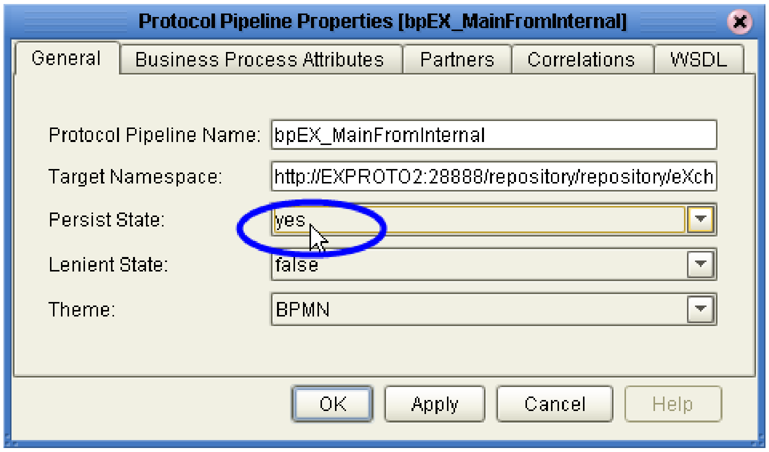
9.4.1 Configuring Specific BPs to Use Persistence

Persistence is set for each BP individually. The default setting is **no**.

To set persistence for a specific BP

1. In Enterprise Designer, in the project tree, right-click the BP (first use **Check Out**, if necessary) and, on the popup context menu, click **Properties.**
2. In the **General** tab (see Figure 94), change **Persist State** to **yes** and then click **OK**.

Figure 94 Setting Persistence for a BP



Protocol Pipeline Properties [bpEX_MainFromInternal]

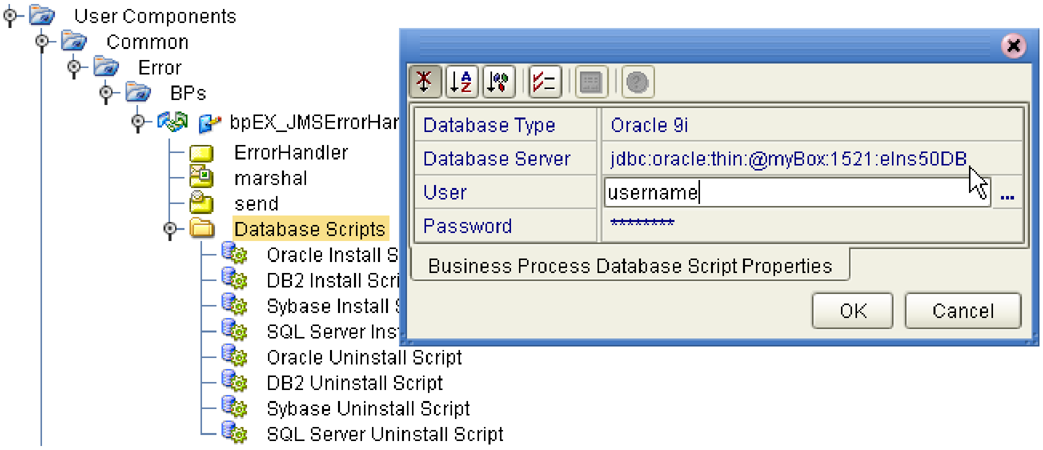
Protocol Pipeline Name: bpEX_MainFromInternal
Target Namespace: http://EXPROTO2:28888/repository/repository/eXch
Persist State: yes
Lenient State: false
Theme: BPMN

3. On the main Enterprise Designer toolbar, click **Save All**.

   The project tree displays a new **Database Scripts** under your BP. In Figure 95, see the highlighted item in the project tree.

4. Right-click **Database Scripts** and, on the popup context menu, click **Properties.**
5. In the Business Process Database Script Properties, supply the correct information for your database. See Figure 95.

Figure 95 Setting Database Script Properties for BPs



Database Type	Oracle 9i
Database Server	jdbc:oracle:thin:@myBox:1521:eIns50DB
User	username
Password	********

9.4.2 Turning Monitoring On or Off for Specific BPs

Monitoring is turned on or off for each BP individually, and only applies to BPs that have been configured to use persistence.

To start running a database script for a particular BP

1. In the project tree, open the BP and its **Database Scripts** folder (see Figure 95).

   If the BP does not have a Database Scripts folder, follow the **procedure on page 158**.

2. Right-click the appropriate database **Install** script and, on the popup context menu, click **Run.**

   Running this script completes the setup process for this BP, and allows you to use Enterprise Manager to monitor the BP at runtime.

To stop running a database script for a BP

1. In the project tree, open the BP and its **Database Scripts** folder (see Figure 95).
2. Right-click the appropriate database **Uninstall** script and, on the popup context menu, click **Run.**

Chapter 10

Monitoring Business Processes at Run Time

Enterprise Manager allows you to identify problems with components or systems. From Enterprise Manager, you can double-click Business Process (BP) components to go directly to a problem.

From Enterprise Manager, you can:

- Filter the list of displayed instances to identify exceptions.
- Navigate to particular versions of a BP to monitor the progress of instances.
- Securely access the monitoring environment remotely.

What's in This Chapter

10.1 Overview of Business Process Monitoring

After you have configured persistence and monitoring for your eInsight engines (see **"Setting Up the Database Schema for eInsight Engine" on page 154**) and for individual BPs (see **"Setting Persistence and Monitoring for Individual BPs" on page 158**) , you can use Enterprise Manager to monitor your Business Process Instances. The procedures in this section help you to ensure that the Business Process appears in Enterprise Manager as expected.

Before you connect to Enterprise Manager, verify that the Adobe SVG Plug-in for Enterprise Manager is installed. For detailed information about installing the Adobe SVG Plug-in, see the *Java Composite Application Platform Suite Installation Guide*. Before you begin monitoring eInsight Business Processes, you must deploy and manage any necessary application servers in Enterprise Manager. For detailed information about using Enterprise Manager for deploying and managing application servers, see the *Sun SeeBeyond eGate Integrator System Administration Guide.*

From the Enterprise Manager’s Business Process Instance Monitor tab, you can start, stop, and evaluate Business Process Instances. You can also refresh the Business Process Instance lists, change monitoring options, filter Business Process Instances, and view the Business Process Instances and their associated Business Process Instances in various ways.

10.2 Monitoring New Business Processes

The following procedure provides the steps for monitoring a Business Process in Enterprise Manager.

To monitor a new Business Process

1. Open and save each Business Process.

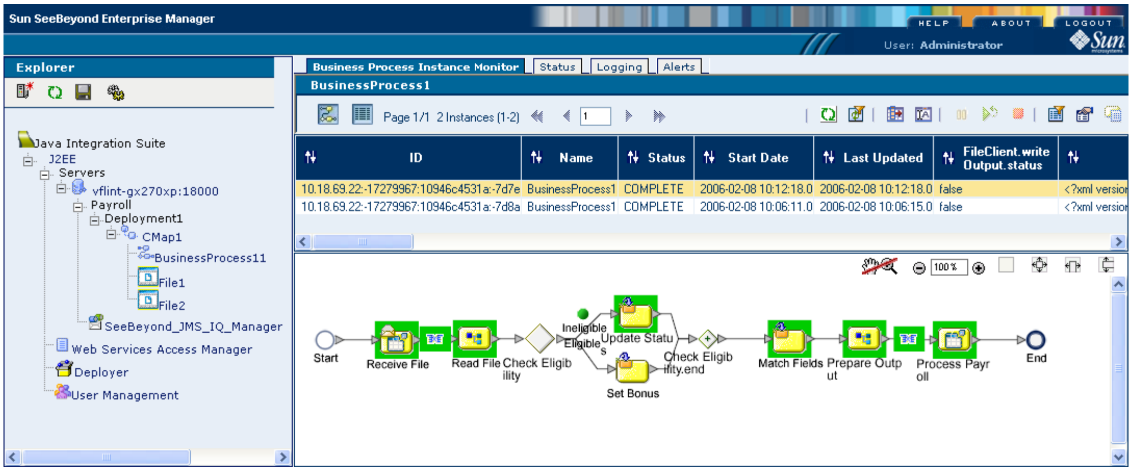
Note: *If a Business Process has User Activities or special OTDs, open and close each of them to ensure that they register with the monitor.*

2. Check in each Business Process.
3. Check out each Business Process.
4. Run the database scripts again for each Business Process.
5. Save each Business Process.
6. Activate the Deployment Profile, rebuild, and redeploy the Project.
7. In your web browser, connect to Enterprise Manager and log in.

Note: *For detailed information about logging into Enterprise Manager, see the Sun SeeBeyond eGate Integrator System Administration Guide.*

8. In the Enterprise Manager Explorer, navigate to the correct Server/Project/Deployment Profile/Connectivity Map, and click the Business Process name.
9. In the Business Process Instance Monitor tab, click the Show/Hide list of Business Process Instances button. This button is located next to the Show/Hide Business Process Model button at the upper left of the Business Process Instance Monitor panel.

Figure 96 Business Process Instance Monitor Tab



ID	Name	Status	Start Date	Last Updated	FileClient.write Output.status	
10.18.69.22:-17279967:10946c4531a:-7d7e	BusinessProcess1	COMPLETE	2006-02-08 10:12:18.0	2006-02-08 10:12:18.0	false	<?xml version
10.18.69.22:-17279967:10946c4531a:-7d8a	BusinessProcess1	COMPLETE	2006-02-08 10:06:11.0	2006-02-08 10:06:15.0	false	<?xml version

10.3 Monitoring Modified Business Processes

The following procedure provides the steps for monitoring a modified Business Process in Enterprise Manager.

To monitor a modified Business Process

1. If necessary, check out the Business Process.
2. Modify the Business Process as necessary.
3. Save all changes.
4. Check in the Business Process.
5. Check out the Business Process.
6. Run the database script again.
7. Activate the Deployment Profile, rebuild, and redeploy the Project.
8. In your web browser, connect to Enterprise Manager and log in.

Note: *For detailed information about logging into Enterprise Manager, see the Sun SeeBeyond eGate Integrator System Administration Guide.*

9. In the Enterprise Manager Explorer, navigate to the correct Server/Project/Deployment Profile/Connectivity Map, and click the Business Process name.
10. In the Business Process Instance Monitor tab, click the Show/Hide list of Business Process Instances button. This button is located next to the Show/Hide Business Process Model button at the upper left of the Business Process Instance Monitor panel.

10.4 Monitoring a Business Process in an Imported Project

The following procedure provides the steps for monitoring a Business Process in an imported Project.

To monitor a Business Process in an imported project

1. Import the project into Enterprise Designer.
2. Check out any Business Processes.
3. Open and save each Business Process.

Note: *If a Business Process has User Activities or special OTDs, open and close each of them to ensure that they register with the monitor.*

4. Check in each Business Process.
5. Check out each Business Process.
6. Run the database script again.
7. Activate the Deployment Profile, rebuild, and redeploy the Project.
8. In your web browser, connect to Enterprise Manager and log in.

Note: *For detailed information about logging into Enterprise Manager, see the Sun SeeBeyond eGate Integrator System Administration Guide.*

9. In the Enterprise Manager Explorer, navigate to the correct Server/Project/Deployment Profile/Connectivity Map, and click the Business Process name.
10. In the Business Process Instance Monitor tab, click the Show/Hide list of Business Process Instances button. This button is located next to the Show/Hide Business Process Model button at the upper left of the Business Process Instance Monitor panel.

10.5 Controlling and Evaluating Business Process Instances

The Business Process Instance Monitor tab provides a set of toolbars for viewing and evaluating Business Process Instances. Before you start monitoring Business Process Instances, be sure to acquaint yourself with these tools.

10.5.1 Displaying Instances and Lists

The buttons at the upper left of the Business Process Instance Monitor tab provide the graphic display of Business Process Instances as well as Business Process Instance Lists. The following table describes each button.

Table 24 Business Process Instance Monitor Tab: Display Buttons

	Show Business Process Model renders the image of a Business Process Instance in the **Details** window.
	Hide Business Process Model hides the rendered image of a Business Process Instance in the **Details** window.
▤	**Show List of Business Process Instances** displays the attributes of the current Business Process Instance in list format, and adds the tools described in the following table.
▤	**Hide List of Business Process Instances** hides the attributes of the current Business Process Instances, and removes the instance tools from the interface.

10.5.2 Controlling the Display of Business Process Instances

When Show Business Process Model is selected, you can manipulate the view of Business Processes using the buttons described in the following table. These buttons are located at the upper right of the Show Business Process Model panel.

Table 25 Toolbar: Show Business Process Model Button

!	**Alert to Enable Monitoring** alerts you to enable monitoring in the eInsight Engine Configuration Properties in order to see Business Process Activity status and details in the eInsight Business Process Instance Monitor.
	Enable Zoom and Pan enables zooming and panning of the Business Process Instance view.
	Disable Zoom and Pan disables zooming and panning of the Business Process Instance view.
⊖	**Zoom Out** zooms out to a lower percentage view of the Business Process Instance.
⊕	**Zoom In** zooms in to a higher percentage view of the Business Process Instance.
□	**100%** changes the percentage of the view to 100%.
✥	**Fit All** changes the percentage of the view so that the entire Business Process Instance fits in the Show Business Process Model panel.

Table 25 Toolbar: Show Business Process Model Button

↔	**Fit Width** changes the percentage of the view so that the width of the Business Process Instance fits in the Show Business Process Model panel.
↕	**Fit Height** changes the percentage of the view so that the height of the Business Process Instance fits in the Show Business Process Model panel.

10.5.3 Controlling the Display of Business Process Instance Data

When Show List of Business Process Instances is selected, you can manipulate the view of Business Process Instance data using the buttons described in the following table. These buttons are located at the upper right of the Show list of Business Process Instances panel.

Table 26 Toolbar: Show List of Business Process Instances Button

↻	**Refresh list of Business Process Instances** refreshes the entire Business Process Instance list.
	Refresh Filtered list of Business Process Instances refreshes a filtered list of Business Process Instances.
	Choose Preferences allows you to add, move, and sort the columns in the Business Process Instance.
IA	**Change Attribute Display Names** allows you to change the display name of all Business Process Instance attributes.
⏸	**Suspend** suspends a Business Process Instance.
▶	**Start** starts a stopped Business Process Instance.
■	**Stop** stops a Business Process Instance.
	Filter Business Process Instances allows you to set criteria to display a specific instance or group of Instances.
	Business Process Instance Attributes displays the XML content of each Business Process Instance attribute.

Table 26 Toolbar: Show List of Business Process Instances Button

	Activity Details displays a dialog box with the details of a selected Activity, including the following: ▪ Start Time ▪ End Time ▪ Status ▪ Message input/output (XML viewer) ▪ Exception content ▪ Each activity represents a step within the Business Process.
	Go To Caller takes you to calling Business Process Instance such as a parent Business Process.
	Go To Callee takes you to a called Business Process Instance such as a child Business Process.
	Legend displays a window with a legend of the colors used to show the status of a Business Process Instance. The Legend displays the following Business Process states: ▪ Completed ▪ In Progress ▪ Failed ▪ Unknown

10.5.4 Choosing Business Process Attributes to Display

In the list of Business Process Instances, you can make the following choices.

- Columns to show
- Columns to hide
- Column order
- Maximum rows per page
- Instance list refresh rate
- Total number of Business Process Instances allowed

The following procedure provides the steps for choosing Business Process attributes to display in the list of Business Process Instances.

To choose Business Process attributes to display

1. In the toolbar of the list of Business Process Instances, click the **Choose Preferences** button.

   The Choose Business Process Attributes to Display dialog box appears.

2. In the Chosen Columns list, select the columns that you want to hide from the list of Business Process Instances.

3. Click the leftward double arrow to move the selected columns to the Available Columns list.

4. In the Maximum Rows Per Page field, enter the number of rows you want to display on each page of the list of Business Process Instances. The default is 10.
5. In the Instance List Refresh Rate field, enter the number of seconds you want to pass between refreshes of the list of Business Process Instances. The default is 120.
6. In the Number of BP Instances field, enter the maximum number of Business Process Instances you want to monitor. The default is 500.
7. Click the Change Preferences button.

   The dialog box closes.

10.5.5 Changing the Display Name of an Attribute

If you prefer to monitor Business Process Instance attributes using shortened names, you can change the display names of the attributes you want to include in the list of Business Process Instances. The following procedure provides the steps for changing the display names of Business Process Instance attributes.

To change the display name of an attribute

1. In the toolbar of the list of Business Process Instances, click the Change Attribute Display Names button.

   The Change Attribute Display Name dialog box appears.
2. In an attribute display name field, edit the text of the attribute display name.
3. Continue editing these text fields as necessary.
4. Click the Submit button.

   If you want to return the attribute display names to their default settings, click the Change Attribute Display Names button again and click the Reset button at the bottom of the dialog box.

10.5.6 Filtering Business Process Instances

You can filter the list of Business Process Instances in order to see only Business Process Instances that meet a specific set of criteria. The Filter Business Process Instance Dialog Box provides the following options.

- Business Process Instance Status
- Start date range
- Update date range
- Business Process attribute

To filter the list of Business Process Instances

1. In the toolbar of the list of Business Process Instances, click the Filter Business Process Instances button.

   The Filter Business Process Instances dialog box appears.
2. In the Status drop-down list, select a Business Process status.

3 In the Time Stamp drop-down list, select a time stamp type.

4 For the From field, click the Select Date and/or Time button and select the date and time.

5 For the To field, click the Select Date and/or Time button and select the date and time.

6 In the Business Process Attribute drop-down list, select the attribute and filter criteria operator and text.

7 Click the Filter button.

10.5.7 Viewing the Content of a Business Process Instance Attribute

If you want to view the XML content of all Business Process Instance attributes, you can click the Business Process Instance Attributes button. A dialog box appears that lists each attribute and its XML content. You can also click the View XML button to see the XML in a structured XML viewer.

10.6 Monitoring Load-Balanced Business Process Instances

When you are monitoring load-balanced Business Process Instances in Recovery mode, single Business Process Instances appear to be mulitple Business Process Instances on multiple eInsight Engines. This is a normal result of load balancing a Business Process across multiple eInsight Engines.

10.7 Using Enterprise Manager's Administrative Tabs

In addition to monitoring Business Processes with Enterprise Manager, you can also check component status and manage application server logging and alerting. For detailed information about using Enterprise Manager's Status, Logging, and Alerts tabs, see the *Sun SeeBeyond eGate Integrator System Administration Guide*.

Chapter 11

eXchange Troubleshooting Tips

This chapter provides troubleshooting and tuning information for eXchange Integrator.

What's in This Chapter

11.1 Identifying and Resolving Problems

The **Readme.txt** file contains a list of known issues for eXchange and other products. It can be browsed directly on the installation media, or via the DOCUMENTATION tab of Enterprise Manager (i.e., *<jc512>*\Repository\server\webapps\ICANDocs\).

For help resolving some commonly encountered difficulties, see Table 27.

Table 27 Troubleshooting: Symptoms, Causes, and Remedies

Symptom	Possible Causes	Suggested Remedies
In Enterprise Manager, while uploading, receive message "installation failed"; and/or log says "Could not install: *<product>*"	One or more prerequisite **.sar** files have not been installed.	Ensure you are licensed to install the product. Install its prerequisite **.sar** files (see **"Before You Install" on page 30**) one by one and then upload the product again.
During installation, the console displays this text: [InstallJavaCollab] Error is: ...ClassNotFoundException: ...NBSystemResourceLoader	This is normal behavior.	If the upload finishes successfully, ignore this error message; there is no problem.
Messages in log files or on the console are too sparse or too abundant.	The level of message output in the log4j.properties file is not set appropriately.	Depending on the module that is too terse or too verbose, edit the file in one or more of these *<jc512>* locations: ▪ edesigner\bin\log4j.properties ▪ ESRs\log4j.properties ▪ monitor\config\log4j.properties ▪ Repository\server\conf\log4j.properties

Table 27 Troubleshooting: Symptoms, Causes, and Remedies (Continued)

Symptom	Possible Causes	Suggested Remedies
In the Deployment Editor, cannot drop an Oracle eWay onto the Oracle external; or, upon Automap, the Oracle eWay is not assigned to any external.	▪ The eWay is Inbound but the external is Outbound, or vice versa. ▪ The Oracle host is not configured to match a running Oracle service. ▪ The Oracle service is not running.	▪ Reconfigure the eWay (in the connectivity map) or the Oracle host (in the environment editor). ▪ In the environment editor, double-check the properties of the Oracle external system. ▪ Start the Oracle service (e.g., using **Control Panel >Admin... > Services**).
Upon running bootstrap, receive message “An instance of bootstrap is already running in this directory.”	A previous bootstrap was shut down improperly.	First, try running the **shutdown** script with the **-c** flag. If the problem continues to persist after this, find and remove the **bootstrap.lock** file.
Any of the following runtime messages are received: ▪ Error while decrypting. ▪ RSA_ not supported ▪ Error while encrypting in PKCS7 format. ▪ java.lang.SecurityException	Incorrect versions of the **US_export_policy.jar** and **local_policy.jar** files are in use.	See **http://java.sun.com** for instructions on how to obtain the correct versions. If you have installed the correct versions and still get this message, then run **shutdown**; delete all subdirectories under *<logicalhost>*, except **bootstrap** and **jre**; and then re-bootstrap to regenerate the keystores.
Changes made to the design, environment, or trading partner profile are not picked up at run time. For example: ▪ A component is added, but ignored. ▪ A configuration error is corrected, but the runtime behavior is unchanged. ▪ A keystore is changed, but the new password is not recognized.	Stale information is being retained by the Repository or Logical Host or eXchange database.	▪ Design changes: Double-check the connections between components; save all changes and check in all components; exit all canvases; refresh all from Repository; reopen and reactivate the deployment profile. ▪ Environment changes: Either right-click the Logical Host and click Apply, or select the Apply Changes checkbox when reactivating the deployment profile, or use the **-f** flag when rebootstrapping. ▪ TP profile changes: In ePM, go through the TP’s parameters and save each one; then reactivate the trading partner.
Message Tracking reports “No messages found”.	▪ The statement is accurate, and no messages have been processed. ▪ The wrong host is specified and/or the filters are too stringent.	▪ Look inside the eXchange database: If the EX_MSG_CONTENT table has no messages, the statement is accurate; ensure that the ChannelManager **track** operation is being used correctly. ▪ Be sure the correct host is selected, and set the Search parameters to **ALL** for both Trading Partner and Protocol.

Table 27 Troubleshooting: Symptoms, Causes, and Remedies (Continued)

Symptom	Possible Causes	Suggested Remedies
The Business Process starts, and ChannelManager retrieves the delivery channel profile, but nothing happens.	▪ There is no input being staged. ▪ There is no binding for the internal delivery channel to the location where input exists.	▪ Double-check the values for input directory and file mask (and, for FTP, hostname and login information). Ensure that input data exists. ▪ In the TP profile, in Messaging Service Configuration, set the internal delivery channel for each messaging action so that it points at the correct input data location. Then, if necessary, reactivate.
Errors occur upon trying to bootstrap the Logical Host; log file text reads as follows: "DCPAccessException: No Host with Network Id ... at ...DCPDBViewV2.lookup"	You are trying to run a project that performs a TP lookup, but no trading partner profile has been activated in ePM.	Create and activate one or more trading partner profiles. See **"Creating and Configuring Transaction Profiles" on page 121**.
Unexpected errors occur at runtime, with the following error message text: "java.sql.SQLException: ORA-01000: maximum open cursors exceeded"	The load on the eXchange Oracle database is too great for the default settings to handle.	In the **init.ora** file for the eXchange database, increase the value for the **open_cursors** parameter to 500. See **"Modifying the init.ora File for the eXchange Database" on page 38**.
Runtime exceptions are seen in the error log, such as a **runtime.BPELInterpreter** exception with this text: "BPELException: Max Instances Exceeded and incoming request timeout for Business Process: *<path>/<GUID>*" — or — Messages that are submitted simultaneously are picked up but not processed.	The load on the eXchange Oracle database, the Logical Host, and/or the eInsight engine is too great for the default settings to handle.	See **"Performance Considerations" on page 172**. For example, when dealing with 100-200 concurrent messages, the following settings might be adjusted. For Oracle: ▪ Change Oracle number of processes to 500. For the Logical Host: ▪ Change heap size from 512 to 1024 (or higher, if the machine has sufficient memory). For LH > IS > eInsight Engine (see **"Configuring the eInsight Engine for Runtime" on page 155**): ▪ Increase the value for Max Concurrent Instances from empty (default) to 30. ▪ Increase the value for Max Runtime Thread Pool from 15 to 90. ▪ Increase the value for Receive Timeout and Max Concurrent Instances thus: RT > (N – MCI) *BPE where RT = Receive Timeout value N = max number of input messages MCI =Max Concurrent Instances value BPE = maximum BP execution time

11.2 Performance Considerations

For Java CAPS Projects that process large messages and transactions, or large amounts of transactions, you can configure the Project in the following ways to increase throughput and improve performance:

- **Increasing the Logical Host Heap Size** on page 172
- **Increasing eInsight Engine Performance** on page 172
- **Increasing the JMS IQ Manager Segment Size Property** on page 174
- **Increasing the Oracle Number of Processes** on page 174

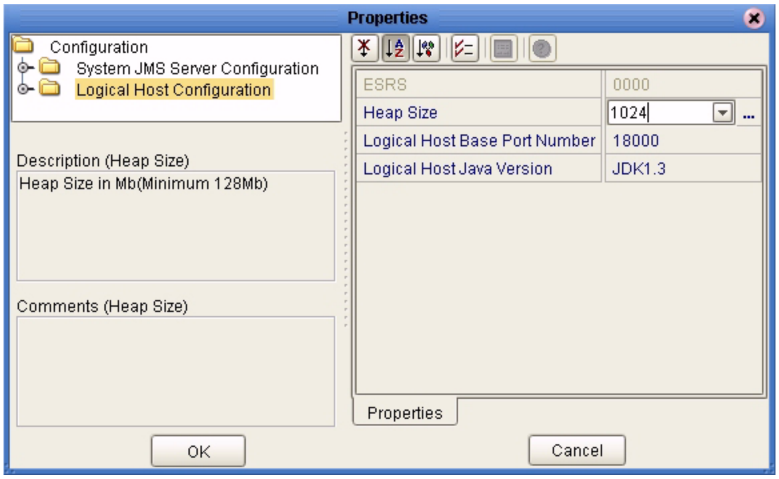
11.2.1 Increasing the Logical Host Heap Size

To avoid memory errors on the Logical Host, increase the heap size for the Logical Host to at least 1024 MB as described below.

To increase the Logical Host heap size

1. In the **Environment Explorer** tab in Enterprise Designer, right-click the Logical Host, and click **Properties**. The **Properties** dialog box appears.
2. In the **Heap Size** box, enter the new heap size.

Figure 97 Increasing Logical Host Heap Size



Description (Heap Size)
Heap Size in Mb(Minimum 128Mb)

Comments (Heap Size)

ESRS	0000
Heap Size	1024
Logical Host Base Port Number	18000
Logical Host Java Version	JDK1.3

3. Click **OK**.

11.2.2 Increasing eInsight Engine Performance

The eInsight engine has three properties that affect performance:

- **Max Concurrent Instances**

- **Max Runtime Thread Pool Size**

The **Max Concurrent Instances** property indicates the amount of inbound messages that each BP processes simultaneously. For example, if you set this property to 10, the BP processes up to ten inbound messages at the same time. When the eleventh message is received, the BP does not process the message until the processing of one of the ten messages has been completed.

The **Max Runtime Thread Pool Size** property defines how many threads BPs can invoke simultaneously.

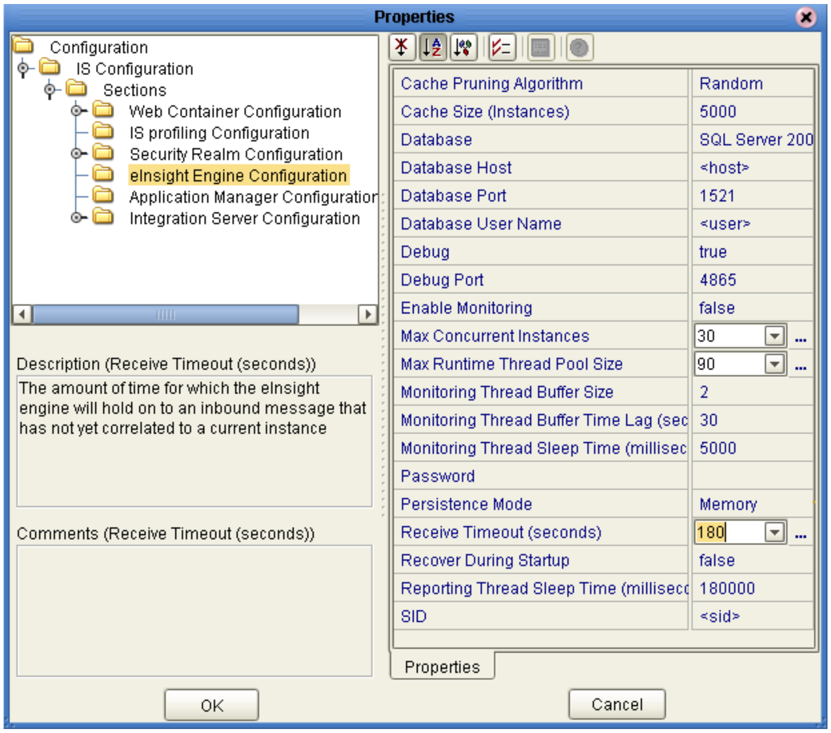
These settings are a trade-off between performance and memory. Depending on how much memory your system has, you can specify these settings accordingly. Unfortunately it is impossible to recommend actual settings here because the memory capacity of your system is unknown. If you find your Project processes too slowly, or messages are lost, increase these settings as your system allows.

For instructions on configuring runtime properties of the eInsight engine, see **“Configuring the eInsight Engine for Runtime” on page 155**. For more information about these properties, refer to the *eInsight Business Process Manager User’s Guide*.

To increase eInsight Engine Performance

1. In the **Environment Explorer** tab of the Enterprise Designer, expand the Logical Host folder, right-click the Integration Server, and click **Properties**. The **Properties** dialog box appears.
2. Expand the **IS Configuration**, **Sections**, and **eInsight Engine Configuration** folders.
3. Increase the settings for the **Max Concurrent Instances** and **Max Runtime Thread Pool Size** properties.

Figure 98 Increasing eInsight Engine Performance


Properties
Configuration
IS Configuration
Sections
Web Container Configuration
IS profiling Configuration
Security Realm Configuration
eInsight Engine Configuration
Application Manager Configuratior
Integration Server Configuration
Properties
OK
Cancel

Description (Receive Timeout (seconds))
The amount of time for which the eInsight engine will hold on to an inbound message that has not yet correlated to a current instance

Comments (Receive Timeout (seconds))

Cache Pruning Algorithm	Random
Cache Size (Instances)	5000
Database	SQL Server 200
Database Host	<host>
Database Port	1521
Database User Name	<user>
Debug	true
Debug Port	4865
Enable Monitoring	false
Max Concurrent Instances	30
Max Runtime Thread Pool Size	90
Monitoring Thread Buffer Size	2
Monitoring Thread Buffer Time Lag (sec	30
Monitoring Thread Sleep Time (millisec	5000
Password	
Persistence Mode	Memory
Receive Timeout (seconds)	180
Recover During Startup	false
Reporting Thread Sleep Time (millisecc	180000
SID	<sid>

4 Click **OK**.

11.2.3 Increasing the JMS IQ Manager Segment Size Property

If a Project that uses the SeeBeyond JMS IQ Manager will need to process messages or transactions over 8 MB (Windows) or 16 MB (UNIX), increase the **Segment size** property of the JMS IQ Manager as described in the *eGate Integrator JMS Reference Guide*.

11.2.4 Increasing the Oracle Number of Processes

To increase throughput performance, increase the Oracle number of processes to handle simultaneous database process requests, such as 500. For information, refer to your Oracle documentation.

Glossary of Acronyms

AD, *AD, *x*AD

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

AS2

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

B2B

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

BAD

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

BPAD

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

CAPS

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

CIDX

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

DPAD

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

EAD

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

ebXML

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

ePM

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

FTP

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

HTTP

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

ICAN

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

LDAP

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

MAD

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

MIME

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

OTD

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

PAD

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

RNIF

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

SME

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

S/MIME

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

SMTP

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

TAD

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

TCP/IP

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

TP, TPP

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

URL

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

XML

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

Appendix A

Obtaining and Setting Up an LDAP Server

This chapter provides instructions for installing and configure the Sun Java™ System Directory Server 5.2 (formerly Sun ONE Directory Server 5.2). It also has information on checking LDAP services and browsing the Directory Server structure.

What's in This Chapter

A.1 Downloading and Installing the LDAP Server

This section is for readers who have never installed an LDAP server, or who want to refresh their memory as to the default installation and configuration options.

To download Sun Java System Directory Server 5.2 (Sun ONE Directory Server 5.2)

1. Point your browser at the following URL:

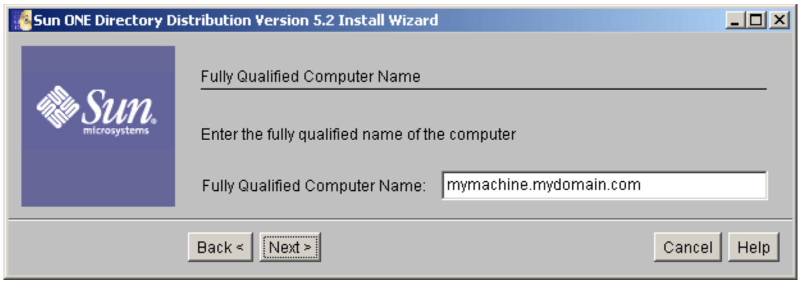
   `http://developers.sun.com/prodtech/dirserver/downloads/index.html`

2. Click the link“Sun Java System Directory Server Free Trial Download“.
3. Double-check that your language and platform are supported, and then click the “Download” link. Registration may be required.
4. In the Sun Downloads page, after accepting the license agreement, locate the appropriate file (such as **d52diu.zip** for Windows - English) and download it.
5. Extract its contents to a local directory.

To install the LDAP Server

1. Change directories to the location where you extracted in the previous procedure, and then run the **setup** script.
2. In the Sun ONE Directory Distribution Version 5.2 Install Wizard, click Next.
3. Click Yes to accept the license agreement.

4 Enter the fully qualified name of the machine where you are installing the software, and then click Next:



Fully Qualified Computer Name

Enter the fully qualified name of the computer

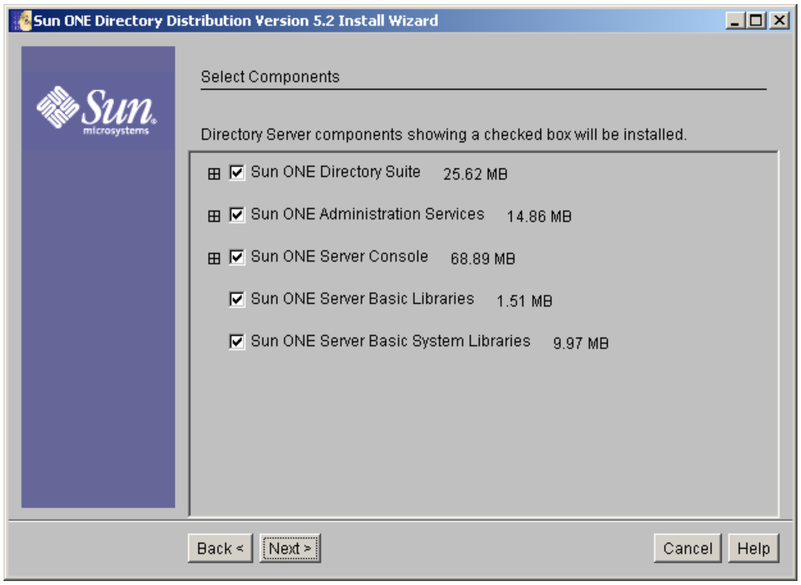
Fully Qualified Computer Name: mymachine.mydomain.com

5 Retain the default option (to install the **servers** and related components, not just the console as a standalone application) and then click Next.

6 Retain the default option (**typical**, not express or custom) and then click Next.

7 Select the directory where you want to install—use C:\Program Files\Sun\MPS\ (creating it if necessary)—and then click Next.

8 For **Server Components**: Retain the selection of all five components and click Next:

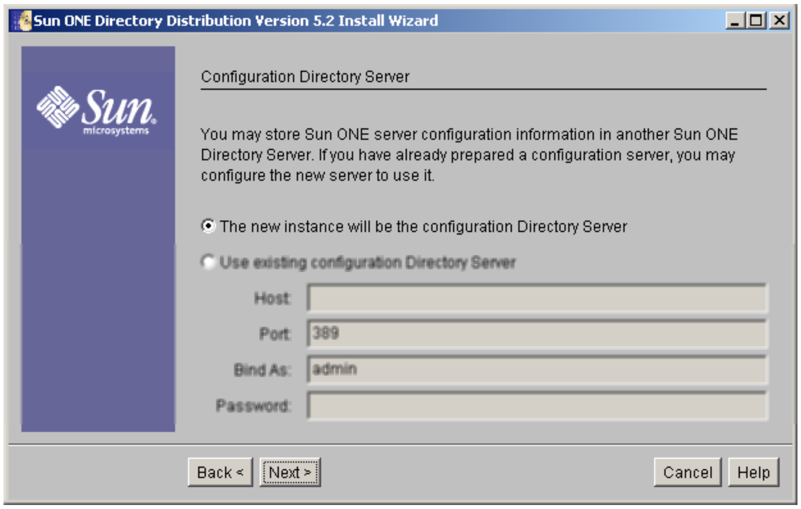


Select Components

Directory Server components showing a checked box will be installed.

⊞ ☑ Sun ONE Directory Suite 25.62 MB
⊞ ☑ Sun ONE Administration Services 14.86 MB
⊞ ☑ Sun ONE Server Console 68.89 MB
☑ Sun ONE Server Basic Libraries 1.51 MB
☑ Sun ONE Server Basic System Libraries 9.97 MB

9 For **Configuration Directory Server**: Retain the default option (to use the **new instance**, not an existing one) and click Next:



Configuration Directory Server

You may store Sun ONE server configuration information in another Sun ONE Directory Server. If you have already prepared a configuration server, you may configure the new server to use it.

● The new instance will be the configuration Directory Server

○ Use existing configuration Directory Server

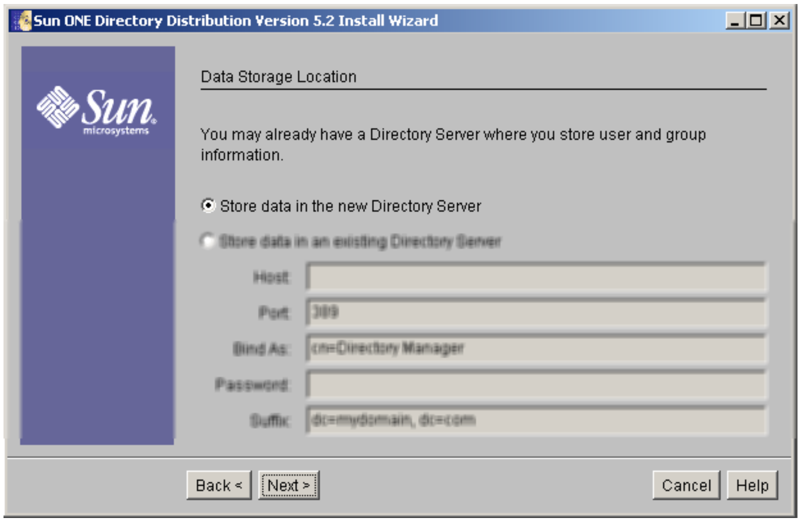
Host:

Port: 389

Bind As: admin

Password:

10 For **Data Storage Location**: Retain the default option (to store in the **new server**, not a pre-existing one) and click Next:



Data Storage Location

You may already have a Directory Server where you store user and group information.

● Store data in the new Directory Server

○ Store data in an existing Directory Server

Host:

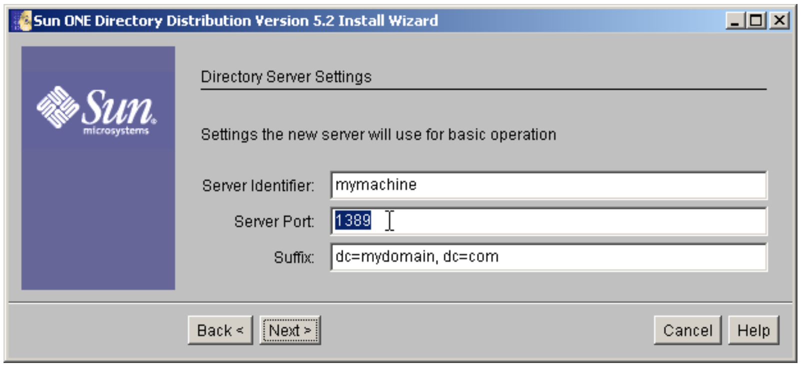
Port: 389

Bind As: cn=Directory Manager

Password:

Suffix: dc=mydomain, dc=com

11 For **Directory Server Settings**: Retain or provide values for server identifier (hostname), port, and suffix (domain components) as needed and click Next:



Directory Server Settings

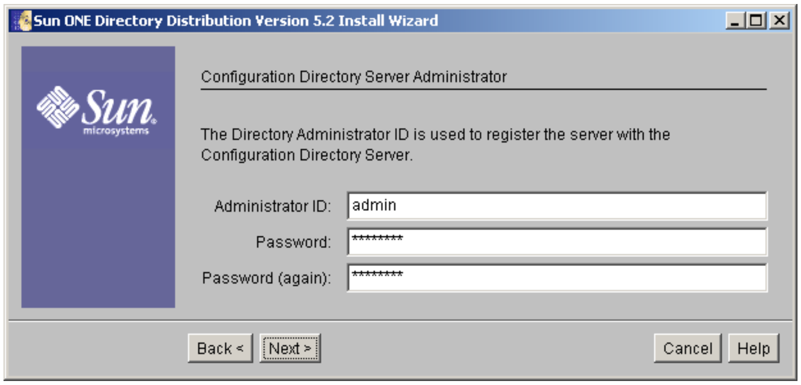
Settings the new server will use for basic operation

Server Identifier: mymachine

Server Port: 1389

Suffix: dc=mydomain, dc=com

12 For **Configuration Directory Server Administrator**: Provide an ID (such as **admin**) and password (such as **password**) of your choosing and then click Next:



Configuration Directory Server Administrator

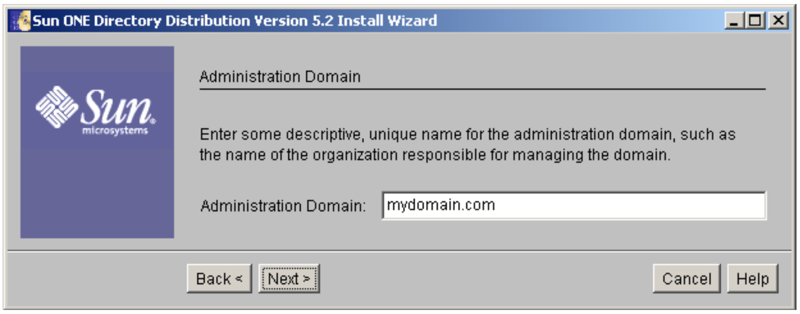
The Directory Administrator ID is used to register the server with the Configuration Directory Server.

Administrator ID: admin

Password: *********

Password (again): *********

13 For **Administration Domain**: Retain the domain name and click Next:

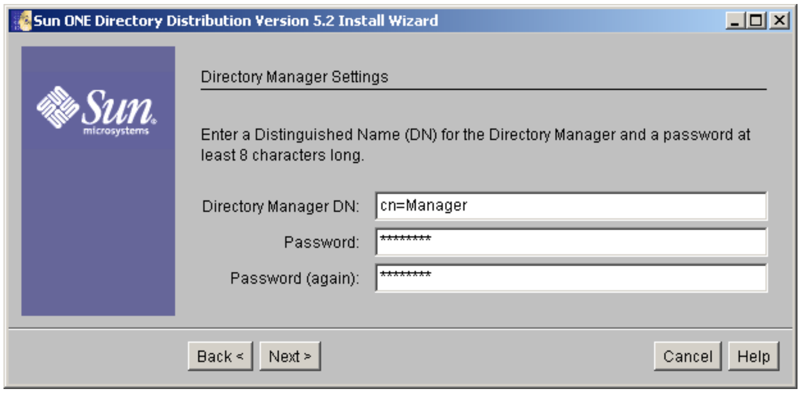


Administration Domain

Enter some descriptive, unique name for the administration domain, such as the name of the organization responsible for managing the domain.

Administration Domain: mydomain.com

14 For **Directory Manager Settings**: Provide a DN (such as **cn=Manager**) and password (such as **exchange**) of your choosing and then click Next:



Directory Manager Settings

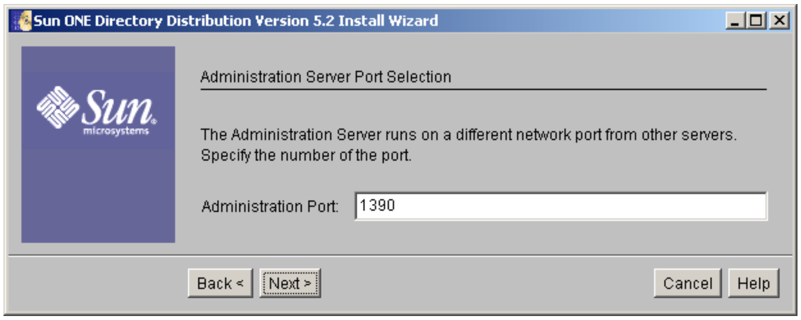
Enter a Distinguished Name (DN) for the Directory Manager and a password at least 8 characters long.

Directory Manager DN: cn=Manager

Password: *********

Password (again): *********

15 For **Administration Server Port Selection**: Retain the default value and click Next:



Sun ONE Directory Distribution Version 5.2 Install Wizard

Administration Server Port Selection

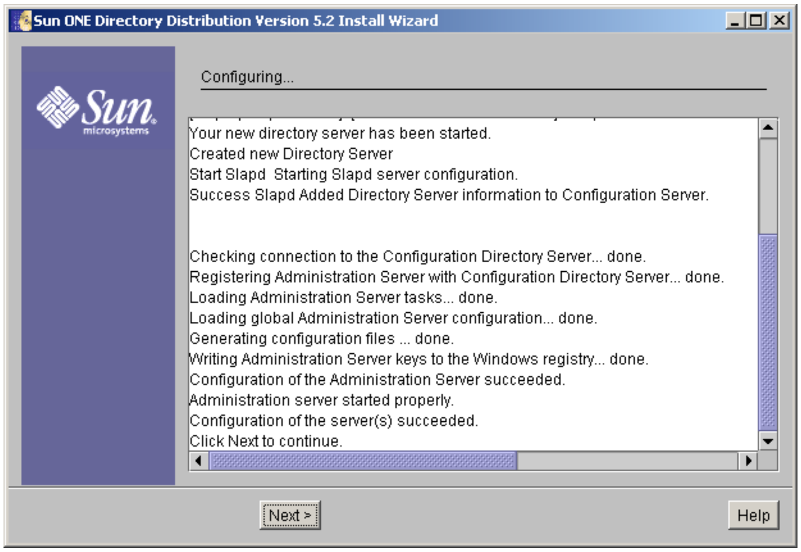
The Administration Server runs on a different network port from other servers. Specify the number of the port.

Administration Port: 1390

16 Review the summary of items to be installed and then click **Install Now**.

After taking some time to install and configure, the install wizard confirms success.

17 For **Configuring**: Scroll to the end of the messages and then click Next:



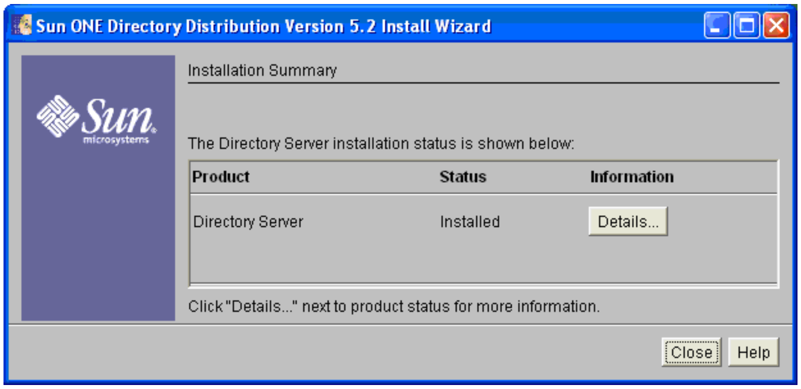
Sun ONE Directory Distribution Version 5.2 Install Wizard

Configuring...

```
Your new directory server has been started.
Created new Directory Server
Start Slapd  Starting Slapd server configuration.
Success Slapd Added Directory Server information to Configuration Server.


Checking connection to the Configuration Directory Server... done.
Registering Administration Server with Configuration Directory Server... done.
Loading Administration Server tasks... done.
Loading global Administration Server configuration... done.
Generating configuration files ... done.
Writing Administration Server keys to the Windows registry... done.
Configuration of the Administration Server succeeded.
Administration server started properly.
Configuration of the server(s) succeeded.
Click Next to continue.
```

18 For **Installation Summary**, the final step of the wizard, click Close:



Sun ONE Directory Distribution Version 5.2 Install Wizard

Installation Summary

The Directory Server installation status is shown below:

Product	Status	Information
Directory Server	Installed	Details...

Click "Details..." next to product status for more information.

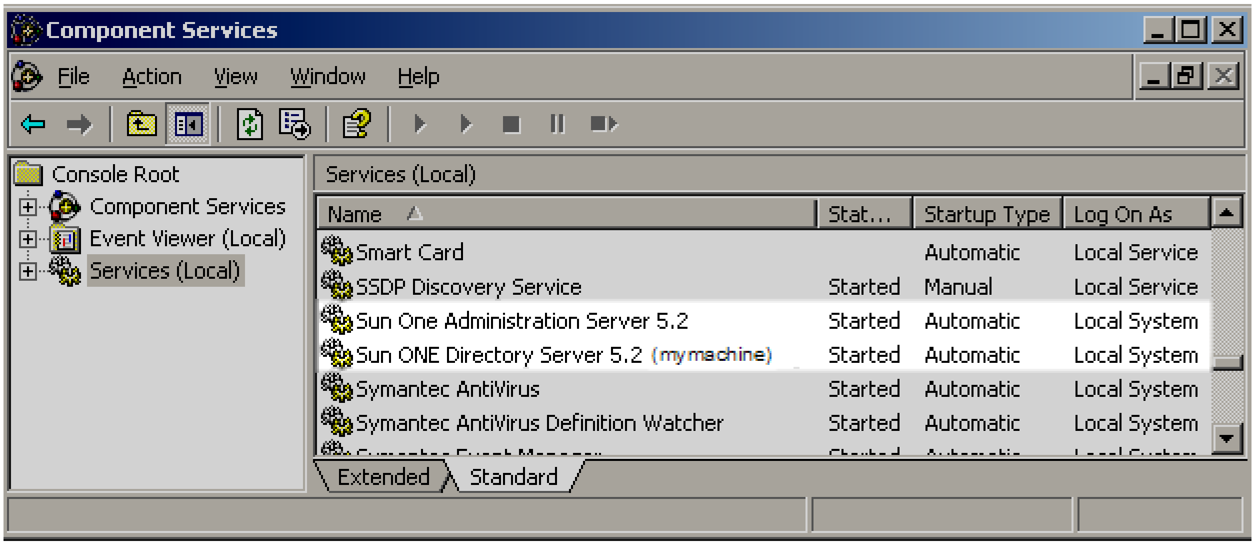
A.2 Verifying and Working With the LDAP Services

After you install the Sun Java Directory Server, two services are added: Administration Server 5.2 and Sun ONE Directory Server 5.2 for <*machinename*>.

To verify LDAP services and their status

- Click Control Panel > Administrative Tools > Services (Local). See Figure 99.

Figure 99 LDAP Services After Installation



Name	Stat...	Startup Type	Log On As
Smart Card		Automatic	Local Service
SSDP Discovery Service	Started	Manual	Local Service
Sun One Administration Server 5.2	Started	Automatic	Local System
Sun ONE Directory Server 5.2 (mymachine)	Started	Automatic	Local System
Symantec AntiVirus	Started	Automatic	Local System
Symantec AntiVirus Definition Watcher	Started	Automatic	Local System
Symantec Event Manager	Started	Automatic	Local System

To start, stop, pause, resume, or configure a service

- In the list of services, double-click the service you want to work with:
  - You can use the **General** tab to start, stop, pause, or resume operation.
  - You can use the **Log On** tab to associate accounts and hardware profiles.
  - You can use the **Recovery** tab to specify responses if the service fails.
  - You can use the **Dependencies** tab to see other services that are affecting or affected by the LDAP service.

A.3 Installing and Using the LDAP Browser/Editor

A freeware LDAP browser/editor (LBE) can be used to connect to the Directory Server and view or modify its contents.

Note: *Before you try to browse a newly created Sun Java Directory Server, be sure to replaced its ...\config\schema\28pilot.ldif with the three eXchange-supplied LDIF files (see* **procedure on page 37***); otherwise, the LDAP browser/editor will have nothing to browse.*

To download and install the free LDAP browser/editor

1 Point your browser at the following URL:

```
http://www-unix.mcs.anl.gov/~gawor/ldap/download.html
```

2 Click the link "Browser282b2.zip" and download the file to a local folder.

3 Extract the contents of Browser282b2.zip, preserving folder structure.

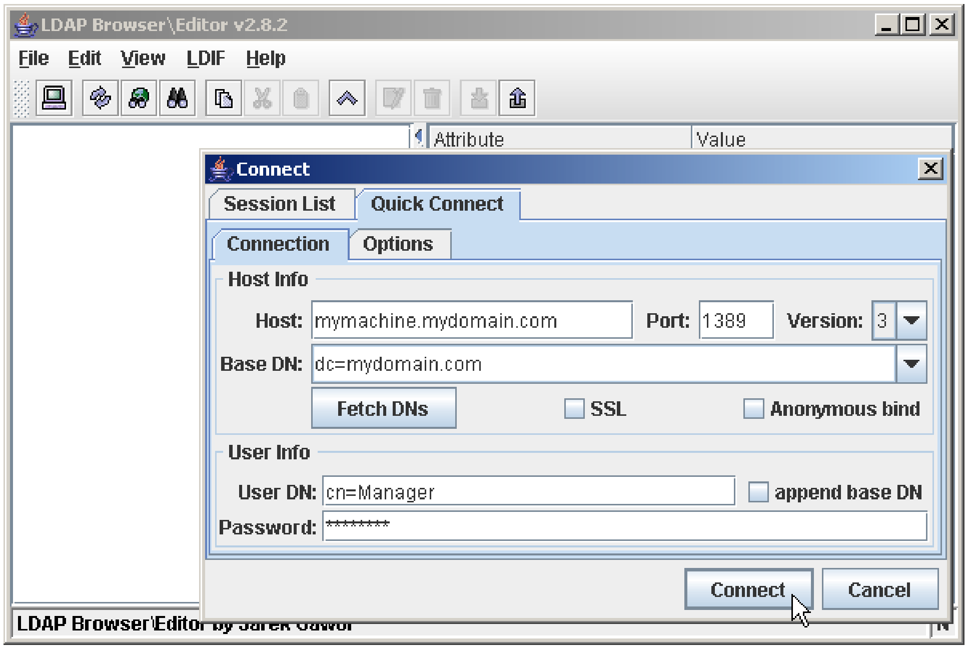
A new subfolder is created: **ldapbrowser**

To establish a new connection and view LDAP entries

1 Start the LBE using the ...\ldapbrowser**lbe.bat** script.

2 Establish a connection to the LDAP server; see Figure 100. (Alternatively, you can establish a connection using the menu: **File > Connect**.)

Figure 100 Establishing an LBE Connection to the LDAP Server



Host: mymachine.mydomain.com Port: 1389 Version: 3

Base DN: dc=mydomain.com

☐ SSL ☐ Anonymous bind

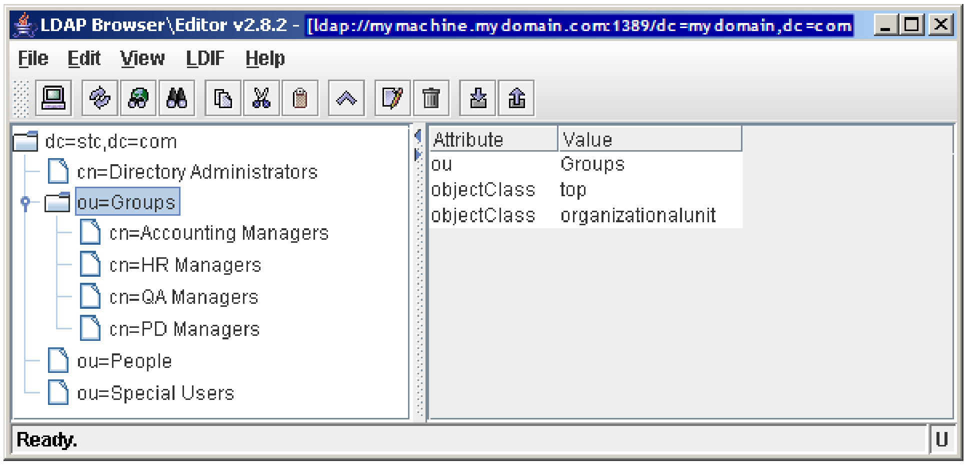
User DN: cn=Manager ☐ append base DN

Password: ********

If the connection fails, double-check be sure that you have replaced 28pilot.ldif with the eXchange LDIF files and that you have stopped and restarted both services.

3 Browse entries in the LDAP hierarchy; see Figure 101.

Figure 101 Browsing Entries in the LDAP Server


LDAP Browser\Editor v2.8.2 - [ldap://mymachine.mydomain.com:1389/dc=mydomain,dc=com
File
Edit
View
LDIF
Help
U

- dc=stc,dc=com
  - cn=Directory Administrators
  - ou=Groups
    - cn=Accounting Managers
    - cn=HR Managers
    - cn=QA Managers
    - cn=PD Managers
  - ou=People
  - ou=Special Users

Attribute	Value
ou	Groups
objectClass	top
objectClass	organizationalunit

Ready.

To find and modify LDAP entries

1 In LBE, to find subordinate entries such as organizations and organizational units, you can either use the View > Search facility or you can drill down as needed from the root level of the tree. See Table 28.

Table 28 Names for LDAP Entries

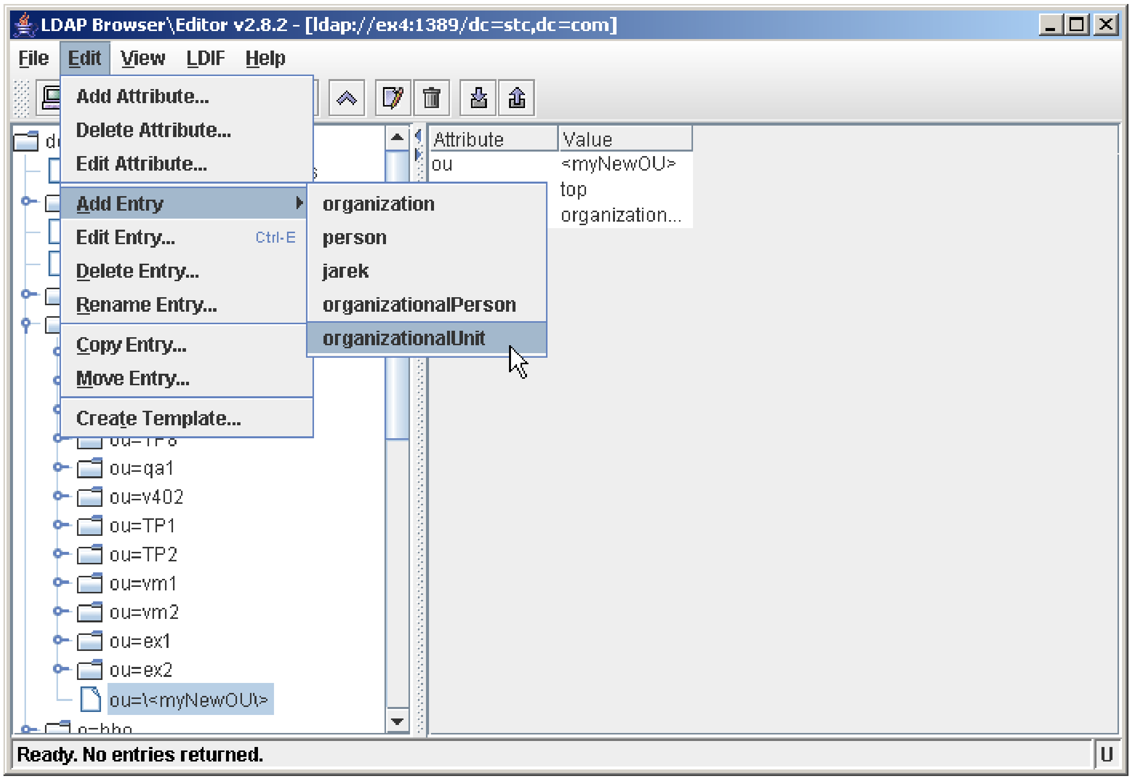
Entity Name	Meaning
dn	distinguished name
dc	domain components
cn	common name
o	organization
ou	organizational unit

2 To create a new o or ou immediately subordinate to another entry:

A First, click the parent entry, highlighting it. For example, if you wanted to create a new entry subordinate to the domain component itself, you would start by clicking the top dc= entry.

B Then, on the Edit menu, point at **Add Entry** and click **organization** (or **organizationalUnit**); see Figure 102.

Figure 102 Adding a New LDAP Entry

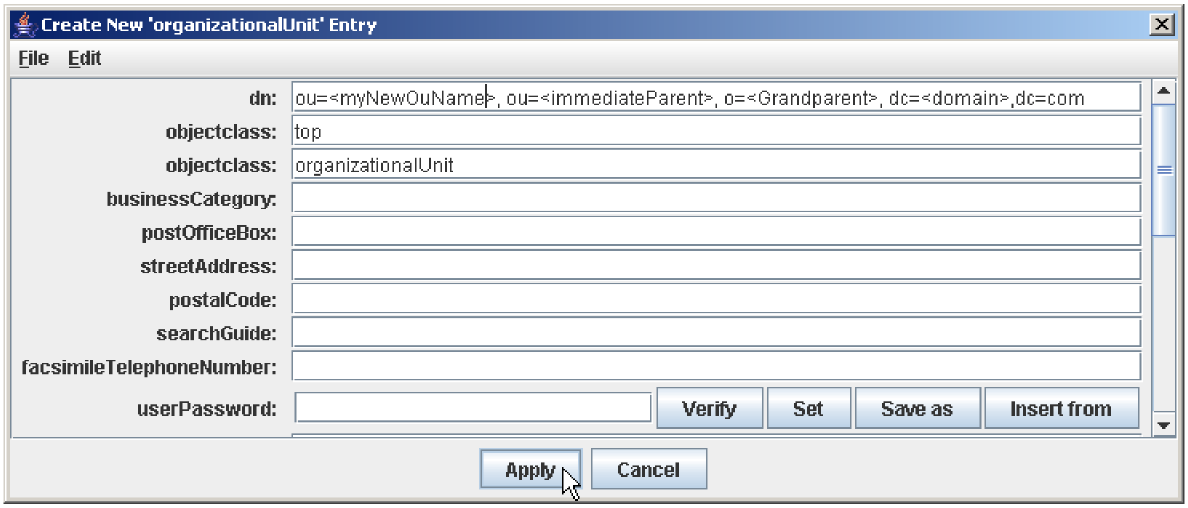


Attribute	Value
ou	<myNewOU>
	top
	organization...

Ready. No entries returned.

C Finally, in the **Create New ... Entry** dialog box, make any needed modifications, and then click the **Apply** button; see Figure 103.

Figure 103 Setting Attributes of a New LDAP Entry

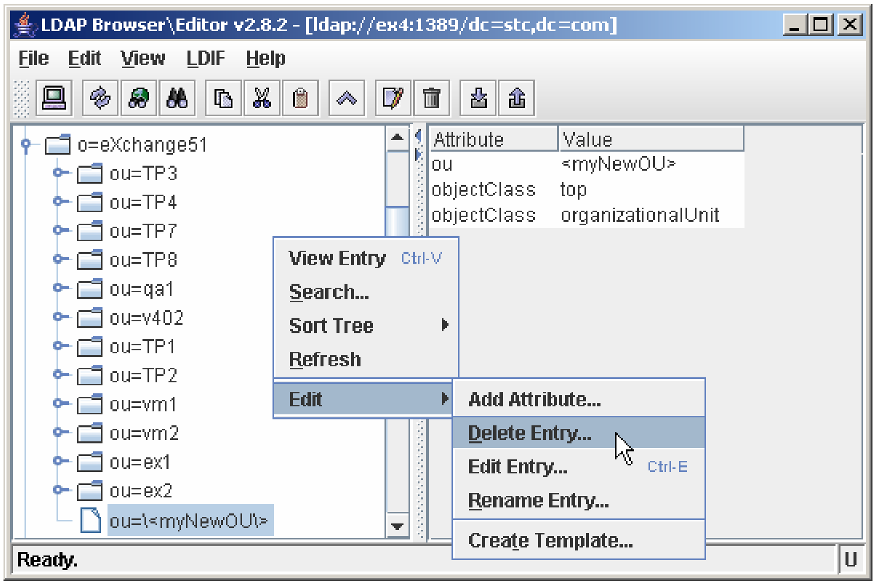


dn: ou=<myNewOuName>, ou=<immediateParent>, o=<Grandparent>, dc=<domain>,dc=com
objectclass: top
objectclass: organizationalUnit
businessCategory:
postOfficeBox:
streetAddress:
postalCode:
searchGuide:
facsimileTelephoneNumber:
userPassword:

3 To delete a non-root entry::

A Right-click the entry and, on the context menu, point at **Edit** and then click **Delete Entry**; see Figure 104.

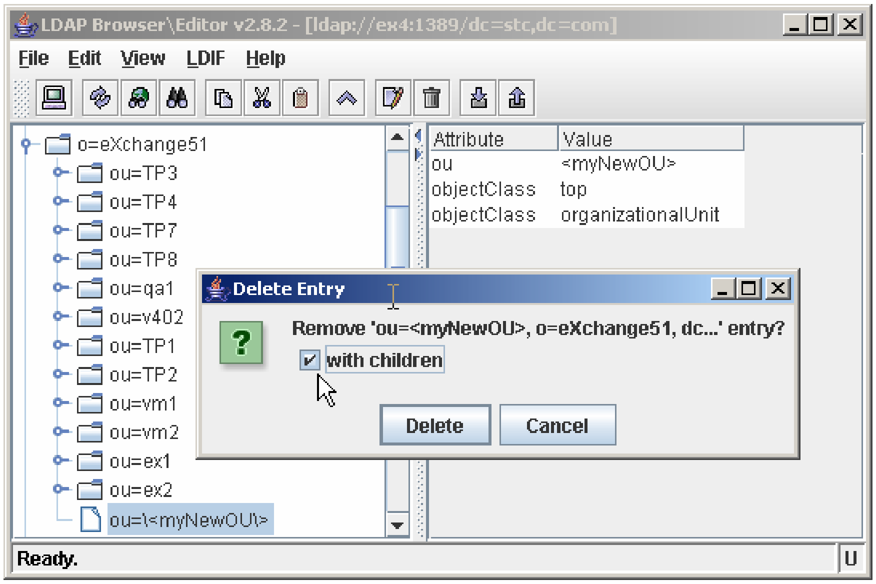
Figure 104 Deleting an LDAP Entry



Attribute	Value
ou	<myNewOU>
objectClass	top
objectClass	organizationalUnit

B In the **Delete Entry** dialog box, click or clear the "**with children**" checkbox and then click the **Delete** button; see Figure 105.

Figure 105 Confirming an LDAP Entry Deletion



Attribute	Value
ou	<myNewOU>
objectClass	top
objectClass	organizationalUnit

Remove 'ou=<myNewOU>, o=eXchange51, dc...' entry?

☑ with children

Appendix B

Method Palette

This appendix describes each method that appears in the Method Palette of the eXchange Protocol Designer. Many of these methods are based on standard XPath functions. Refer to the W3C XML Path Language documentation for additional information.

What's in This Appendix

B.1 Method Palette Overview

The Method Palette displays a menu of methods that you can incorporate into your process models to manipulate data in a variety of ways. These methods are grouped logically by the type of function performed. By default, not all methods appear in the Method Palette menus. To view complete lists of operators for each type, click **Settings** from any of the Method Palette menus. The Method Palette dialog appears, and you can select and deselect functions to indicate which functions should appear in each menu.

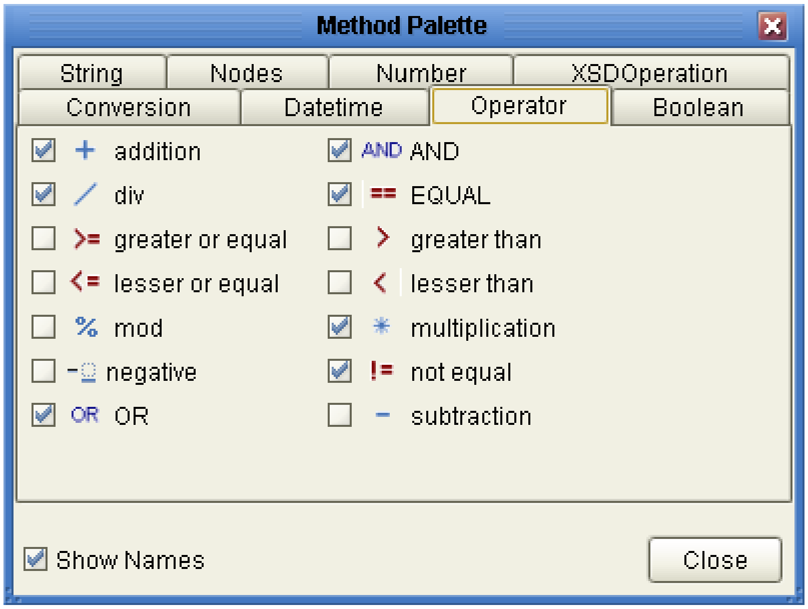
To display a method on the eXchange Protocol Designer canvas, click the menu containing the method you want to use and then click the method name or symbol. A method box appears on the canvas, and you can then map the input and output values for the method.

For most of the methods, you can either drag an object from the attributes tree to the method box to enter a value into a parameter, or you can right click a parameter in the method box to add a literal value directly into the box.

B.2 Operator Methods

Operator methods allow you to manipulate data with standard mathematical and comparison operators. To display a complete list of available operator methods, click **Operator** and then click **Settings** (see Figure 106).

Figure 106 Method Palette: Operator Tab



☑ + addition
☑ / div
□ >= greater or equal
□ <= lesser or equal
□ % mod
□ - negative
☑ OR OR
☑ AND AND
☑ == EQUAL
□ > greater than
□ < lesser than
☑ * multiplication
☑ != not equal
□ - subtraction

☑ Show Names

Table 29 Operator Methods

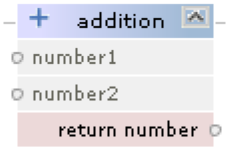
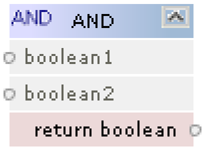
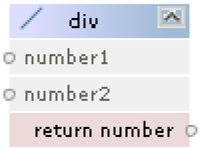
Method Box	Description
	The **addition** method adds the numerical value of *number1* to the numerical value of *number2*, and returns the sum.
	The **AND** method returns Boolean true if both *boolean1* and *boolean2* are true; otherwise, it returns Boolean false. **Note:** Boolean parameters cannot accept values with capital letters. For example, "True" and "TRUE" are not valid Boolean values, but "true" is.
	The **div** method divides the numerical value of *number1* by the numerical value of *number2*, and returns the quotient.

Table 29 Operator Methods (Continued)

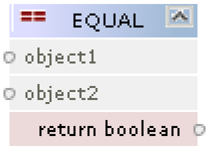
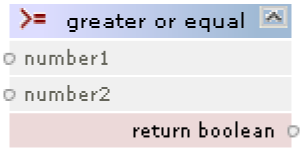
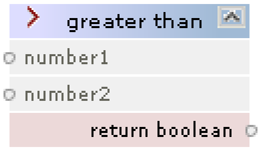
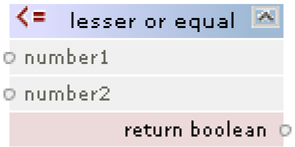
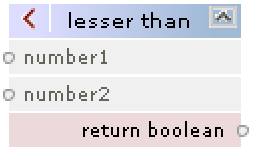
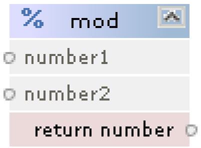
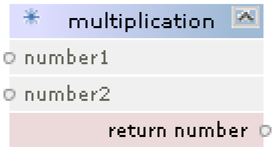
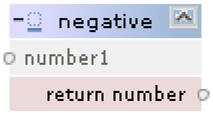
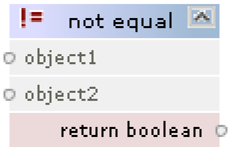
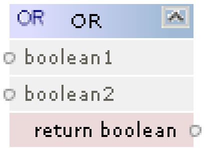
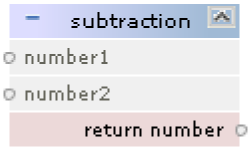
Method Box	Description
 == EQUAL object1 object2 return boolean	The **EQUAL** method returns Boolean true if *object1* is equal to *object2*; otherwise, it returns Boolean false.
 >= greater or equal number1 number2 return boolean	The **greater or equal** method returns Boolean true if the numerical value of *number1* is greater than or equal to the numerical value of *number2*; otherwise, it returns Boolean false.
 > greater than number1 number2 return boolean	The **greater than** method returns Boolean true if the numerical value of *number1* is greater than the numerical value of *number2*; otherwise, it returns Boolean false.
 <= lesser or equal number1 number2 return boolean	The **lesser or equal** method returns Boolean true if the numerical value of *number1* is less than or equal to the numerical value of *number2*; otherwise, it returns Boolean false.
 < lesser than number1 number2 return boolean	The **lesser than** method returns Boolean true if the numerical value of *number1* is less than the numerical value of *number2*; otherwise, it returns Boolean false.
 % mod number1 number2 return number	The **mod** method divides the numerical value of *number1* by the numerical value of *number2,* and returns only the remainder.
 * multiplication number1 number2 return number	The **multiplication** method multiplies the numerical value of *number1* by the numerical value of *number2*, and returns the product.
 - negative number1 return number	The **negative** method converts the numerical value of *number1* to its negative. The result is a number having the same absolute value as the input number. For example, if the input is “10”, it is converted to “-10”; if the input is “-10”, it is converted to “10”.

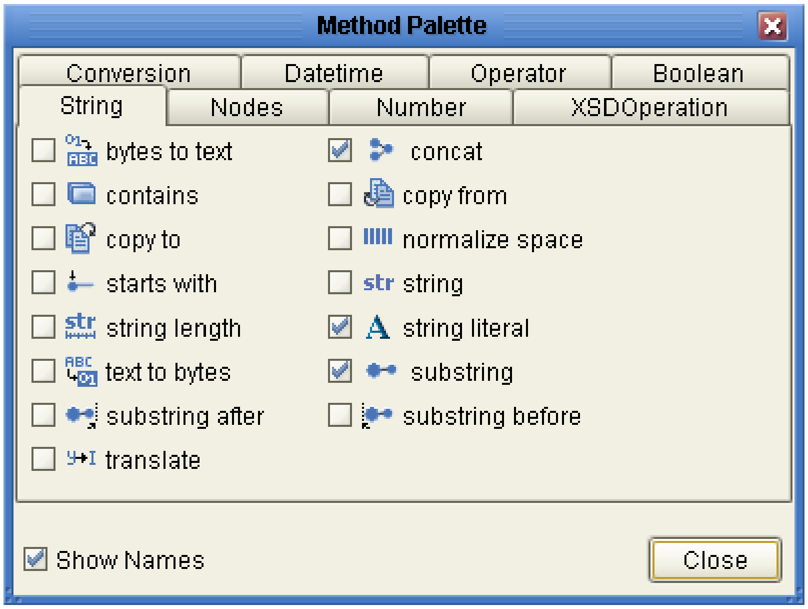
Table 29 Operator Methods (Continued)

Method Box	Description
 != not equal object1 object2 return boolean	The **not equal** method returns Boolean true if *object1* is not equal to *object2*; otherwise, it returns Boolean false.
 OR OR boolean1 boolean2 return boolean	The **OR** method returns Boolean false if both *boolean1* and *boolean2* are false; otherwise, it returns Boolean true. **Note:** Boolean parameters cannot accept values with capital letters. For example, "True" and "TRUE" are not valid Boolean values, but "true" is.
 – subtraction number1 number2 return number	The **subtraction** method subtracts the numerical value of *number2* from the numerical value of *number1*, and returns the difference.

B.3 String Methods

The String methods allow you to manipulate string data. To display a complete list of available string methods, click **String** and then click **Settings** (see Figure 107).

Figure 107 Method Palette: String Tab



- ☐ bytes to text
- ☐ contains
- ☐ copy to
- ☐ starts with
- ☐ string length
- ☐ text to bytes
- ☐ substring after
- ☐ translate
- ☑ concat
- ☐ copy from
- ☐ normalize space
- ☐ string
- ☑ string literal
- ☑ substring
- ☐ substring before

☑ Show Names

Table 30 String Methods

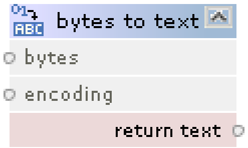
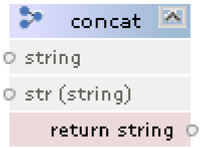
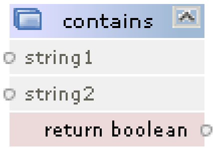
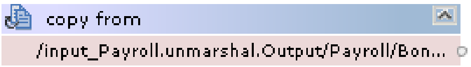
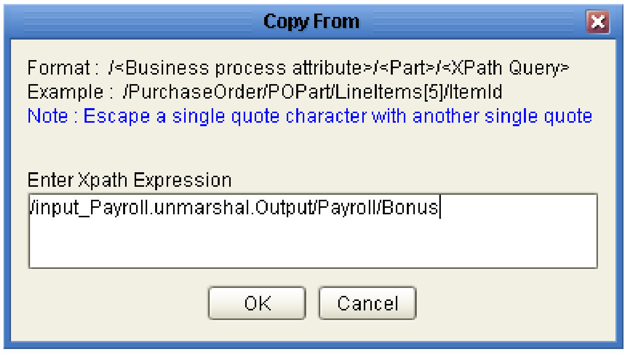
Method Box	Description
	The **bytes to text** method decodes a bytes object into text using the specified encoding. If no encoding is specified, the platform's default encoding is used. For a list of possible values for the *encoding* parameter, see **Encoder Parameter Values** on page 204.
	The **concat** method returns the string created by concatenating the value of the *str* parameter to the end of the value of the *string* parameter. **Note:** You can add multiple *str* parameters to the method. The method automatically adds an unmapped node as needed.
	The **contains** method returns Boolean true if the value of *string1* is contained within the value of *string2*; otherwise, it returns false
	The **copy from** method allows you to type in an XPath expression as the source of a copy operation. This is useful for entering XPath predicates. When you select **copy from**, the Copy From dialog box appears for you to enter the XPath.  Copy From Format : /<Business process attribute>/<Part>/<XPath Query> Example : /PurchaseOrder/POPart/LineItems[5]/ItemId Note : Escape a single quote character with another single quote Enter Xpath Expression /input_Payroll.unmarshal.Output/Payroll/Bonus **Note:** This feature is for advanced users who are familiar with XPath and BPEL syntax.

Table 30 String Methods (Continued)

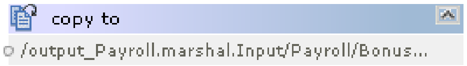
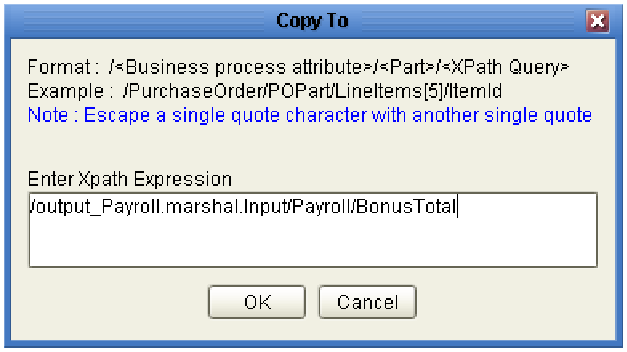
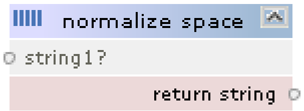
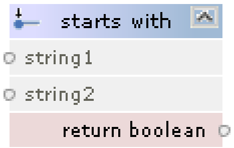
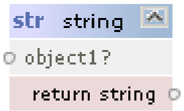
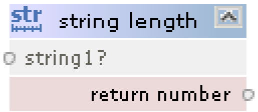
Method Box	Description
 copy to /output_Payroll.marshal.Input/Payroll/Bonus...	The **copy to** method allows you to type in an XPath expression as the destination of a copy operation. This is useful for entering XPath predicates. When you select **copy to**, the Copy To dialog box appears for you to enter the XPath.  Copy To Format : /<Business process attribute>/<Part>/<XPath Query> Example : /PurchaseOrder/POPart/LineItems[5]/ItemId Note : Escape a single quote character with another single quote Enter Xpath Expression /output_Payroll.marshal.Input/Payroll/BonusTotal OK Cancel **Note:** This feature is for advanced users who are familiar with XPath and BPEL syntax.
 normalize space string1? return string	The **normalize space** method removes leading and trailing spaces from the string contained in *string1*.
 starts with string1 string2 return boolean	The **starts with** method returns Boolean true if the value of *string1* starts with the value of *string2*; otherwise, it returns false.
 string object1? return string	The **string** method converts the value of *object1* to a string.
 string length string1? return number	The **string length** method returns the number of characters in *string1*.

Table 30 String Methods (Continued)

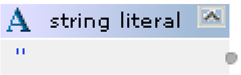
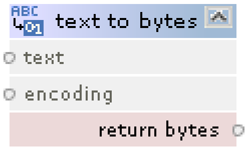
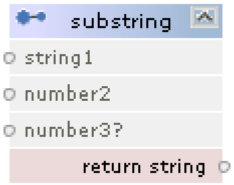
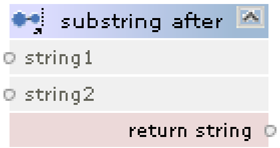
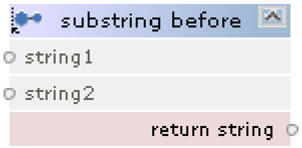
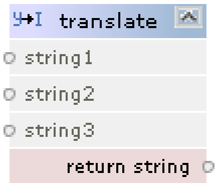
Method Box	Description
	The **string literal** method allows you to create a literal string; that is, a sequence of characters of fixed length and content. Type the literal value directly into the method box. **Tips:** ▪ Use *Control+Enter* to insert a line break in a literal. ▪ To add a string literal directly into a parameter without using the string literal method, right-click the parameter, and then click **Add Literal**. Type the literal directly into the parameter field. ▪ To edit an existing literal from a method, right-click the literal, click **Edit**, and then modify the value. ▪ To delete a literal from a method, right-click the literal, and then click **Delete Literal**.
	The **text to bytes** method encodes the input text into a sequence of bytes using the specified encoding. If no encoding is specified, the platform's default encoding is used. For a complete list of available values for the encoding parameter, see **Encoder Parameter Values** on page 204.
 substring string1 number2 number3? return string	The **substring** method returns a substring of *string1*, with *number2* indicating the position of the first character in the substring and *number3* indicating the number of characters to include. *number2* represents the position of a character from the beginning of *string1*. If *number3* is empty, the return string includes the characters from the position indicated by *number2* to the end of *string1*. For example, if the parameters are ("Input Data", 6, 3), the return string is "Dat". If the third parameter is empty, the return string is "Data". **Note:** The first character position is 1, not 0.
	The **substring after** method returns the substring in *string1* that occurs after the first instance of the substring in *string2*. For example, if the parameters are ("Input Data", "Input "), the return string is "Data". You would get the same return string if the second parameter contained only a space. **Note:** The first character position is 1, not 0.

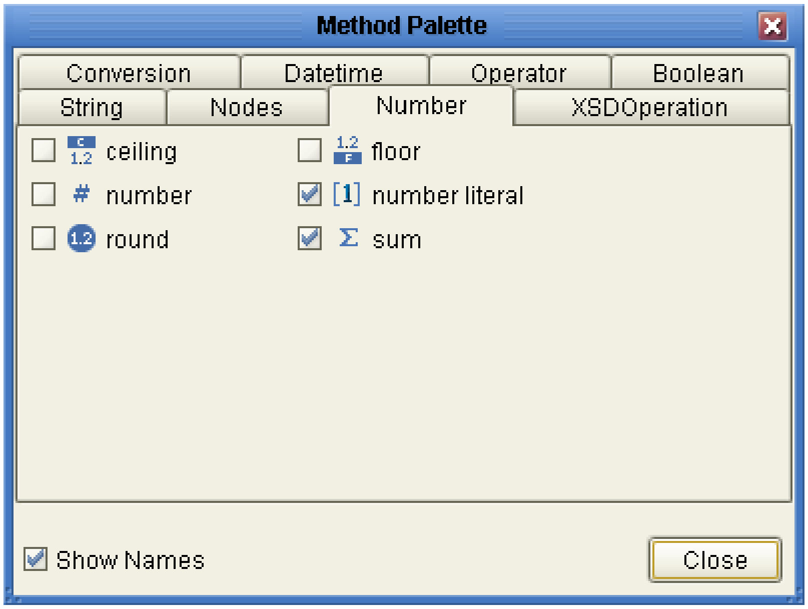
Table 30 String Methods (Continued)

Method Box	Description
 substring before string1 string2 return string	The **substring before** method returns the substring in *string1* that occurs before the first instance of the substring in *string2*. For example, if the parameters are ("Input Data", " Data", the return string is "Input". You would get the same return string if the second parameter contained only a space. **Note:** The first character position is 1, not 0.
 translate string1 string2 string3 return string	The **translate** method returns the value of *string1*, replacing any occurrences of the characters in *string2* with the characters in the corresponding positions in *string3*. If *string2* contains more than one instance of a character, only the first instance is taken into account. If *string3* is longer than *string2*, the extra characters in *string3* are ignored. If *string2* is longer than *string3*, the extra characters in *string2* are removed from *string1*. For example, if the parameters are ("Sun","un","MI"), the original string, Sun, is translated to SMI.

B.4 Number Methods

The Number methods allow you to manipulate numeric data. To display a complete list of available number methods, click **Number** and then click **Settings** (see Figure 108).

Figure 108 Method Palette: Number Tab

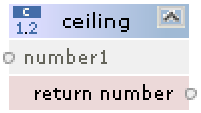
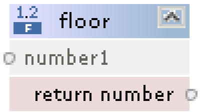
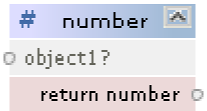
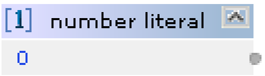
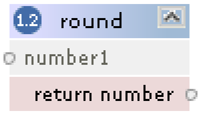
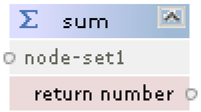


Method Palette

☐ ceiling ☐ floor
☐ number ☑ number literal
☐ round ☑ sum

☑ Show Names

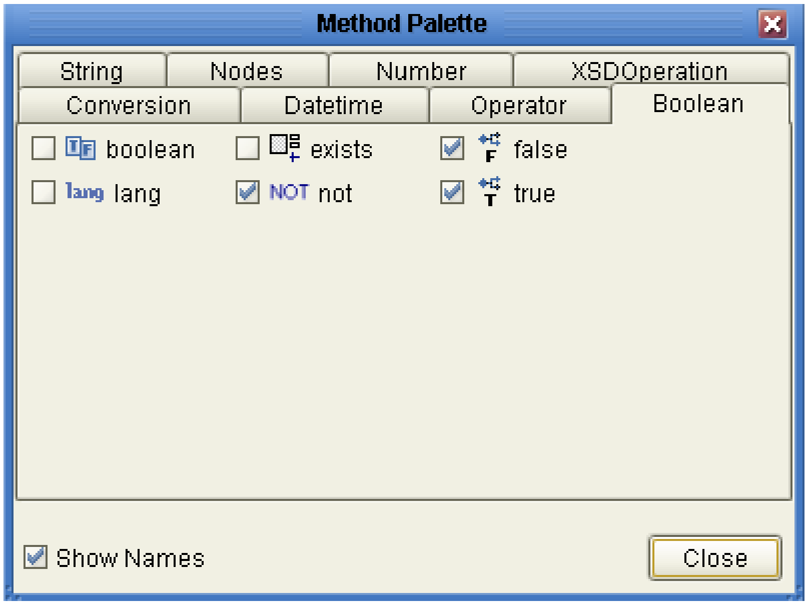
Table 31 Number Methods

Method Box	Description
	The **ceiling** method returns the smallest integer that is not less than the value of *number1*.
	The **floor** method returns the largest integer that is not greater than the value of *number1*.
	The **number** method converts *object1* to a numerical value according to the following rules: ▪ Boolean true is converted to 1; Boolean false is converted to 0. ▪ A string that includes whitespace at the beginning and end, a minus sign, and a number (where the whitespace and minus sign are optional) is converted to the closest IEEE 754 number. ▪ Strings that do not compute to a numerical value are converted to NaN. ▪ A set of nodes is converted to a string and then converted as a string (as described above).
	The **number literal** method allows you to create a number of fixed length and content. You can type the value directly into the method box.
	The **round** method rounds the value of *number1* to the nearest integer.
	The **sum** method returns the total value of the set of numeric values in *node-set1*.

B.5 Boolean Methods

Boolean methods allow you to apply boolean logic to your data. To display a complete list of available Boolean methods, click **Boolean** and then click **Settings** (see Figure 109).

Figure 109 Method Palette: Boolean Tab



□ boolean □ exists ☑ false
□ lang ☑ NOT not ☑ true

☑ Show Names

Table 32 Boolean Methods

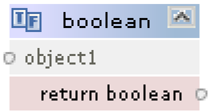
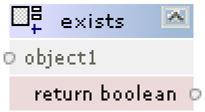
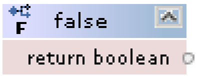
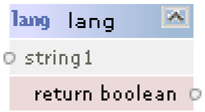
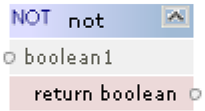
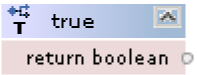
Method Box	Description
 boolean object1 return boolean	The **boolean** method converts the value of *object1* to Boolean, and returns true or false.
 exists object1 return boolean	The **exists** method checks whether a value is present in *object1*. If a value is present, it returns Boolean true; otherwise, it returns Boolean false.
 false return boolean	The **false** method always returns Boolean false.
 lang string1 return boolean	The **lang** method returns Boolean true if the value of *string1* matches the language of the xsl:lang element; otherwise, it returns Boolean false.

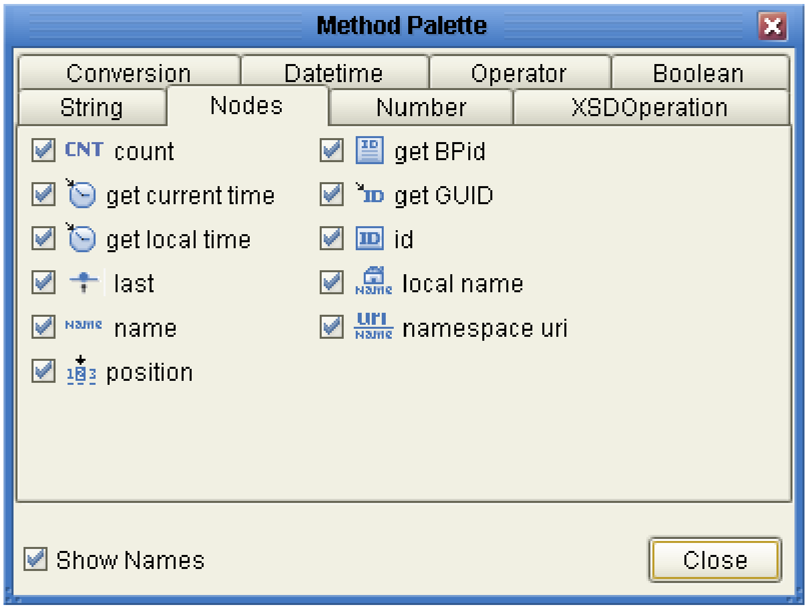
Table 32 Boolean Methods (Continued)

Method Box	Description
	The **not** method returns Boolean true if *boolean1* is false, and returns Boolean false if *boolean1* is true. **Note:** Boolean parameters cannot accept values with capital letters. For example, “True” and “TRUE” are not valid Boolean values, but “true” is.
	The **true** method always returns Boolean true.

B.6 Node Methods

Node methods allow you to manipulate node data. To display a complete list of available nodes methods, click **Nodes** and then click **Settings** (see Figure 110).

Figure 110 Method Palette: Nodes Tab



☑ count
☑ get current time
☑ get local time
☑ last
☑ name
☑ position
☑ get BPid
☑ get GUID
☑ id
☑ local name
☑ namespace uri
☑ Show Names

Table 33 Nodes Methods

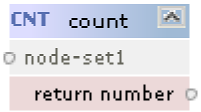
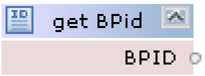
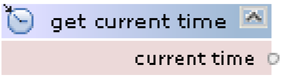
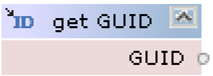
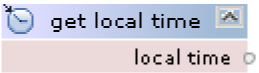
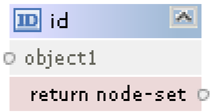
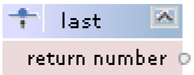
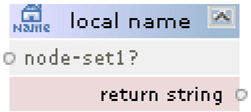
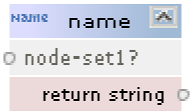
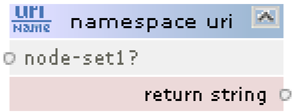
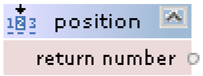
Method Box	Description
	The **count** method returns the number of nodes in the node-set contained in *node-set1*.

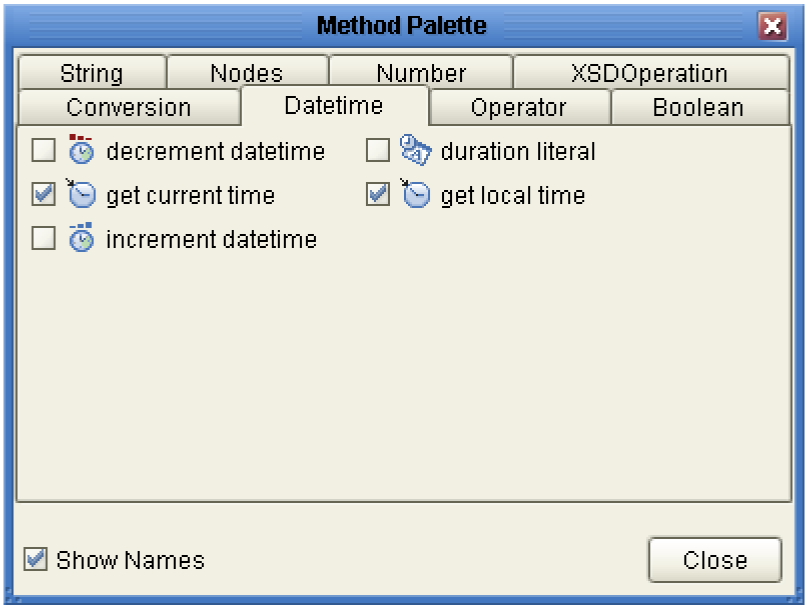
Table 33 Nodes Methods (Continued)

Method Box	Description
	The **get BPid** method returns the business process instance ID.
	The **get current time** method returns the current time in ISO 8601 format (for example, 2003-08-15T02:03:49.92Z).
	The **get GUID** method returns a randomly generated globally unique ID.
	The **get local time** method returns the local time in ISO 8601 format (for example, 2003-08-15T02:03:49.92Z).
	The **id** method returns a node-set of elements selected by the unique ID specified in *object1*.
	The **last** method returns the position number of the last node in the processed node list.
	The **local name** method returns the local part of the first node in *node-set1*. A node usually consists of a prefix and a colon, followed by the local name.
	The **name** method returns the name of the first node in *node-set1*.
	The **namespace url** method returns the namespace URL of the first node in *node-set1*.
	The **position** method returns the position in the node list of the node that is currently being processed

B.7 Datetime Methods

Datetime methods allow you to manipulate dates, times, and duration of data. To display a complete list of available Datetime methods, click **Datetime** and then click **Settings** (see Figure 111).

Figure 111 Method Palette: Datetime Tab



☐ decrement datetime
☐ duration literal
☑ get current time
☑ get local time
☐ increment datetime
☑ Show Names

Table 34 Datetime Methods

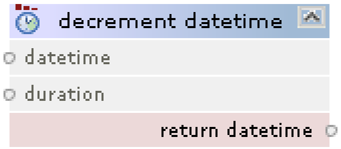
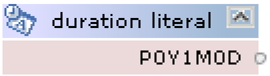
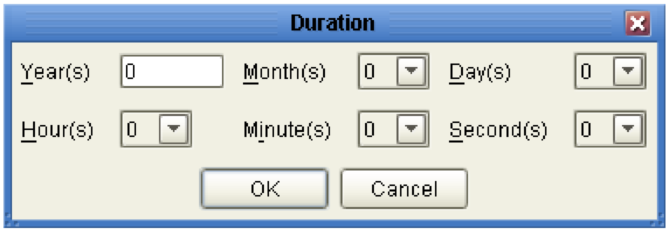
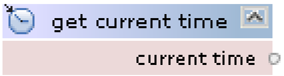
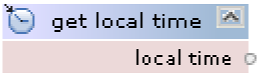
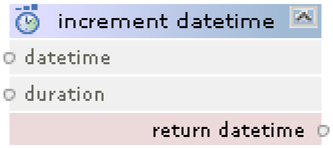
Method Box	Description
 decrement datetime datetime duration return datetime	The **decrement datetime** method dynamically decreases the date or time by a certain duration, such as days or hours. Use the **duration literal** method to create the value of the *duration* parameter to ensure the string is in the correct format.

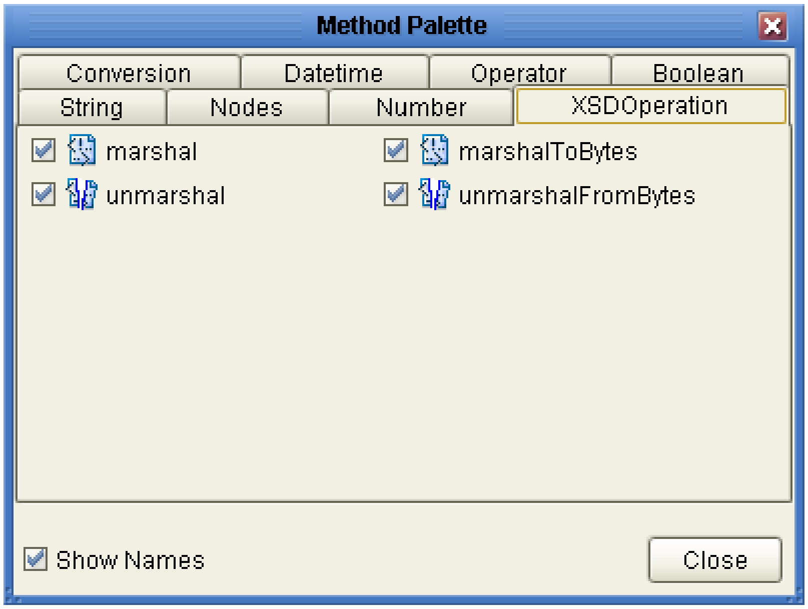
Table 34 Datetime Methods (Continued)

Method Box	Description
	The **duration literal** method allows you to set a specific date or time. When you select **duration literal**, the Duration dialog box appears, and you can select the duration in years, months, days, hours, minutes, and seconds.  Duration Year(s) 0 Month(s) 0 Day(s) 0 Hour(s) 0 Minute(s) 0 Second(s) 0
	The **get current time** method returns the current time in ISO 8601 format (for example, 2003-08-15T02:03:49.92Z).
	The **get local time** method returns the local time in ISO 8601 format (for example, 2003-08-15T02:03:49.92Z).
 increment datetime datetime duration return datetime	The **increment datetime** dynamically increases the date or time by a certain duration, such as days or hours. Use the **duration literal** method to create the value of the *duration* parameter to ensure the string is in the correct format.

B.8 XSD Operation

The XSD Operation methods enable you to marshal and unmarshal messages. To display a complete list of available XSD operation methods, click **XSDOperation** and then click **Settings** (see Figure 112).

Figure 112 Method Palette: XSDOperation Tab



☑ marshal ☑ marshalToBytes

☑ unmarshal ☑ unmarshalFromBytes

☑ Show Names

Table 35 XSD Operation Methods

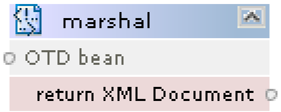
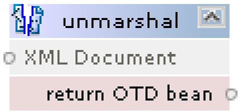
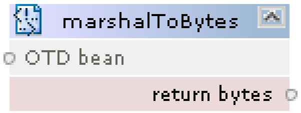
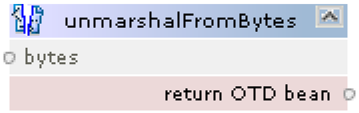
Method Box	Description
	The **marshal** method marshals the information in the OTD bean, and returns an XML document.
	The **unmarshal** method unmarshals the information in an XML document, and returns an OTD bean.
	The **marshalToBytes** method marshals the information in the OTD bean, and returns the data as bytes.

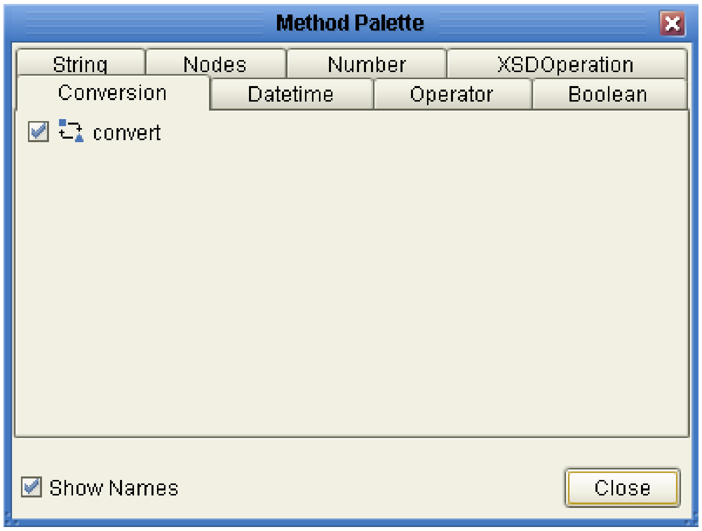
Table 35 XSD Operation Methods

Method Box	Description
	The **unmarshalFromBytes** method unmarshals the bytes input, and returns an OTD bean.

B.9 Conversion

There is one Conversion method, **convert**, that allows you to make conversions from various data types. To display the settings for the Conversion menu, click **Conversion** and then click **Settings** (see Figure 113).

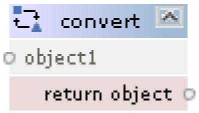
Figure 113 Method Palette: Conversion Tab



☑ convert

☑ Show Names

Table 36 Conversion Methods

Method Box	Description
	The **convert** method converts the input object to the output object. This method is only supported in version 5.0.X Projects.

B.10 Encoder Parameter Values

The **text to bytes** and **bytes to text** methods both take a parameter that specifies the type of encoding to use when performing the conversion. The possible values for the encoder parameter include the following:

- Big5
- Big5-HKSCS
- EUC-JP
- EUC-KR
- GB18030
- GBK
- ISO-2022-JP
- ISO-2022-KR
- ISO-8859-1
- ISO-8859-13
- ISO-8859-15
- ISO-8859-2
- ISO-8859-3
- ISO-8859-4
- ISO-8859-5
- ISO-8859-6
- ISO-8859-7
- ISO-8859-8
- ISO-8859-9
- JIS_X0201
- JIS_X0212-1990
- KO18-R
- Shift_JIS
- TIS-620
- US-ASCII
- UTF-16
- UTF-16BE
- UTF-16LE
- UTF-8
- windows-1250
- windows-1251
- windows-1252
- windows-1253
- windows-1254
- windows-1255
- windows-1256
- windows-1257
- windows-1258
- windows-31j
- x-EUC-CN
- x-euc-jp-linux
- x-EUC-TW
- x-eucJP-Open
- x-ISC1191
- x-JIS0208
- x-JISAutoDetect
- x-Johab
- x-MS950-HKSCS
- x-mswin-936
- x-windows-50220
- x-windows-50221
- x-windows-949
- x-windows-950
- x-windows-iso2022jp

Appendix C

Scalability

This appendix is a modified version of a white paper published previously as "Sun B2B Suite 2.0 Scalability: Scaling Horizontally to Increase Throughput".

What's in This Appendix

C.1 Abstract and Summary

This appendix discusses how to increase throughput for Sun B2B Suite 2.0 by scaling horizontally—that is, by running additional instances of Sun B2B Suite on a single machine so that each instance communicates with the same set of back-end servers running on different machines.

Summary of Findings

At its first release (on Java CAPS 5.1.2), Sun B2B Suite 2.0 has been verified to have a throughput performance gain as high as 24% when two instances are run in a single machine. However, at this time the performance gain is limited to a scenario where each instance has a unique combination of B2B Host, Trading Partner, and Transaction Profile. (If additional instances are added that do not have such a unique combination, the instances function properly, but do not show any significant gain in performance.)

Background Discussion: Considerations for Verifying Correlation

For horizontal scalability, it is necessary to ensure that each instance of Sun B2B Suite instance can operate independently of other instances. Specifically, documents that take part in a long-running dialog must not be bound to a specific instance.

For example, if an outbound X12 850 document is sent from one instance of Sun B2B Suite, then subsequent documents in the dialog should not be required to return to the same instance that originated the outbound 850. Instead, another instance should be used to correlate the responding TA1, X12 997, or X12 855 documents, and to trace this dialog in the shared LDAP and Oracle database back-end servers.

After correlation independence is achieved on a long-running dialog, it becomes possible to run scalability scenarios to measure and compare results of one instance and two instances of Sun B2B Suite. However, each scenario must include preparation steps to clear out database and logs; this cleans up the environment and primes the instances for optimal performance.

To verify that correlation was occurring between separate instances, a preliminary test was conducted on a three-machine setup wherein document requests and replies were correlated independently of the originating instance. This appendix omits discussion of the preliminary test to verify correlation; for details, refer to the white paper.

C.2 Testing to Verify Scalability

Scenarios were set up in pairs: In one case, a single instance of Sun B2B Suite was used to process a fixed number of outbound X12 850 documents; in the other case of the pair, two instances were used to process the same fixed number of X12 850 documents. The throughputs from the two cases were compared to show performance gain or loss. For example, the throughput of one instance processing 100 documents was compared to the throughput of two instances each processing 50 documents.

C.2.1 Test Environment

To ensure reliability and reproducibility of results, all testing was conducted on a single trial machine that ran either one instances or two instances of Sun B2B Suite. The trial machine interacted with an unvarying support machine running back-end servers such as for LDAP and Oracle, as well as other back-end processes.

Trial Machine

The trial machine was a Sun Fire™ V440 with four SPARC® CPUs running Solaris 10 with 8 GB of memory. It was dedicated to running a Logical Host containing two domains (domainScale1 and domainScale2), as well as Enterprise Manager (to view and control the deployed domain or domains).

During execution of the scenarios, CPU utilization on the trial machine ranged from 50% to 90%, and memory usage ranged from 700 MB to 900 MB.

Support Machine

The support machine was a Dell Latitude™ 610 with a single CPU running Windows XP SP2 with 2 GB of memory. It ran the back-end servers (Sun Java Directory Server 5.2 and Oracle 9*i*) as well as an instance of a Logical Host (for ePM) on a domain named domain1. During scenario execution, CPU utilization on the support machine never exceeded 35%.

Networking

Network communication between the trial machine and the support machine took place over a wired 100-Mbit switched network connection.

Logging Levels

All tests were run with the Module Log Level set to SEVERE for all individual server modules. (This is neither the default nor the recommended setting; it was used to minimize spent writing to log files.)

C.2.2 Test Scenario

To ensure reliability and reproducibility of results, all testing was conducted on a single trial machine that ran either one instances or two instances of Sun B2B Suite. The trial machine interacted with an unvarying support machine running back-end servers such as for LDAP and Oracle, as well as other back-end processes.

C.3 Results

A 24% performance gain when running two domains compared to one domain; each domain was running one instance of eXchange.

- One-domain scenario: Batch size 100, running 5000 messages outbound takes 18 minutes 55 seconds.
- Two-domain scenario: Batch size 100, running 2500 messages per domain takes 14:27. Our current limitation requires that each instance of eXchange handle unique combination of: host, trading partner, transaction type. So for our two domain scenario, each domain processed 850 transactions for different partner: TPBerlin and TPBerlin2.
- Difference = 4:28—a 24% performance gain.

C.4 Additional Technical Notes

- An Oracle v9.2 database was running on local desktop: P4 3.4 GHz, 2 GB RAM. Utilization was 25-30% during all tests.
- For LDAP, Sun Java System Directory Server 5.2 server running on local desktop.
- Duration was measured by monitoring the timestamp of first message entering EX_FROMINTERNAL topic to the last outbound file produced.
- Messages were queued on each domain to a queue called Q_FROMINTERNAL. SendFromInternal project was modified to load messages to this queue for each domain.

- Q2T project on each domain took messages on Q_FROMINTERNAL to EX_FROMINTERNAL topic to begin outbound processing. This project was disabled to queue up messages, and once all messages were ready in the queue, enabled (on both domains for two domain scenario) to begin outbound processing.
- CPU utilization on was up to around 60-90% during all tests.
- Memory usage for each domain was around 700MB to 900MB during all tests.
- The following tasks were completed before each test run:

  A Stop all domains

  B Recreate eXchange database.

  C Delete STCMS directory to clear all topics and queues.

  D Archive domain log directories. archive other files generated from topics and outbound files.

  E Start domains

  F Process enough messages to produce one outbound file to prime the eXchange instances.

  G Disable Q2T project and load messages to process using SendFromInternal project.

  H Enable Q2T project to begin sending queued messages to EX_FROMINTERNAL topic to begin outbound processing.

Appendix D

Reliability

This appendix is a modified version of a white paper published previously as "Sun B2B Suite 2.0 Reliability: Configuring the System for Continuous Run".

What's in This Appendix

D.1 Abstract and Summary

This appendix discusses how to maintain reliability for Sun B2B Suite 2.0 so as to run a scenario continuously—without any restarts—for more than a week.

Summary of Findings

At its first release (on Java CAPS 5.1.2), Sun B2B Suite 2.0 has been verified to have a reliability of no downtime in more than a 4.5 days of continuous operation, with a steady throughput of approximately 3.3 transactions per second (TPS) in the test environment, using the selected test case.

(*Editor's Note:* After the original white paper was published, additional testing showed similar reliability in a run that lasted more than a week with no down time.)

Various parameter settings were chosen to help achieve this result; the purpose of this paper is to serve as a guide to duplicating these results.

D.2 Testing to Verify Reliability

To certify the run, the ASC X12 protocol was selected for the test.

D.2.1 Test Case

The test has been performed using a single trading partner. Two hundred X12 810 messages with a payload of 947 bytes were fed to the system every minute through the FILE eWay receives. The system was configured to batch up 100 messages at a time.

The protocol, payload and batch size where based on data provided by a customers. They had been used for testing with a previous version of the product and thus provided a good benchmark to verify the reliability of our current release.

The test case isolated the scenario described above from any external interference. No other application was allowed on either the server and the database machines during the test.

The messages were being sent without expecting (and later on, receiving) any acknowledgements. More precisely, the system was configured to expect responses and issue resends neither at the message level, nor at the batch (Interchange) level, and no paired instance was consuming the messages at the other end. The processed batches were simply being written to the disk using a FILE eWay write and then periodically deleted by a custom Java program to avoid disk overflowing.

The criterion for test success was to achieve at least 1.2 million messages processed over 4.5 days without any restarts or active maintenance of the system.

D.2.2 Test Environment

To ensure reliability and reproducibility of results, all testing was conducted on a single trial machine that ran either one instances or two instances of Sun B2B Suite. The trial machine interacted with unvarying support machines running back-end servers such as for LDAP and Oracle, as well as other back-end processes.

Trial Machine

The trial machine was a Sun Fire™ V440 with four SPARC® CPUs running Solaris 10 with 8 GB of memory. It was dedicated to running a Logical Host containing two domains (domainScale1 and domainScale2), as well as Enterprise Manager (to view and control the deployed domain or domains).

During execution of the scenarios, CPU utilization on the trial machine ranged from 50% to 90%, and memory usage ranged from 700 MB to 900 MB.

Database and LDAP Servers

Oracle 9i was set up on a Windows XP desktop machine that was on the local network. The Oracle instance had been installed using the default multi-purpose configuration, and no special database tuning had been performed for this test.

The Sun Java System Directory Server 5.2 was used for LDAP support. It is highly recommended to use this server over OpenLDAP because of performance considerations.

Networking

Network communication between the trial machine and the support machines took place over a wired 100-Mbit switched network connection.

Logging Levels

All tests were run with the Module Log Level set to SEVERE for all individual server modules. (This is neither the default nor the recommended setting; it was used to minimize spent writing to log files.)

D.2.3 Methodology of Certifying Reliability

Although the logging was set to SEVERE for the entire Application Server, in order to increase the system's performance, the server log has been monitored all along the run and verified at the end to contain no error messages.

The system was continuously supervised by connecting to Enterprise Manager running on the same machine, using a local browser. The queues were found to be constantly cleaning up shortly after the message feed and had never shown any backlog.

The most important consideration was this: topic Min Sequence, Max Sequence and Available counters were used to certify over 1.2 million being processed by the system at the end of the 4.5 days. In the EX_MSG_BATCH_RCVR topic, Min Sequence and Max Sequence were both showing an identical number, higher than 1.2 million, meaning that many messages went into the Batcher Service. For the EX_TODELIVERY topic, these numbers were also equal and higher than 12,000, meaning more than 12,000 batches of 100 messages have been posted to and later on retrieved from that topic. EX_TODELIVERY being the last topic in the B2B Suite's pipeline, the conclusion was that all messages were picked up and sent towards the Trading Partner via the Delivery Channel Handler. The Available was zero for all topics.

Finally, the simple Java program that was used to delete the output files also printed the sequence number of each file immediately after its deletion. The program logged over 1.2 such deletions, certifying that all the messages came out of the FILE eWay-based Delivery Channel in use.

D.3 Important Settings

One of the most valuable lessons to come from the reliability testing was the tuning of various parameters to achieve the results. In order to achieve the goal of 1.2 million messages processed within 4.5 days, several of the settings used by the B2B suite had to be customized. Some of these settings are provided when the product is shipped; others must be manually configured at design time.

Many of these same settings are also needed to ensure reliability.

Note: *The following parameters should be used as a good starting point when setting up an environment for customers. However, actual parameters will vary depending on the actual environment.*

D.3.1 Design-Time Parameters (Enterprise Designer)

Design-time parameters affecting performance are found both in the Environment (properties of Oracle external systems and, within each Logical Host, properties of Integration servers) and in each Project (properties of BPs).

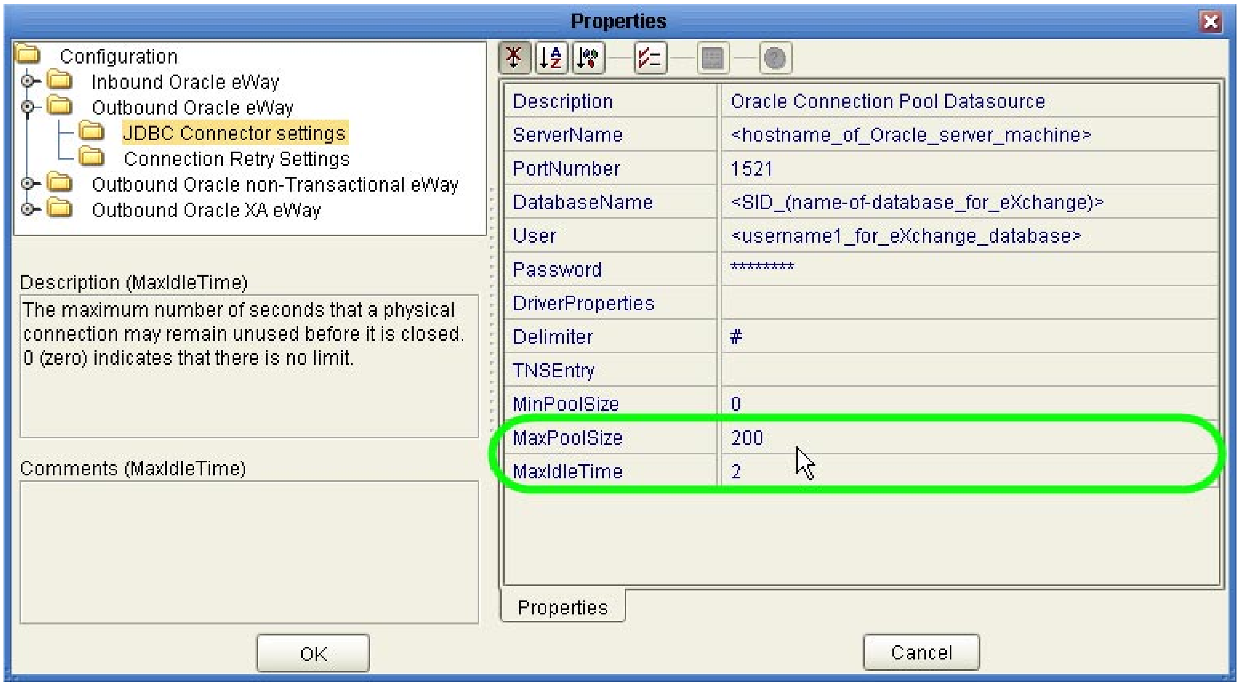
Environment Settings for External Systems: Oracle

The following properties for Oracle externals are set under **Configuration > Outbound Oracle eWay** and **Outbound Oracle non-Transactional eWay**. Parameter settings should be adjusted whenever a system does not have enough connections available to the Oracle database, or if the system times out on connecting to Oracle when the database is up and running.

JDBC Connector Settings

Under JDBC Connector Settings, parameter settings shown in Figure 114 and described below ensure there are sufficient connections available for the Oracle eWay.

Figure 114 Performance Settings for Oracle Externals: JDBC Connector Settings



Description (MaxIdleTime)

The maximum number of seconds that a physical connection may remain unused before it is closed. 0 (zero) indicates that there is no limit.

Comments (MaxIdleTime)

Description	Oracle Connection Pool Datasource
ServerName	<hostname_of_Oracle_server_machine>
PortNumber	1521
DatabaseName	<SID_(name-of-database_for_eXchange)>
User	<username1_for_eXchange_database>
Password	********
DriverProperties	
Delimiter	#
TNSEntry	
MinPoolSize	0
MaxPoolSize	200
MaxIdleTime	2

MaxPoolSize: 200

Increase the setting from the default (10) to a value that corresponds to the number of physical connections expected.

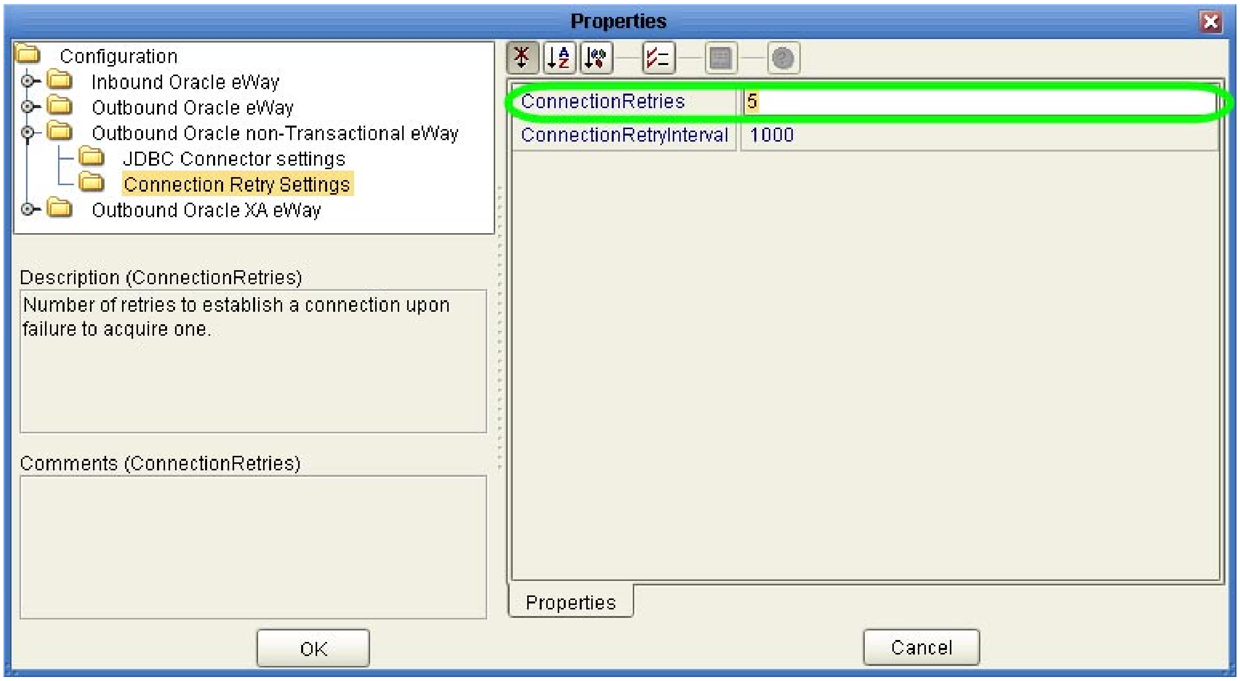
MaxIdleTime: 2

Change the setting from the default (0, signifying no upper limit) to a value that forces unused connections to be closed when idle for along time.

Connection Retry Settings

Under Connection Retry Settings, parameter settings shown in Figure 115 and explained below ensure there are sufficient connections available for the Oracle eWay.

Figure 115 Performance Settings for Oracle Externals: Connection Retry Settings



Description (ConnectionRetries)
Number of retries to establish a connection upon failure to acquire one.

Comments (ConnectionRetries)

ConnectionRetries	5
ConnectionRetryInterval	1000

ConnectionRetries: 5

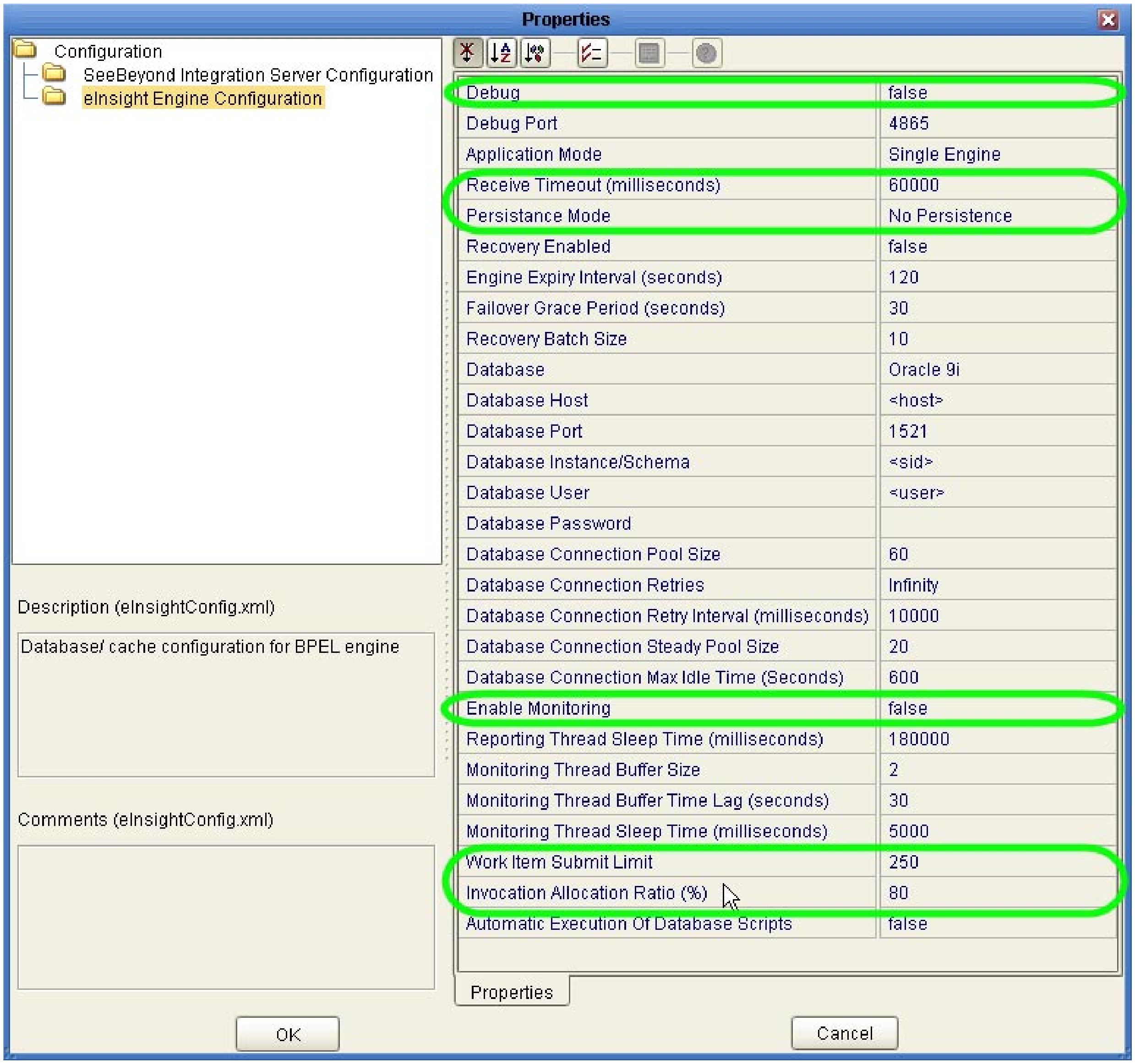
Increase the setting from default (0) to a value that ensures connections are not dropped prematurely.

Environment Settings for Integration Server: eInsight Engine

Settings for Sun SeeBeyond Integration Server, eInsight Engine Configuration

The following properties for the Integration Server are set under **Configuration > eInsight Engine Configuration**, as shown in Figure 116 and described below.

Figure 116 Performance Settings for Oracle Externals: JDBC Connector Settings



Properties
Configuration
SeeBeyond Integration Server Configuration
eInsight Engine Configuration
Properties
OK
Cancel

Description (eInsightConfig.xml)

Database/ cache configuration for BPEL engine

Comments (eInsightConfig.xml)

Debug	false
Debug Port	4865
Application Mode	Single Engine
Receive Timeout (milliseconds)	60000
Persistance Mode	No Persistence
Recovery Enabled	false
Engine Expiry Interval (seconds)	120
Failover Grace Period (seconds)	30
Recovery Batch Size	10
Database	Oracle 9i
Database Host	<host>
Database Port	1521
Database Instance/Schema	<sid>
Database User	<user>
Database Password	
Database Connection Pool Size	60
Database Connection Retries	Infinity
Database Connection Retry Interval (milliseconds)	10000
Database Connection Steady Pool Size	20
Database Connection Max Idle Time (Seconds)	600
Enable Monitoring	false
Reporting Thread Sleep Time (milliseconds)	180000
Monitoring Thread Buffer Size	2
Monitoring Thread Buffer Time Lag (seconds)	30
Monitoring Thread Sleep Time (milliseconds)	5000
Work Item Submit Limit	250
Invocation Allocation Ratio (%)	80
Automatic Execution Of Database Scripts	false

Debug: false

Retain the default setting (false) to avoid the performance degradation that occurs when the BPEL debugger is attached.

Receive Timeout: 60000

Increase the setting from the default (15000 milliseconds) to a value that prevents the eInsight engine from timing out if there are not enough instances left to start a BP. If messages are rolled back because of the time-outs to create BP instances, consider increasing this value even further.

Explanation: By default provided with the B2B Suite, the maximum number of concurrent BP instances is 5. But if all five instances of a BP are in use and more messages are available to process, the eInsight engine can time out and roll back messages. Better performance is achieved by holding messages in memory and not allowing the eInsight engine to time out.

Persistence Mode: No Persistence

Retain the default setting (No Persistence). Running with Persistence enabled incurs a significant performance hit.

Enable Monitoring: false

Retain the default setting (false), because monitoring requires persistence mode.

Work Item Submit Limit: 250

Decrease the setting from the default (500) to a value that is close to the Application Server thread pool size, whose default value is 200. Adjust this up or down from 250 only if the system has such problems as eInsight engine freeze, causing BPs to not run to completion. Do not adjust it in an effort to change BP performance.

Explanation: The value 250 is recommended because it is slightly larger than the Application Server thread pool size default value (200). If you increase the Work Item Submit Limit, increase Application Server thread pool size correspondingly.

Invocation Allocation Ratio

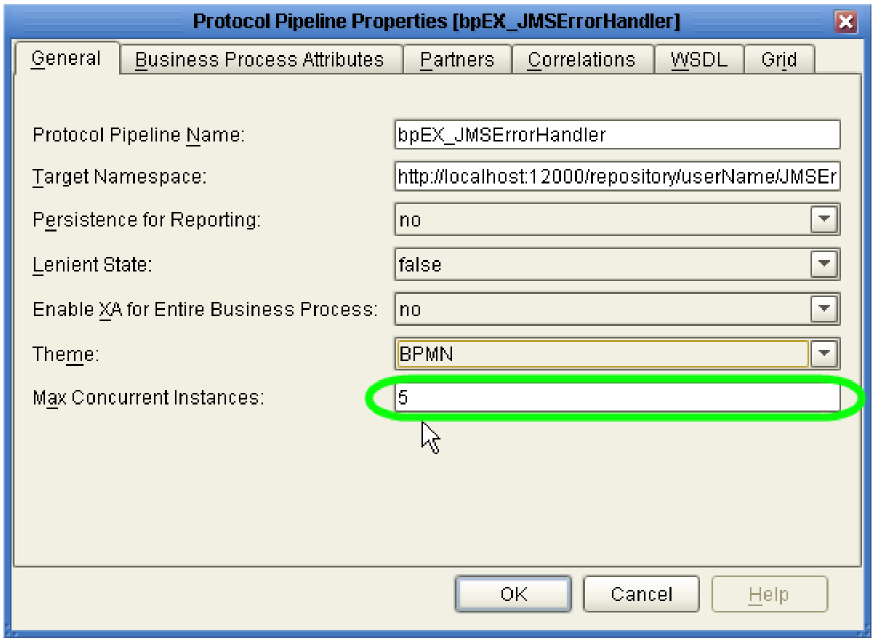
Retain the default setting (80 percent). Adjust this up or down from 80 only if the system has such problems as eInsight engine freeze, causing BPs to not run to completion. Do not adjust this in an effort to change BP performance.

Explanation: When applied to the recommended value for Work Item Submit Limit, this yields a request of no more than 200 threads (80% of 250) for processing invoked work items. “Invoked work items” include activities such as unmarshaling an OTD, calling a JCD, or calling another BP. “Non-invoked work items” include activities such as Assigns in a BP, JMS.receive, and a Wait (timer).

Project Settings for Components: BPs

The following properties for BPs (Business Processes and B2B Protocol Pipelines) are set in the **General** tab, as shown in Figure 117 and described below.

Figure 117 Performance Settings for BPs



Protocol Pipeline Properties [bpEX_JMSErrorHandler]

Protocol Pipeline Name: bpEX_JMSErrorHandler
Target Namespace: http://localhost:12000/repository/userName/JMSEr
Persistence for Reporting: no
Lenient State: false
Enable XA for Entire Business Process: no
Theme: BPMN
Max Concurrent Instances: 5

Max Concurrent Instances: 5

Retain the default setting (5).

Explanation: The Max Concurrent Instances parameter acts as a throttle on the number of instances of a particular BP. If it is set too high, extreme memory paging can occur, with a drastic drop in performance. This parameter also has the single most influence on the amount of memory used.

All BP code shipped with the B2B suite (except for the protocol-specific validations) has this value preset to 5. Testing has shown that this setting strikes a good balance between memory usage and performance, and we recommend that for any BP you create yourself, use this value as a starting point.

We also recommend that you update this value to 5 for any protocol-specific validations you use. Although there is no deterministic algorithm for calculating the best value of this parameter, tests showed that it worked well for handling large volumes of messages at a time. If log file messages indicate you are running out of concurrent instances, consider changing this default to a higher value. However, because increasing this value causes more threads to be created, higher values tend to decrease performance.

D.3.2 Run-Time Parameters (Enterprise Manager)

- In the Sun SeeBeyond Integration Server Security Gateway, change all Module Log Levels to SEVERE. This reduces the amount of trace that is printed to the server.log file, thus improving throughput.

- Give the Java Virtual Machine (JVM) at least 2 GB of RAM. (In other words: Use the **-Xmx2048M** setting.)
- Make sure JPDA debugging is turned off in domain.xml. (In other words: Use the **debug=false** setting). Enabling debugging can cause a significant drag on the server, impacting its performance and potentially impairing its reliability.

D.4 Results

The system ran with no interruption for 4.5 days, having processed 1.2 million messages in the end.

D.5 Lessons Learned

- To achieve the goal, it was essential to properly configure many options within various parts of the B2B Suite and Java CAPS. When tuning a system, it is vital to have good understanding of the underlying options in the Java CAPS components.
- Plan to have sufficient space on the machine. The ***.dbs** files—located in **logicalhost/is/domains/<*domainName*>/stcms/instance/**—can grow without limit the system becomes backlogged in the course of a long run. However, with proper configuration, no substantial sustained growth occurs. For guidance, consult the appropriate Java CAPS documentation, especially the *eGate Integrator JMS Reference*.

Appendix E

Performance

This appendix is a modified version of a white paper published previously as "Sun B2B Suite 2.0 Performance: Optimizing Throughput".

What's in This Appendix

E.1 Abstract and Summary

This appendix discusses how to optimize throughput for Sun B2B Suite 2.0 using the ASC X12 Protocol Manager. The optimization detailed here does not include scaling, which can result in further performance gains. The appendix includes details on the test environment used, the throughput achieved, and the settings used to achieve this performance.

You can use this appendix to help configure a system to achieve maximal throughput for Sun B2B Suite 2.0.

Summary of Findings

At its first release (on Java CAPS 5.1.2), Sun B2B Suite 2.0 has been verified to have a throughput of at least 3 transactions per second (TPS). The test case distinguishes two scenarios that were measured independently: *inbound* (receiving messages from a trading partner), and *outbound* (sending messages to a trading partner.

To achieve best throughput performance, it was important to understand various settings in Enterprise Designer that are supplied by the Java Composite Application Platform Suite (Java CAPS). Many of these settings concerned the proper configuration of the eInsight engine.

This white paper details the test environment and the settings used to achieve the 3 TPS throughput. Its intent is to serve as a guide to duplicating these results.

E.2 Testing

The Sun B2B Suite ASC X12 Protocol Manager was selected for measuring throughput with a single trading partner (TP). One thousand X12 810 messages with a payload of 947 bytes were sent at a time and transported using FTP to the same system. The system was configured to batch up 100 messages at a time, and these same batches were used to simulate receiving message from a TP.

The choice of protocol, payload, and batch size were based on data provided by a Sun B2B customer. They had been used for testing with a previous version of eXchange Integrator and thus provided a good benchmark to verify the performance improvements.

For reproducibility and measurement precision, the inbound and outbound test cases were run independently, and thus without processing acknowledgments. The criterion for success was defined as at least 3 TPS on each scenario.

E.2.1 The Test Environment

All tests were run on a server dedicated to the minimum set of applications: Sun B2B Suite on Java CAPS (one Logical Host domain, as well as Enterprise Manager - Monitor and Runtime Administrator), and a utility (**top**) for measuring performance.

Hardware

Applications ran on a Sun Fire™ V440 with four SPARC® CPUs running Solaris 10 with 8 GB of memory.

Database and LDAP Servers

Oracle 9i was set up on a desktop machine accessed via the local network. No special database tuning was done for the tests.

Sun Java System Directory Server 5.2 was used for LDAP support. For performance considerations, the Sun LDAP server is highly recommended over OpenLDAP.

Logging Levels

All tests were run with the Module Log Level set to SEVERE for all individual server modules. (This is neither the default nor the recommended setting; it was used to minimize spent writing to log files.)

E.2.2 Methodology for Measuring Throughput

Trace logging statements at the SEVERE level were added to the BP code to measure the time of the complete run. To calculate TPS, elapsed time was converted to seconds.

Outbound Scenario

The elapsed time was calculated by subtracting the time when the first message entered the eXchange system (that is, when it was read from bpEX_MainFromInternal) from the time when the final message was processed by the delivery channel (that is, when it left bpEX_DeliveryChannel_FTP).

Inbound Scenario

The elapsed time was calculated by subtracting the time when the first batch of messages was picked up by the eXchange (that is, when it was read from bpEX_MainFromTP) from the time when the final message was sent to the internal system (that is, when it left bpEX_Unbatcher_X12).

E.3 Important Settings

Throughput of 3 TPS was achieved by customizing several parameter settings used by the Sun B2B Suite. These included settings for the Oracle eWay, the Integration Server properties for the eInsight engine, and the level of logging used by the system. Some of these settings are presupplied with the as-shipped products; others must be manually configured at design time. Many of the same settings are also needed to ensure reliability. (For details, refer to **Appendix D "Reliability" on page 209**.)

Note: *The parameter settings in this section should be considered a good starting point for performance tuning. For each particular operating environment, parameter settings are likely to require adjustment.*

E.3.1 Design-Time Parameters (Enterprise Designer)

Design-time parameters affecting performance are found both in the Environment (properties of Oracle external systems and, within each Logical Host, properties of Integration servers) and in each Project (properties of BPs).

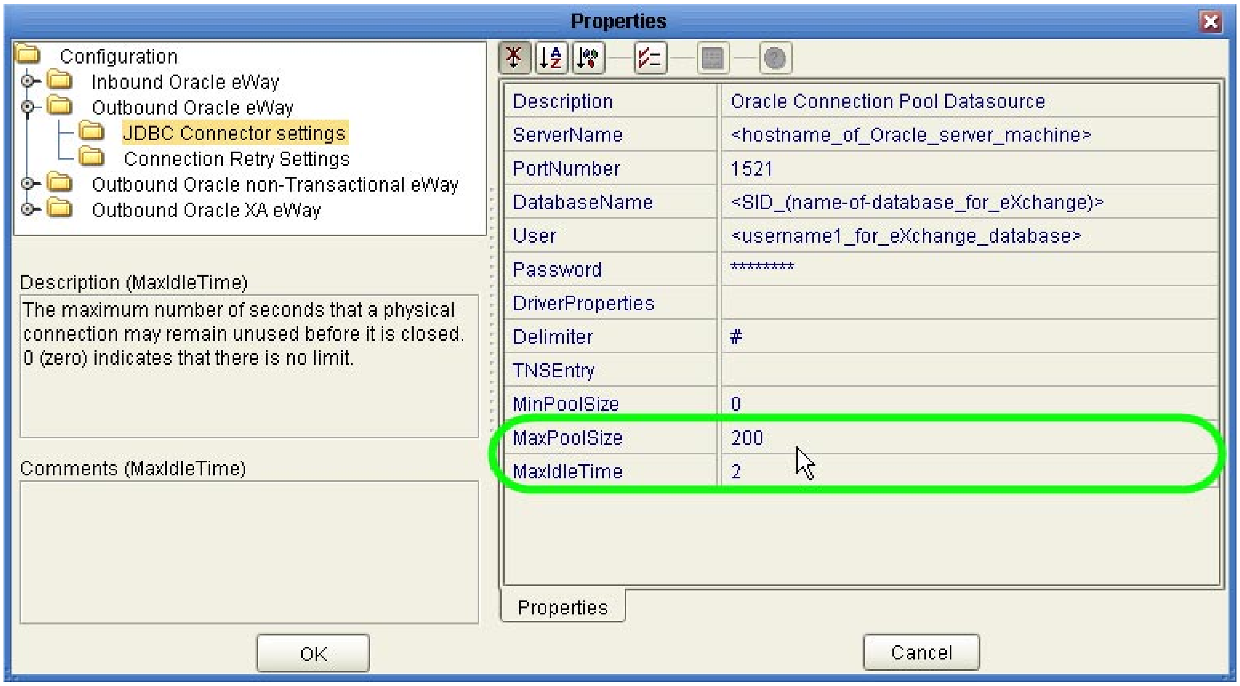
Environment Settings for External Systems: Oracle

The following properties for Oracle externals are set under **Configuration > Outbound Oracle eWay** and **Outbound Oracle non-Transactional eWay**. Parameter settings should be adjusted whenever a system does not have enough connections available to the Oracle database, or if the system times out on connecting to Oracle when the database is up and running.

JDBC Connector Settings

Under JDBC Connector Settings, parameter settings shown in Figure 118 and described below ensure there are sufficient connections available for the Oracle eWay.

Figure 118 Performance Settings for Oracle Externals: JDBC Connector Settings



Description (MaxIdleTime)

The maximum number of seconds that a physical connection may remain unused before it is closed. 0 (zero) indicates that there is no limit.

Comments (MaxIdleTime)

Description	Oracle Connection Pool Datasource
ServerName	<hostname_of_Oracle_server_machine>
PortNumber	1521
DatabaseName	<SID_(name-of-database_for_eXchange)>
User	<username1_for_eXchange_database>
Password	********
DriverProperties	
Delimiter	#
TNSEntry	
MinPoolSize	0
MaxPoolSize	200
MaxIdleTime	2

MaxPoolSize: 200

Increase the setting from the default (10) to a value that corresponds to the number of physical connections expected.

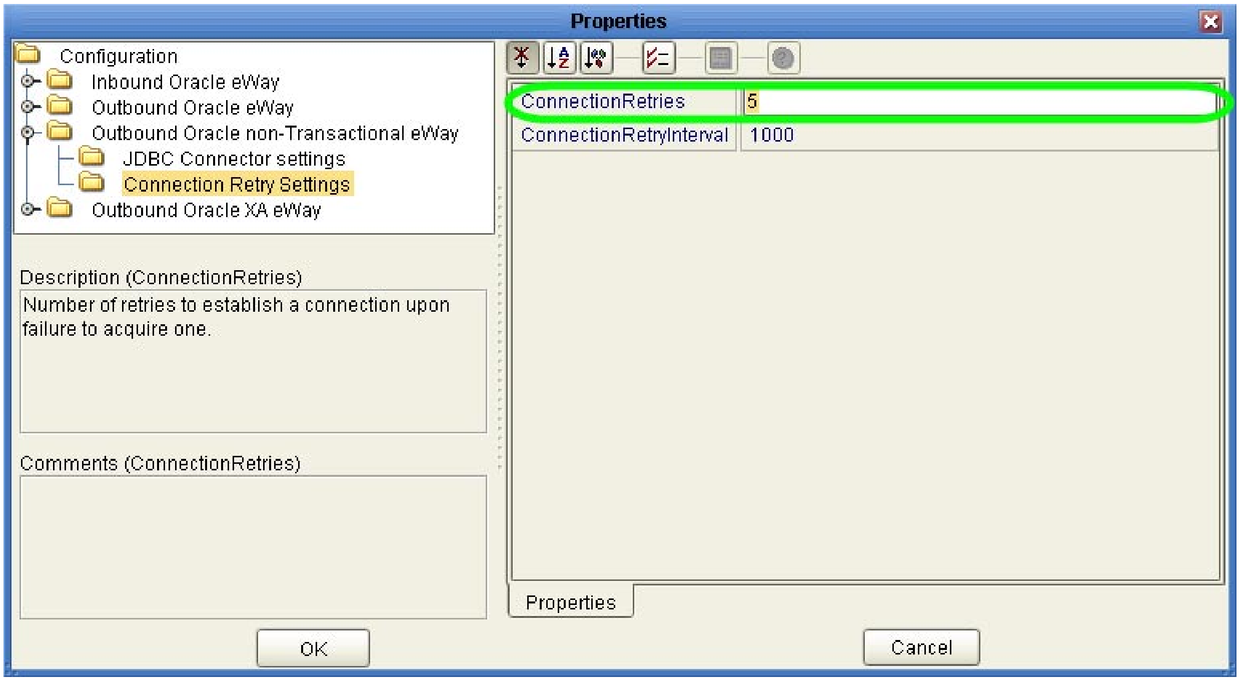
MaxIdleTime: 2

Change the setting from the default (0, signifying no upper limit) to a value that forces unused connections to be closed when idle for along time.

Connection Retry Settings

Under Connection Retry Settings, parameter settings shown in Figure 119 and explained below ensure there are sufficient connections available for the Oracle eWay.

Figure 119 Performance Settings for Oracle Externals: Connection Retry Settings



Properties

Configuration
- Inbound Oracle eWay
- Outbound Oracle eWay
- Outbound Oracle non-Transactional eWay
  - JDBC Connector settings
  - Connection Retry Settings
- Outbound Oracle XA eWay

Description (ConnectionRetries)

Number of retries to establish a connection upon failure to acquire one.

Comments (ConnectionRetries)

ConnectionRetries	5
ConnectionRetryInterval	1000

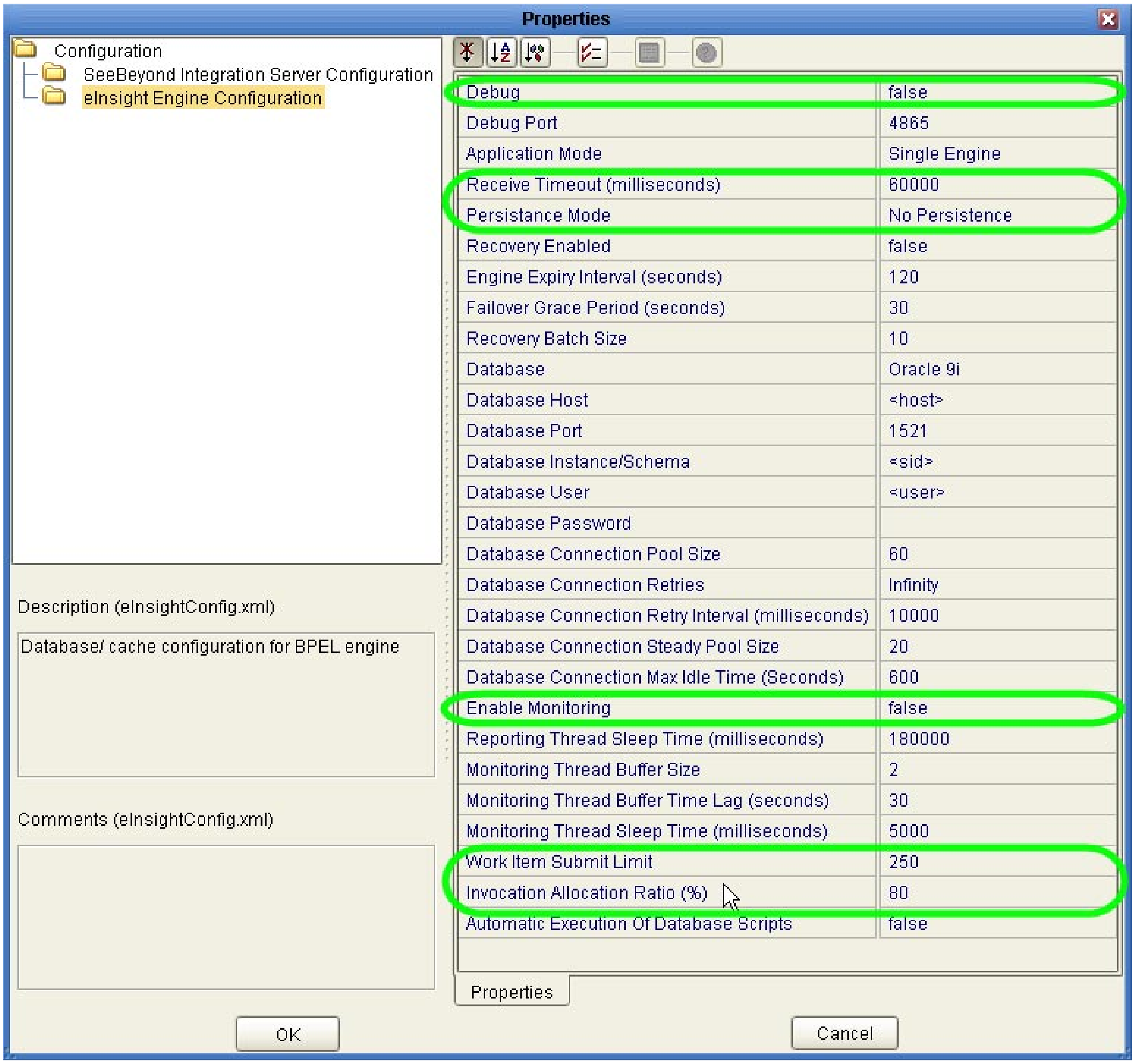
ConnectionRetries: 5

Increase the setting from default (0) to a value that ensures connections are not dropped prematurely.

Environment Settings for Integration Server: eInsight Engine

The following properties for the Integration Server are set under **Configuration > eInsight Engine Configuration**, as shown in Figure 120 and described below.

Figure 120 Performance Settings for Oracle Externals: JDBC Connector Settings



Properties
Configuration
SeeBeyond Integration Server Configuration
eInsight Engine Configuration
OK
Cancel

Description (eInsightConfig.xml)

Database/ cache configuration for BPEL engine

Comments (eInsightConfig.xml)

Properties	
Debug	false
Debug Port	4865
Application Mode	Single Engine
Receive Timeout (milliseconds)	60000
Persistance Mode	No Persistence
Recovery Enabled	false
Engine Expiry Interval (seconds)	120
Failover Grace Period (seconds)	30
Recovery Batch Size	10
Database	Oracle 9i
Database Host	<host>
Database Port	1521
Database Instance/Schema	<sid>
Database User	<user>
Database Password	
Database Connection Pool Size	60
Database Connection Retries	Infinity
Database Connection Retry Interval (milliseconds)	10000
Database Connection Steady Pool Size	20
Database Connection Max Idle Time (Seconds)	600
Enable Monitoring	false
Reporting Thread Sleep Time (milliseconds)	180000
Monitoring Thread Buffer Size	2
Monitoring Thread Buffer Time Lag (seconds)	30
Monitoring Thread Sleep Time (milliseconds)	5000
Work Item Submit Limit	250
Invocation Allocation Ratio (%)	80
Automatic Execution Of Database Scripts	false

Debug: false

Retain the default setting (false) to avoid the performance degradation that occurs when the BPEL debugger is attached.

Receive Timeout: 60000

Increase the setting from the default (15000 milliseconds) to a value that prevents the eInsight engine from timing out if there are not enough instances left to start a BP. If messages are rolled back because of the time-outs to create BP instances, consider increasing this value even further.

Explanation: By default provided with the B2B Suite, the maximum number of concurrent BP instances is 5. But if all five instances of a BP are in use and more messages are available to process, the eInsight engine can time out and roll back messages. Better performance is achieved by holding messages in memory and not allowing the eInsight engine to time out.

Persistence Mode: No Persistence

Retain the default setting (No Persistence). Running with Persistence enabled incurs a significant performance hit.

Enable Monitoring: false

Retain the default setting (false), because monitoring requires persistence mode.

Work Item Submit Limit: 250

Decrease the setting from the default (500) to a value that is close to the Application Server thread pool size, whose default value is 200. Adjust this up or down from 250 only if the system has such problems as eInsight engine freeze, causing BPs to not run to completion. Do not adjust it in an effort to change BP performance.

Explanation: The value 250 is recommended because it is slightly larger than the Application Server thread pool size default value (200). If you increase the Work Item Submit Limit, increase Application Server thread pool size correspondingly.

Invocation Allocation Ratio

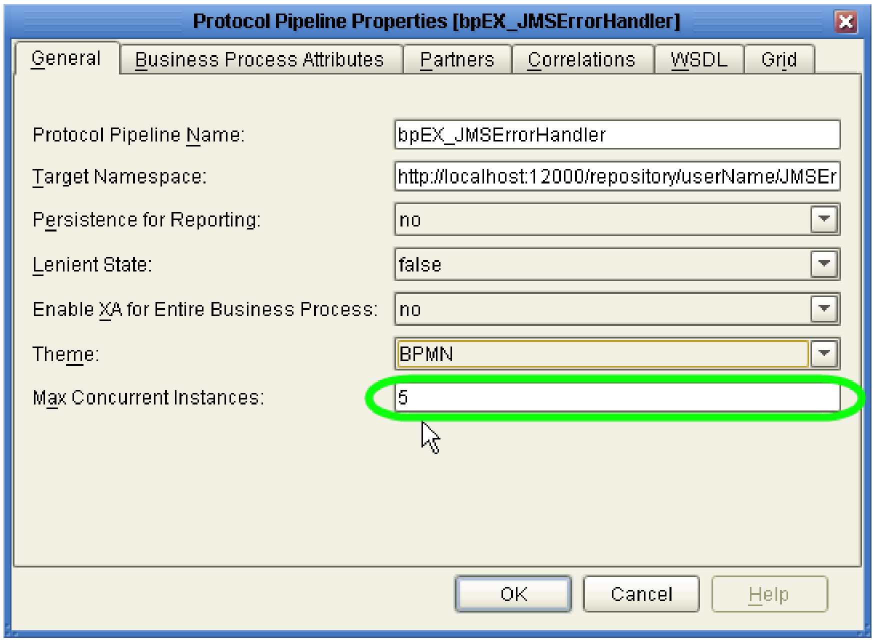
Retain the default setting (80 percent). Adjust this up or down from 80 only if the system has such problems as eInsight engine freeze, causing BPs to not run to completion. Do not adjust this in an effort to change BP performance.

Explanation: When applied to the recommended value for Work Item Submit Limit, this yields a request of no more than 200 threads (80% of 250) for processing invoked work items. “Invoked work items” include activities such as unmarshaling an OTD, calling a JCD, or calling another BP. “Non-invoked work items” include activities such as Assigns in a BP, JMS.receive, and a Wait (timer).

Project Settings for Components: BPs

The following properties for BPs (Business Processes and B2B Protocol Pipelines) are set in the **General** tab, as shown in Figure 121 and described below.

Figure 121 Performance Settings for BPs



Protocol Pipeline Properties [bpEX_JMSErrorHandler]
General
Business Process Attributes
Partners
Correlations
WSDL
Grid
OK
Cancel
Help

Protocol Pipeline Name: bpEX_JMSErrorHandler

Target Namespace: http://localhost:12000/repository/userName/JMSEr

Persistence for Reporting: no

Lenient State: false

Enable XA for Entire Business Process: no

Theme: BPMN

Max Concurrent Instances: 5

Max Concurrent Instances: 5

Retain the default setting (5).

Explanation: The Max Concurrent Instances parameter acts as a throttle on the number of instances of a particular BP. If it is set too high, extreme memory paging can occur, with a drastic drop in performance. This parameter also has the single most influence on the amount of memory used.

All BP code shipped with the B2B suite (except for the protocol-specific validations) has this value preset to 5. Testing has shown that this setting strikes a good balance between memory usage and performance, and we recommend that for any BP you create yourself, use this value as a starting point.

We also recommend that you update this value to 5 for any protocol-specific validations you use. Although there is no deterministic algorithm for calculating the best value of this parameter, tests showed that it worked well for handling large volumes of messages at a time. If log file messages indicate you are running out of concurrent instances, consider changing this default to a higher value. However, because increasing this value causes more threads to be created, higher values tend to decrease performance.

E.3.2 Run-Time Parameters (Enterprise Manager)

In the Sun SeeBeyond Integration Server Security Gateway, change all Module Log Levels to SEVERE. This reduces the amount of trace that is printed to the server.log file, thus improving throughput.

E.4 Conclusions

In the specified test environment, with proper configuration of key design-time and run-time parameters of components in Java CAPS and the B2B Suite, both scenarios (outbound and inbound) achieved throughput of 3 transactions per second or greater.

Index