



Sun B2B Suite eXchange Integrator User's Guide



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 820-1282-10
December 2007

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and SunTM Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

Preface	9
1 Overview of eXchange Integrator	15
What's in This Chapter	15
Summary of Features	15
eXchange Integrator and Java CAPS	16
Integration with Java CAPS	16
Architectural Overview	17
Process Overview	22
Design Phase: Using Enterprise Designer	23
Design Phase: Using ePM	23
Runtime Phase	24
Prerequisites Checklist	24
▼ One-Time Setup Tasks	24
▼ Enterprise Designer Tasks	24
▼ ePM Tasks	25
▼ Run-Time Tasks	25
2 Installing eXchange Integrator	27
What's in This Chapter	27
Supported Operating Systems	27
Supported External Applications	28
Database for Message Tracker	28
Database for Persistence and Monitoring via eInsight Engine	28
Installing the Product Files	28
Uploading B2B Suite Product Files to the Repository	29
Refreshing Enterprise Designer	34

After You Install	37
LDAP Server	38
About LDAP, Directory Services, Servers, and Clients	38
Database Scripts	40
Creating and Configuring the eXchange Integrator Database Instance	40
Extracting, Customizing, and Running Database Setup Scripts	43
3 eXchange Integrator Features	49
What's in This Chapter	49
Project Tree Organization	49
Transport Attribute Definitions	50
Overview	51
BatchFTP	52
BatchLocalFile	54
FILE	55
FTP	56
HTTP	56
HTTPS	57
JMS	57
SMTP	57
Channel Manager	58
In this section	59
Message Tracker	64
B2B Protocols for X12, HIPAA, EDIFACT, AS2, and ebXML	65
Handling Errors	65
Message Tracker	65
eXchange Integrator Standard Error-Handling Topics	65
Error Logs	66
Creating a Nondefault Error Handler	66
4 Using eXchange Integrator in Enterprise Designer	71
What's in This Chapter	71
Overview	71
Transaction Profiles	72
Using Enterprise Designer	72

B2B Host Procedures	73
Setting Up a B2B Host and Its Components	74
▼ To create a B2B host	74
To populate a B2B host with attribute definitions	75
Setting Up Attribute Definitions	77
Creating and Configuring TADs	77
Creating and Configuring DPADs	81
Creating and Configuring BPADs	83
Building a B2B Host	88
Creating an Environment	89
Optional: Configuring an Environment to Use HTTPS	92
Connecting the B2B Host to Oracle and LDAP Externals	94
Building the B2B Host	96
Building and Deploying the GUI Projects	99
▼ Before you begin	100
▼ To build and deploy the Deployment Profile for the ePM Project	100
▼ To build and deploy the Deployment Profile for the Tracker Project	102
Building and Deploying the Error-Handling Projects	103
▼ Before you begin	103
▼ To build and deploy the Deployment Profile for the Sub_DLQ Project	103
▼ To build and deploy the Deployment Profile for the Sub_ProcErrors Project	105
5 Using B2B Web Facilities	107
What's in This Chapter	107
General Topics	107
Topics Specific to eXchange Partner Manager (ePM)	107
Topics Specific to Message Tracker	108
Providing Access to B2B Web Facilities	108
▼ To add or set up user access to B2B web facilities	108
eXchange Partner Manager (ePM) Overview	112
Operational Overview of the ePM GUI	112
Structural Overview of the ePM GUI	114
Basic Operations in ePM	115
Accessing ePM	115
Viewing and Setting Business Protocol Parameters	122

Configuring Business Action Groups and Business Actions	123
Protocol-Outbound (ToPartner) Parameter and ACK Settings	126
Protocol-Inbound (FromPartner) Parameter and ACK Settings	130
Batch-Outbound (ToPartner) Parameter and ACK Settings	133
Viewing and Setting Delivery Protocol Parameters	136
Configuring Delivery Action Groups and Delivery Actions	137
Parameter and ACK Settings for Delivery Protocols	140
Creating and Configuring Transaction Profiles	140
▼ To create a new transaction profile	141
Inheriting and Overriding Parameter Settings	144
Quick Summary of Inheritance	144
Inheritance and Overrides in B2B Host Configuration	144
Inheritance and Overrides in Trading Partner Configuration	145
Example of Multiple Inheritance and Overrides	145
Creating and Configuring Trading Partners	153
Basic Operations for Trading Partners	154
Configuring Trading Partners	158
Message Tracker	163
Accessing Message Tracker	163
Using Message Tracker	164
6 Designing Business Processes	167
What's in This Chapter	167
Using the eXchange Protocol Designer GUI	167
Modeling Elements	168
Activity Elements	168
Validating a Business Process	173
Saving an Unfinished Business Process	173
Using Business Processes in a Connectivity Map	174
▼ To include a Business Process as a service on a connectivity map	174
▼ To connect the Business Process activities to the externals	175
7 Configuring Exception Handling	177
What's in This Chapter	177
Overview	177

Exception Handling Configuration	178
Scope	179
Scope or Process-level exceptions	179
Compensation	180
Using Scope and Exceptions to Trigger Compensation	180
Validating the B2B Protocol	180
8 eXchange Troubleshooting Tips	181
What's in This Chapter	181
Identifying and Resolving Problems	181
Performance Considerations	185
Increasing the Logical Host Heap Size	186
Increasing eInsight Engine Performance	187
Increasing the JMS IQ Manager Segment Size Property	188
Increasing the Oracle Number of Processes	189
Glossary	191
Index	195

Preface

The Sun B2B Suite eXchange Integrator User's Guide provides instructions and background information for all users of eXchange Integrator. This guide is designed for managers, system administrators, and others who use eXchange Integrator to create, implement, and maintain B2B business protocols and delivery protocols for the exchange of business data between Trading Partners.

The purpose of this guide is to help you to:

- Understand the basic architecture and functionality of eXchange Integrator
- Understand the relationship of eXchange Integrator to other components in the B2B Suite and Java CAPS
- Learn about the eXchange Integrator components and editors and how to use them in your environment

Who Should Use This Book

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.

Before You Read This Book

Before you try to understand the concepts presented in this book and begin using the reference material it presents, make sure you are familiar with the references listed under . You must be especially proficient in the use of eGate Integrator™, eWay Adapters™, and eInsight Business Process manager™.

How This Book Is Organized

This guide is organized into the following chapters:

- [Chapter 1, “Overview of eXchange Integrator,”](#) discusses general features and architecture of eXchange Integrator.
- [Chapter 2, “Installing eXchange Integrator,”](#) provides step-by-step instructions for installing eXchange Integrator and setting it up for use.
- [Chapter 3, “eXchange Integrator Features,”](#) highlights and discusses the key features and components supplied with eXchange Integrator.
- [Chapter 4, “Using eXchange Integrator in Enterprise Designer,”](#) provides step-by-step procedures for working with eXchange Integrator at design time and deploying projects.
- [Chapter 5, “Using B2B Web Facilities,”](#) provides step-by-step procedures for working with eXchange Integrator’s Web-based GUIs: eXchange Partner Manager (ePM) and Message Tracker.
- [Chapter 6, “Designing Business Processes,”](#) provides step-by-step procedures for designing and deploying business processes using eXchange Protocol Designer.
- [Chapter 7, “Configuring Exception Handling,”](#) explains the use of business processes for handling exceptions.
- [Chapter 8, “eXchange Troubleshooting Tips”](#) provides troubleshooting and tuning information for and suggests diagnoses and remedies.
- The [Glossary](#) lists and explains special acronyms and initialisms that occur in this guide.

Related Books

For late-breaking information on the B2B Suite, refer to the *Sun B2B Suite Release Notes* available in HTML and PDF formats on <http://docs.sun.com>. 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:

- *Java CAPS Installation Guide*
- *Java CAPS Deployment Guide*
- *eGate Integrator User’s Guide*
- *eGate Integrator System Administration Guide*
- *eGate Integrator JMS Reference*
- *eInsight Business Process Manager User’s Guide*

- *eWay Batch Adapter User's Guide*
- *eWay HTTPS Adapter User's Guide*
- *eWay File Adapter User's Guide*
- *eWay LDAP Adapter User's Guide*
- *eWay Adapter for Oracle User's Guide*
- *ASC X12 Protocol Manager User's Guide*
- *ASC X12 OTD Library User's Guide*
- *HIPAA Protocol Manager User's Guide*
- *HIPAA OTD Library User's Guide*

Additional References and Recommended Reading

Additional documentation is available through various media.

- For white papers on scalability, reliability, performance and other topics, such as importing and exporting Host, Trading Partner, and Scheduler via command line, go to <http://jcaps-b2b.dev.java.net>
- For installation and configuration instructions for LDAP, go to http://www.sun.com/software/products/directory_srvr_ee/get1.jsp
- For information on eInsight Business Process Manager, see the *eInsight Business Process Manager User's Guide*.
- For information on Enterprise Manager, see the *eGate Integrator User's Guide*.

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (<http://www.sun.com/documentation/>)
- Support (<http://www.sun.com/support/>)
- Training (<http://www.sun.com/training/>)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your .login file. Use <code>ls -a</code> to list all files. machine_name% you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	machine_name% su Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <i>rm filename</i> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in this guide. A <i>cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell	machine_name%
C shell for superuser	machine_name#
Bourne shell and Korn shell	\$
Bourne shell and Korn shell for superuser	#

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>

Overview of eXchange Integrator

This chapter provides a general overview of eXchange Integrator 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

This chapter covers the following topics:

- [“Summary of Features” on page 15](#)
- [“eXchange Integrator and Java CAPS” on page 16](#)
- [“Architectural Overview” on page 17](#)
- [“Process Overview” on page 22](#)
- [“Prerequisites Checklist” on page 24](#)

Summary of Features

eXchange Integrator 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 non-repudiation.

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 Integrator 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.

eXchange Integrator and Java CAPS

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

Integration with Java CAPS

eXchange Integrator is tightly integrated with Java CAPS and runs as a component within the Java CAPS environment. [Figure 1–1](#) shows how eXchange Integrator and other Java CAPS components work together.

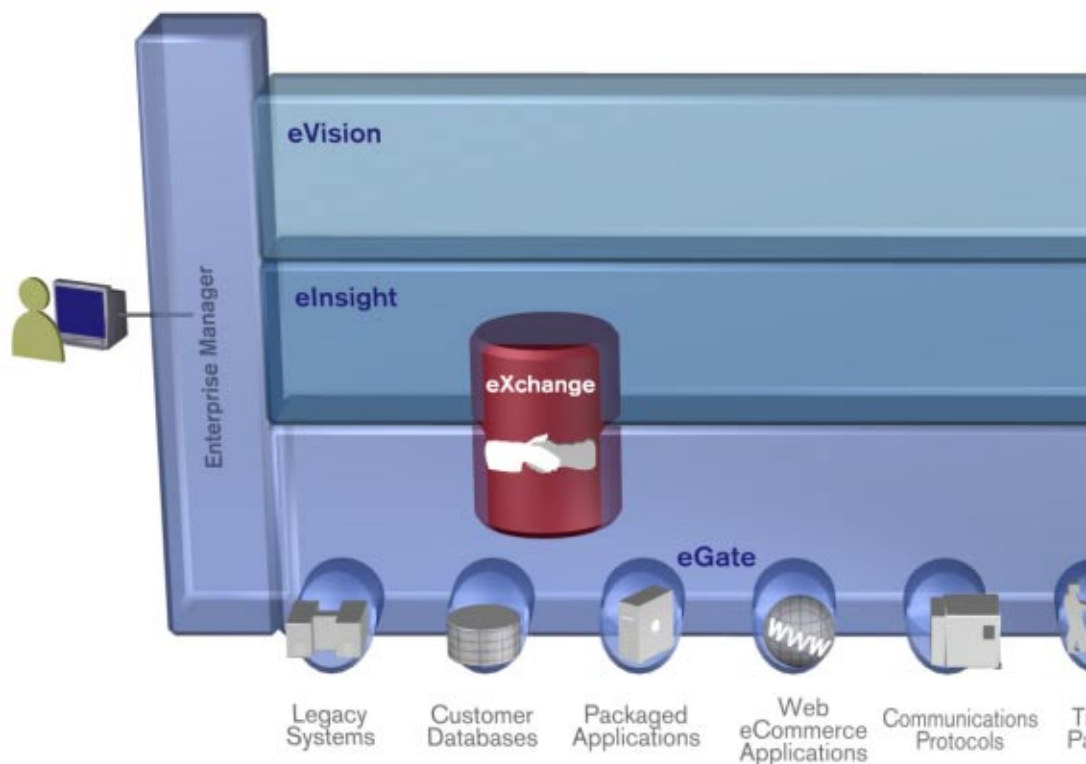


FIGURE 1-1 eXchange Integrator and the Java CAPS Framework

Architectural Overview

eXchange Integrator 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 Integrator uses the following key components:

- B2B Host Designer and attribute definitions— Using the Enterprise Designer GUI framework, eXchange Integrator 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. See [Figure 1-2](#).
- Business Services (BPs, JCDs, OTDs)—eXchange Integrator 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 Integrator, 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 Integrator 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 1-5](#).
- LDAP Server—eXchange Integrator uses an LDAP-compliant directory server to mediate retrieval of trading partner information.
- eXchange Integrator Database—eXchange Integrator uses an Oracle database to store run-time information on correlation and message tracking.
- Message Tracker — eXchange Integrator provides a Web-based message tracking GUI with powerful filtering and searching capabilities. See [Figure 1-6](#).

The illustrations in [Figure 1-2](#), [Figure 1-3](#), and [Figure 1-4](#) show some of the features provided by the various GUIs.

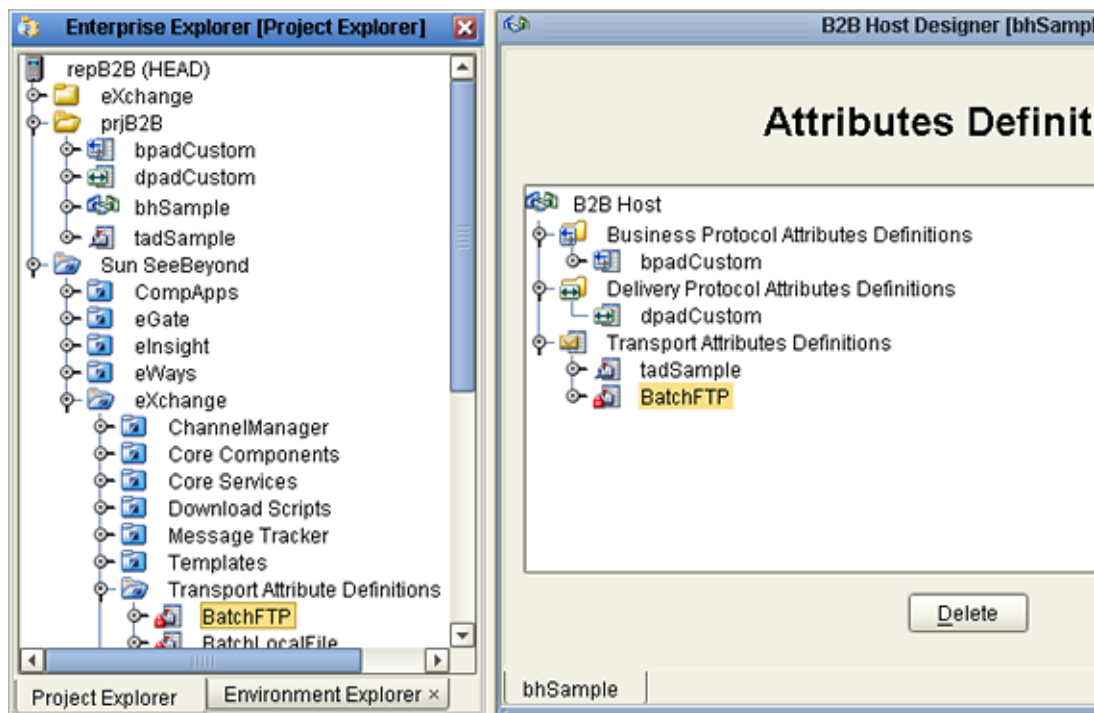


FIGURE 1-2 B2B Host Designer in Enterprise Designer

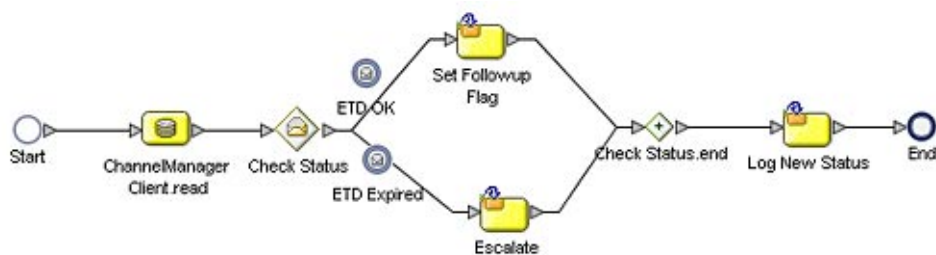


FIGURE 1-3 Example of a User-Created Business Process

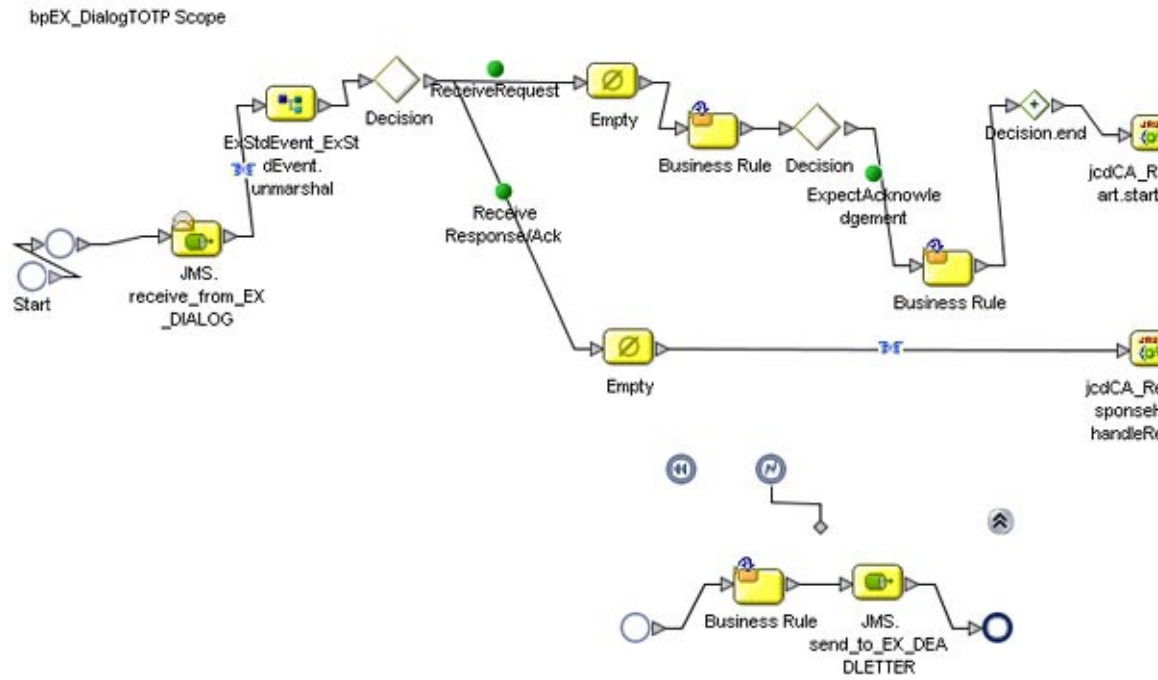


FIGURE 1-4 Example of a Prebuilt Business Process (bpEX_DialogTOTP)

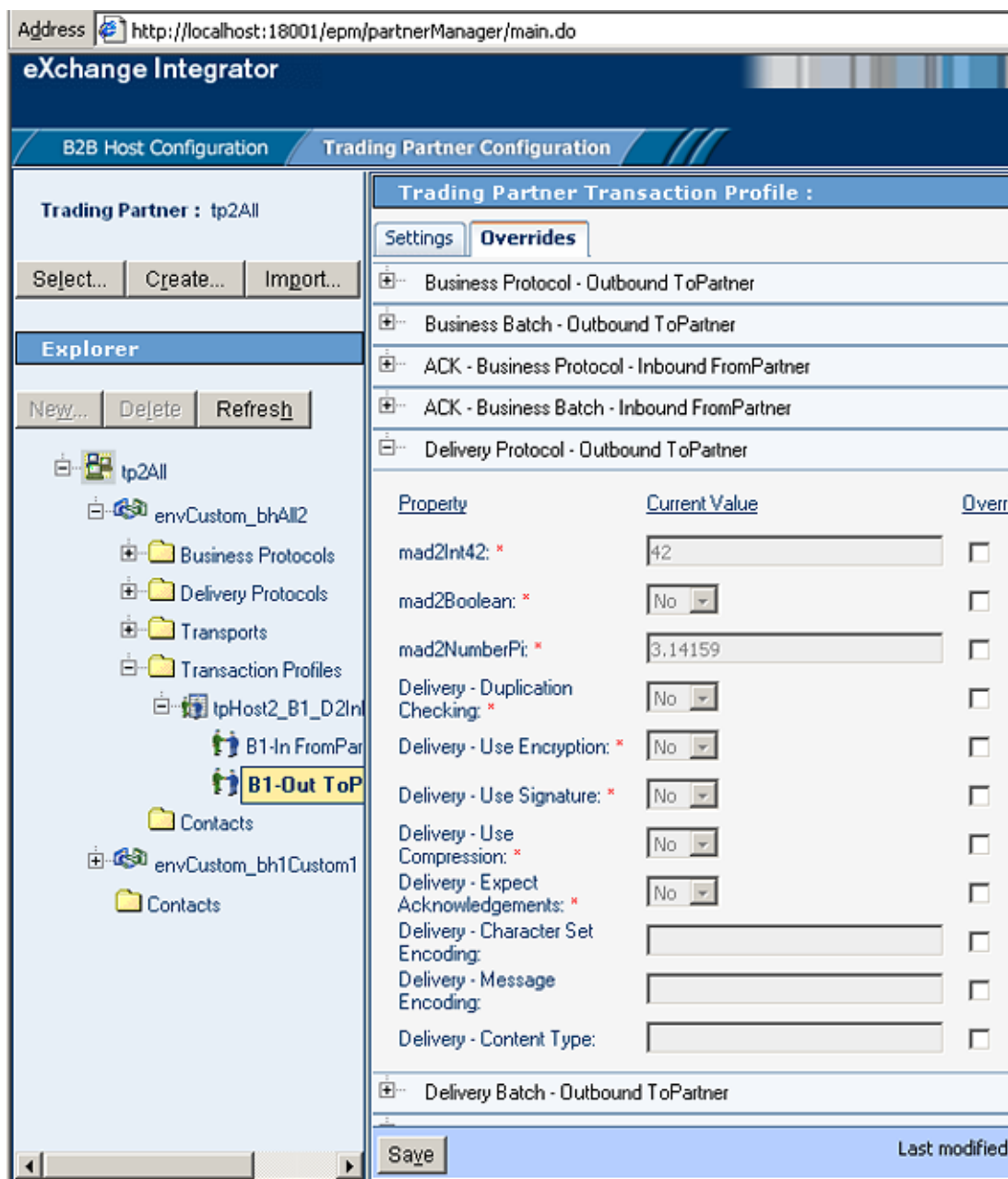


FIGURE 1-5 eXchange Trading Partner Configuration

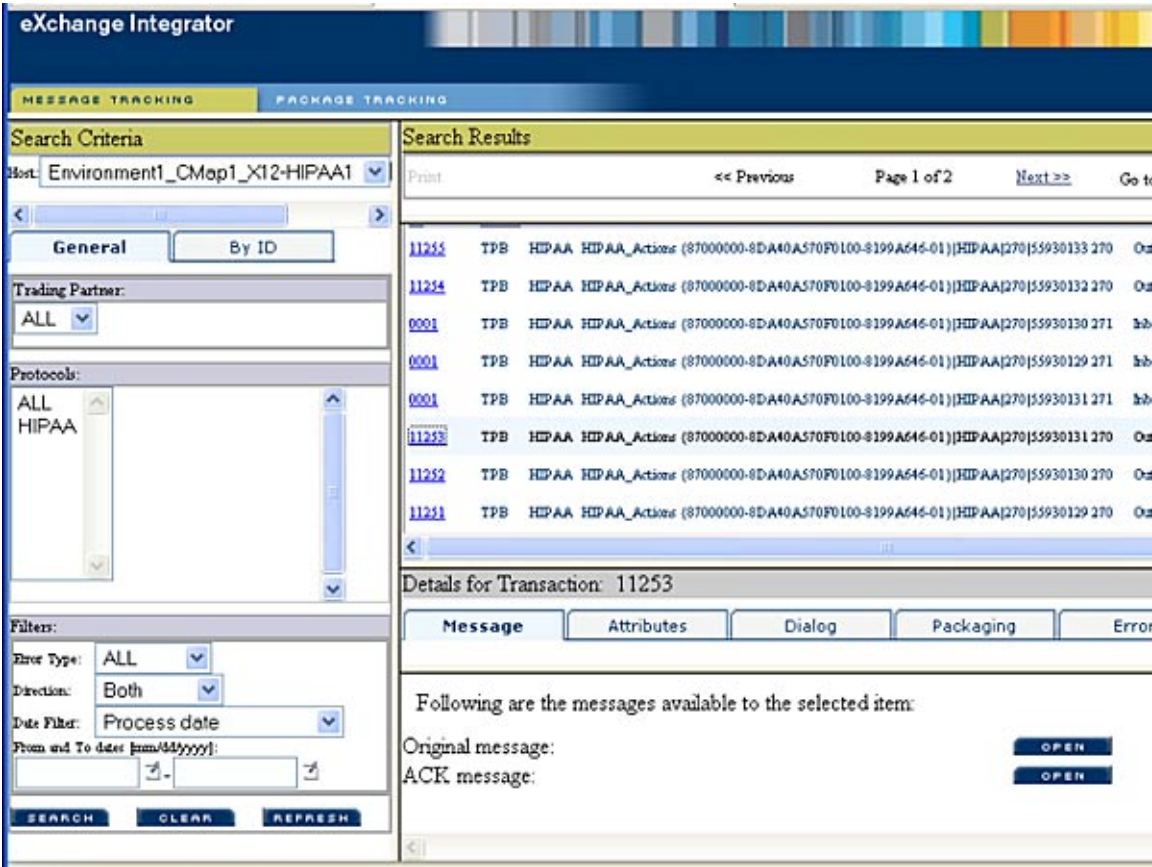


FIGURE 1-6 eXchange Message Tracker

Process Overview

Using eXchange Integrator 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 saved, its transaction profile settings are stored on the LDAP server. See [Figure 1-3](#).

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 explained in the following sections:

- [“Design Phase: Using Enterprise Designer” on page 23](#)
- [“Design Phase: Using ePM” on page 23](#)
- [“Runtime Phase” on page 24](#)

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 eXchange Service object in the same environment that contains the LDAP and Oracle externals and a B2B Configuration object. Entries related to the host attributes are updated in the LDAP database. (In future releases, the eXchange Service corresponding to the B2B Host is configurable with keystores, trust stores, and certificates for authentication and non-repudiation.) 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 eXchange Service is created, it is used in connectivity maps (both user-created and also pre-supplied) 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.

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.

Saving a Trading Partner profile stores all of its transaction profiles' configuration settings in the LDAP database.

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.

Prerequisites Checklist

This section provides a checklist of prerequisite tasks that you must complete prior to runtime deployment.

▼ One-Time Setup Tasks

- 1 Ensure that external systems are installed and available.
- 2 Make sure that you have done the following:
 - a. Installed the product .sar files.
 - b. Created Logical Host domains.
 - c. Downloaded and run the Oracle scripts.

▼ Enterprise Designer Tasks

- 1 Create or import environments, and then configure them.
- 2 Create and build the B2B Host projects.

- 3 Start the Logical Host (if you have not already done so).
- 4 Build and deploy the GUI and Error projects.
- 5 Create a validation Connectivity Map, and then build and deploy the eXchange deployment.
- 6 Customize the sample project components, and then build and deploy the sample projects.

▼ **ePM Tasks**

- 1 Start ePM.
- 2 Import or create Hosts and Trading Partners.
- 3 Configure or customize Hosts and Trading Partners as needed.

▼ **Run-Time Tasks**

- 1 Feed the input data (run the projects).
- 2 Verify the output data.
- 3 Track messages.

Installing eXchange Integrator

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

What's in This Chapter

- “Supported Operating Systems” on page 27
- “Supported External Applications” on page 28
- “Installing the Product Files” on page 28
- “Database Scripts” on page 40

Supported Operating Systems

The *Java Composite Application Platform Suite Installation Guide*, available on the product media and the Enterprise Manager Documentation tab, and the *Sun B2B Suite Release Notes*, available on <http://docs.sun.com>, contain up-to-date operating system requirements for each supported platform.

Sun B2B Suite 5.1.1 is compatible with the following operating system platforms:

- Sun Solaris 8 (on SPARC), 9 (on SPARC), and 10 (on SPARC) with required patches
- HP-UX v11i (11.11) on PA-RISC, and 11i v2.0 (11.23) 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 SP1 and SP2, and Windows Server 2003 SP1
- Red Hat Enterprise Linux AS 4 (on Intel x86)

Supported External Applications

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

Database for Message Tracker

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

- Oracle 9.01
- Oracle 9.2
- Oracle 10.1

Database for Persistence and Monitoring via eInsight Engine

In addition, eXchange Integrator 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 8i (8.1.7), 9i (9.0.1 and 9.2), and 10g

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

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 Composite Application Platform Suite Installation Guide*, which is available on the product media and can also be accessed via the Enterprise Manager Documentation tab.

Uploading B2B Suite Product Files to the Repository

▼ To upload product files to the Repository

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

- You have checked the *Sun B2B Suite Release Notes* document 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.

A Repository server must be running on the machine where you will be uploading the product files and the following .sar files must have already been uploaded to this Repository:

- eGate Enterprise Designer (eGate.sar) Release 5.1.3
- eInsight Business Process Manager (eInsight.sar) Release 5.1.3
- Batch eWay (BatcheWay.sar) Release 5.1.3
- File eWay (FileeWay.sar) Release 5.1.3
- LDAP eWay (LDAPeWay.sar) Release 5.1.3
- HTTPS eWay (HTTPeWay.sar) Release 5.1.3
- Oracle eWay (OracleWay.sar) Release 5.1.3

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

- 1 **On a Windows machine, start a Web browser and point it at the machine and port (usually 12000) 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:12000`
`http://serv1234.company.com:22000`

- 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 5.1.3.

- 3 When the Suite Installer displays the Administration tab, click the link to install additional products.

See Figure 2–1.

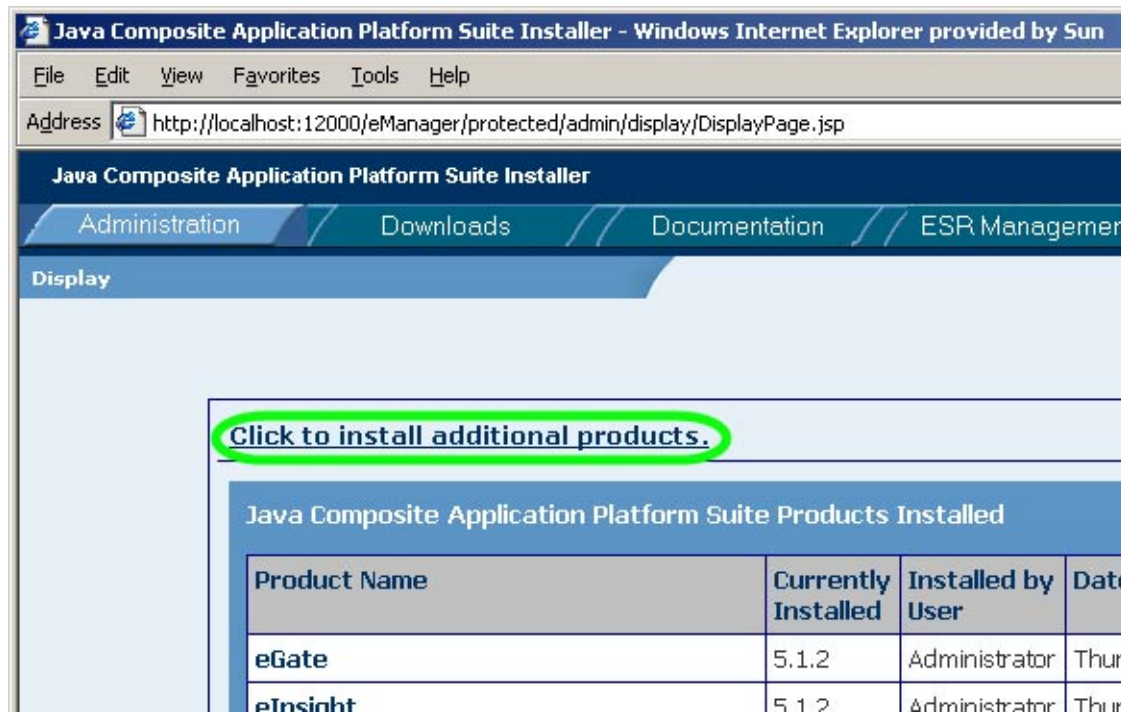


FIGURE 2–1 Suite Installer: Installing Additional Products

- 4 In the Display”>> Select page, open the Core Products category

See Figure 2–2. Depending on the previous installation, eXchange Integrator 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.

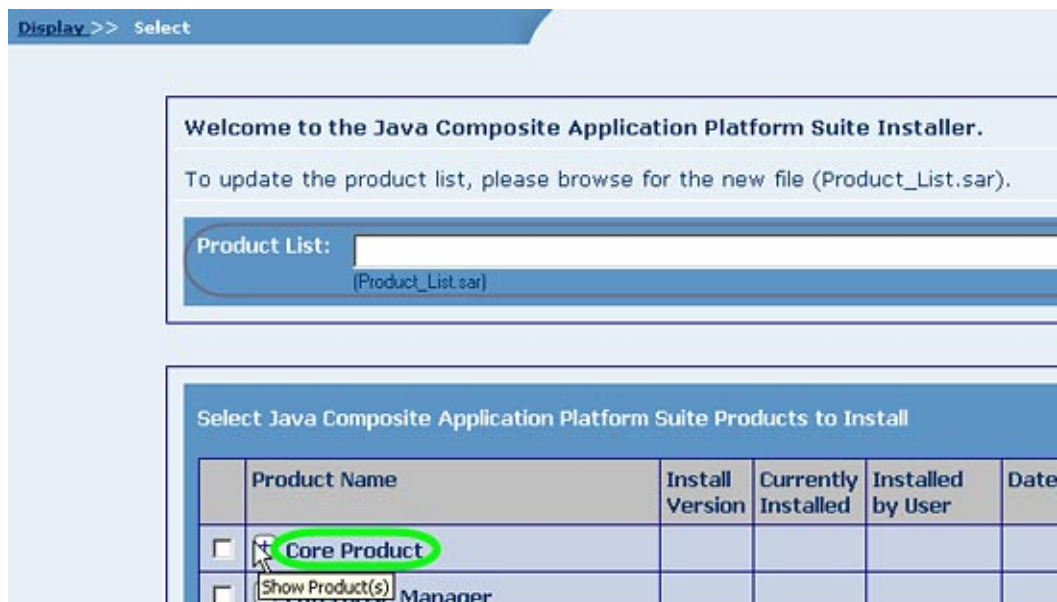
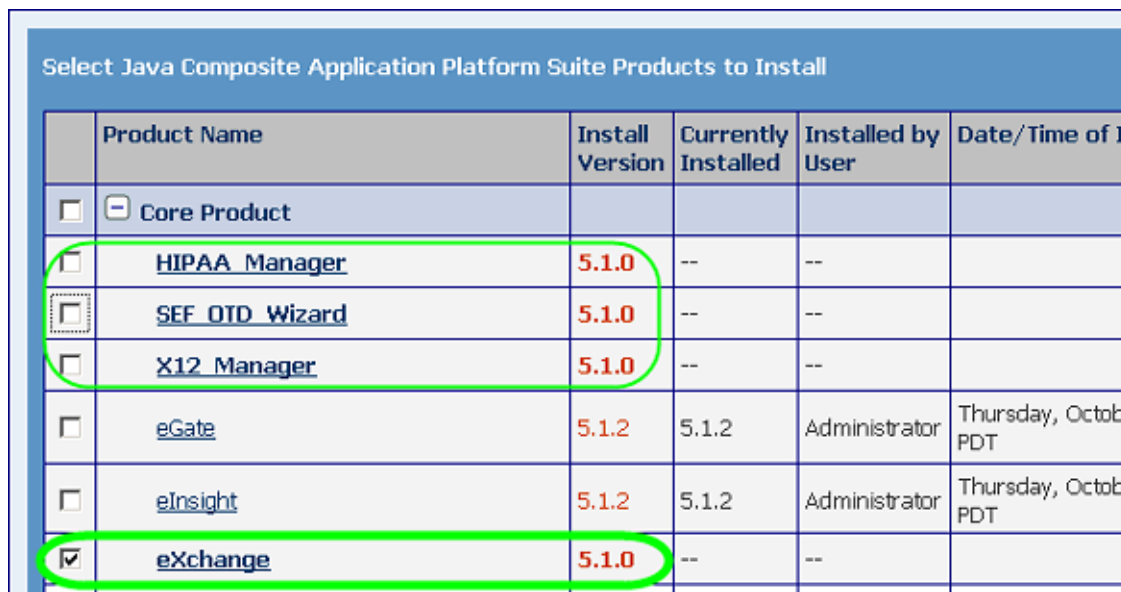


FIGURE 2-2 Suite Installer: Opening the List of Core Products

- 5 In the list of core products, select eXchange.
See Figure 2-3.



	Product Name	Install Version	Currently Installed	Installed by User	Date/Time of Installation
<input type="checkbox"/>	Core Product				
<input type="checkbox"/>	HIPAA Manager	5.1.0	--	--	
<input type="checkbox"/>	SEF OTD Wizard	5.1.0	--	--	
<input type="checkbox"/>	X12 Manager	5.1.0	--	--	
<input type="checkbox"/>	eGate	5.1.2	5.1.2	Administrator	Thursday, October 11, 2007 10:10:10 PDT
<input type="checkbox"/>	eInsight	5.1.2	5.1.2	Administrator	Thursday, October 11, 2007 10:10:10 PDT
<input checked="" type="checkbox"/>	eXchange	5.1.0	--	--	

FIGURE 2-3 Suite Installer: List of Core Products Showing eXchange

- 6 **Select other core B2B products you want to install, and 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 Integrator.
 - If you have not previously installed a required eWay or a Logical Host, you can do so now.
 - To access the documentation and sample Projects for eXchange Integrator and other products, select the appropriate items under the Documentation tab.
- 7 **When you have selected all the products you want to install, click the Next button.**
- 8 **In the Select >> Upload page, browse to and select eXchange.sar. Repeat with each of the other SAR files you want to install.**
See [Figure 2-4](#).
- 9 **When done, click Next.**



FIGURE 2-4 Suite Installer: Selecting Files to Install

Note – eXchange.sar may take some time to install.

The .sar files are uploaded to the Repository. To install additional products, click the `install additional products` link. The `Select >> Upload >> Install` page allows you to select other products. See [Figure 2-5](#).

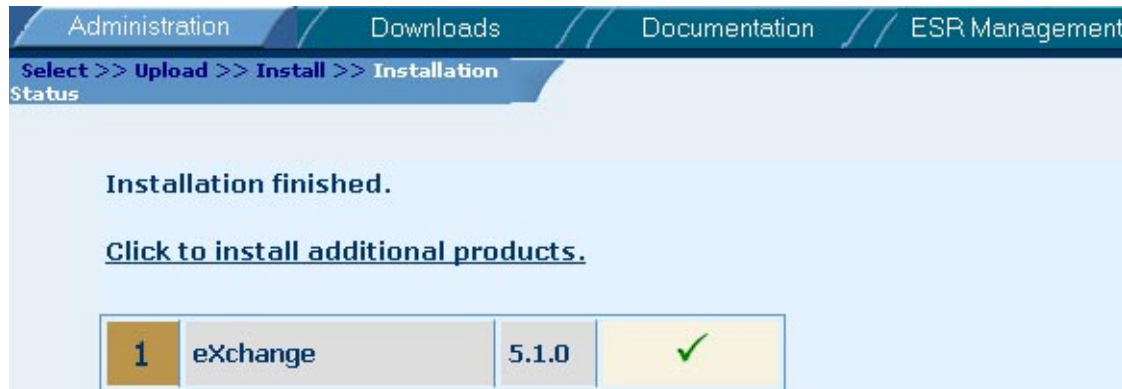


FIGURE 2-5 Suite Installer: Successful Installation of eXchange.sar

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.

Refreshing Enterprise Designer

The following steps are needed only if you have uploaded (or re-uploaded) a SAR file, which requires a refresh to 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 5.1.3” besides “Base ESR”, you need to take the following steps.

▼ To refresh an existing installation of Enterprise Designer

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 Integrator product files.

- 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 2–6.

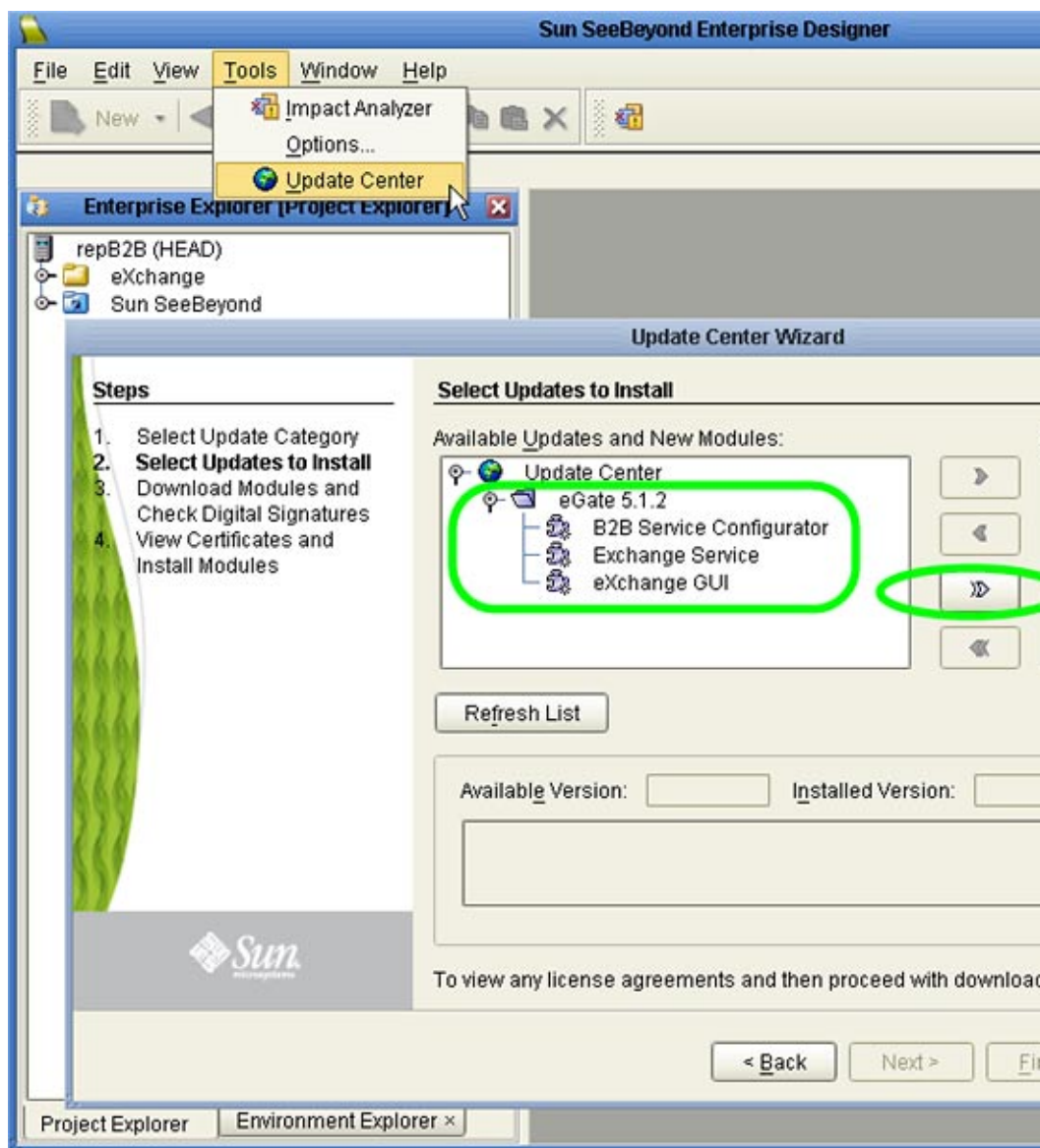


FIGURE 2-6 Update Center Wizard: Select Modules to Install

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

- 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 2–7](#).

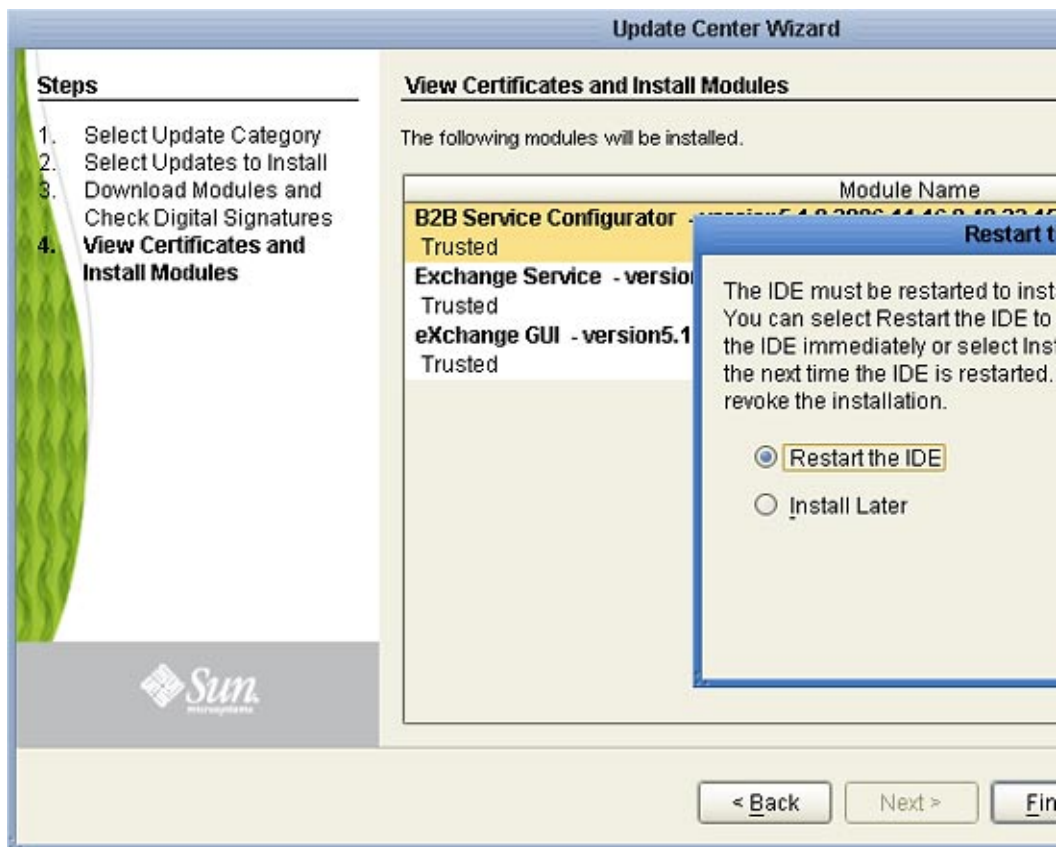


FIGURE 2-7 Update Center Wizard: Restart Enterprise Designer

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

After You Install

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

- First-time installation of eXchange Integrator: You must configure an LDAP-compliant directory server to hold Trading Partner information. Follow the steps described in “LDAP Server” on page 38. For additional information on obtaining, installing, and viewing LDAP Servers, go to http://www.sun.com/software/products/directory_srvr_ee/get1.jsp.
- First-time installation of eXchange Integrator: You must create an eXchange Integrator database schema and configure a database instance. Follow the steps described in “Database Scripts” on page 40.

- 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 the *eInsight Business Process Manager User's Guide*.

LDAP Server

eXchange Integrator 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 version 5.2 or 6 is recommended. After it is installed and set up, follow all steps described in [Table 2-1](#).
- Otherwise: Have your LDAP administrator configure the LDAP server as noted in [Table 2-1](#), and then install the eXchange Integrator schema using the steps in “[Installing the eXchange LDAP Schema](#)” on [page 39](#), or their equivalent on your server.

To configure the Directory Server

Provide values appropriate for your site. (See [Table 2-1](#); or see *eGate Integrator System Administration Guide*. The following are supplied for illustrative purposes:

TABLE 2-1 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	ldap://host.sun.com:1389/ou=B2B1,ou=myName	Change appropriately.
Directory Manager DN	cn=Manager,dc=sun,dc=com	Change common name and domain components appropriately.
Password	password	Change to an appropriate value.
	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.	

Installing the eXchange LDAP Schema

This section describes how to install the eXchange Integrator LDAP schema. For more information, see [Table 2-1](#).

▼ To install the eXchange Integrator 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 [Table 2-1](#) to download and set up a new LDAP server before continuing with the steps below.

- 1 Shut down the LDAP server.**
- 2 If installed from SunONE or JES, change directories to the following location:**
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.**

Database Scripts

The oracle510.zip file contains scripts for creating a database instance that uses the eXchange Integrator database schema. This eXchange Integrator 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 Integrator Database Instance” on page 40](#)
- [“Extracting, Customizing, and Running Database Setup Scripts” on page 43](#)
- [“Running Database Scripts to Set Up the eXchange IntegratorDatabase” on page 46](#)

Do not confuse the eXchange Integrator 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 the *eInsight Business Process Manager User's Guide*.

Creating and Configuring the eXchange Integrator 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 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 9i wizard.

▼ To create a new database instance for eXchange Integrator

- 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 Integrator Database

If you create a new database, you must increase the `open_cursors` parameter for the eXchange Integrator 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.

▼ To edit the value of `open_cursors` in the `init.ora` file for the eXchange Integrator 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 43](#); at this release, you must extract and run database scripts whether you are installing from scratch or upgrading a previous release of eXchange Integrator.

▼ To configure the value of `open_cursors` in the eXchange Integrator 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.

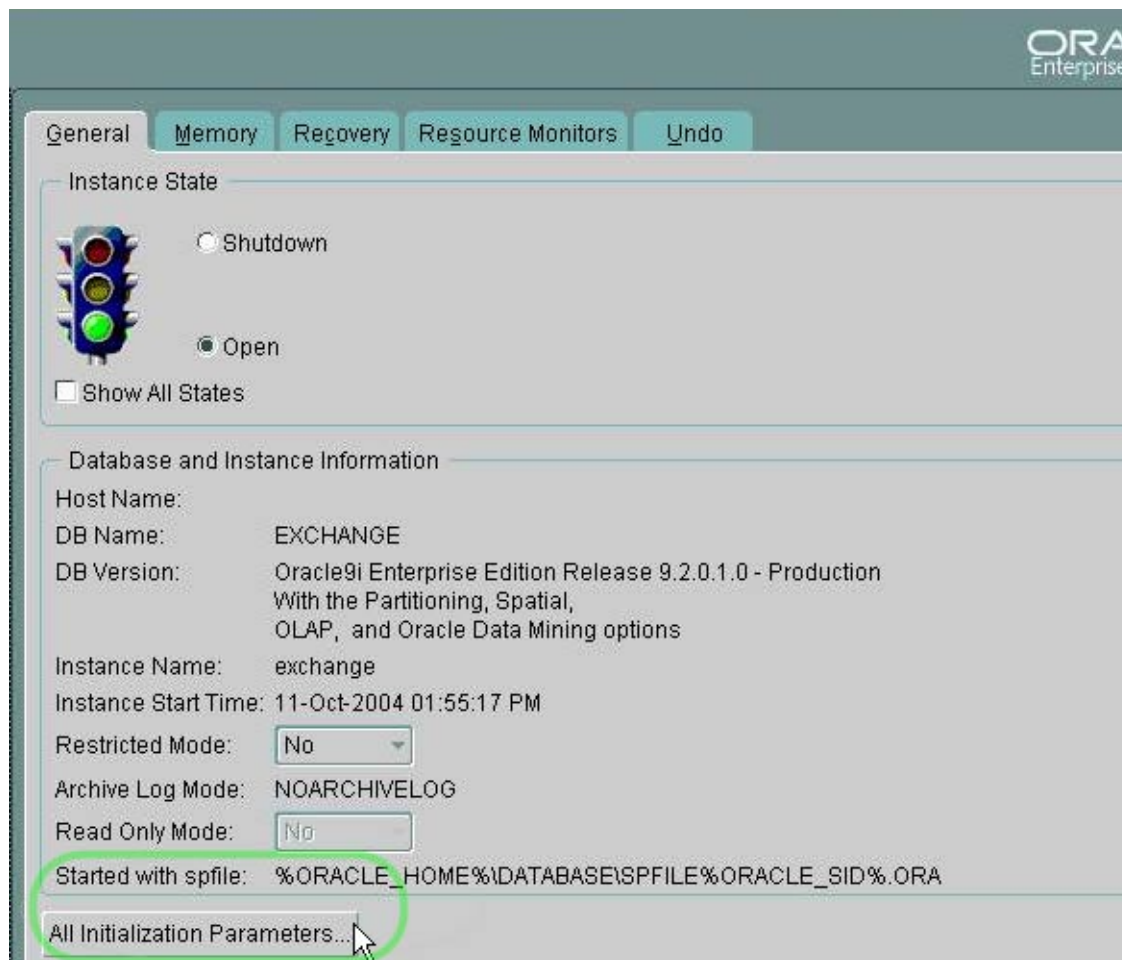
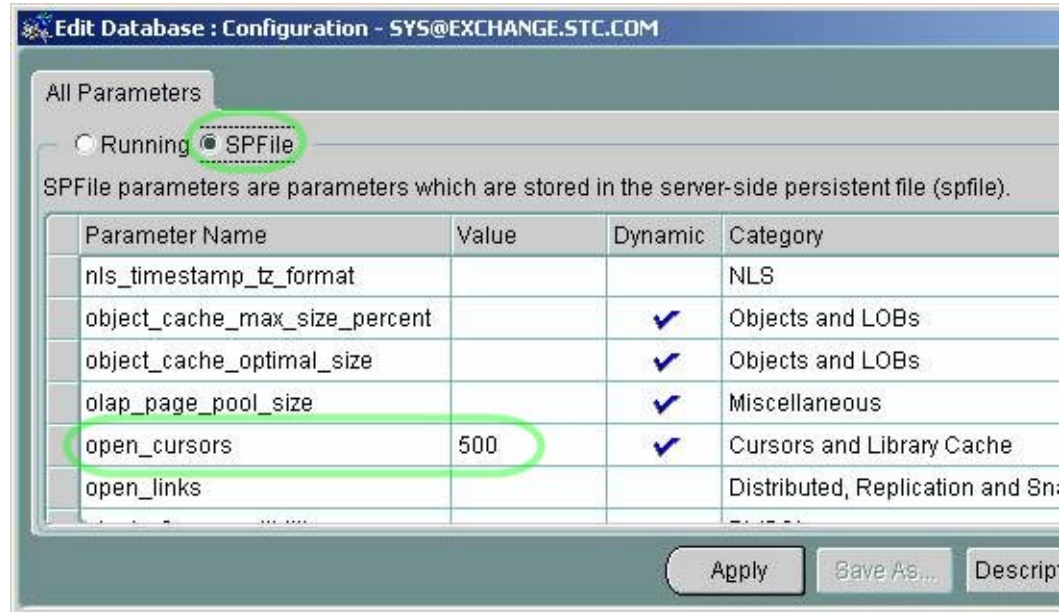


FIGURE 2-8 Oracle Configuration of Initialization Parameters: General Tab

- 4 In the “Edit Database: Configuration [...]” dialog box, click the SPFile option button, scroll to the open_cursors parameter, and change its value to 500.



- Click the Apply button. In response to the prompt ("Would you like to apply this change to the current database?"), click Yes.

Extracting, Customizing, and Running Database Setup Scripts

Note – Do not skip this section. To use eeXchange Integrator, 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 Integrator 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 Integrator schema on the database by doing one of the following:

- Edit setenv.cmd so it reflects your system environment (see [Table 2-2](#)), and then run the two other command scripts (see [“Running Database Scripts to Set Up the eXchange IntegratorDatabase” on page 46](#));— or —
- Run the SQL scripts directly, supplying system information at run time; see [“Running Database Scripts to Set Up the eXchange IntegratorDatabase” on page 46](#).

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 2-2](#).

TABLE 2-2 Typical, Default, or Presupplied Values for eXchange Integrator Database Setup

Item	Typical or presupplied value
UserID/password combination for Oracle system	sys/manager or system/manager
Name of eXchange database instance (see “Creating and Configuring the eXchange Integrator Database Instance” on page 40)	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
- createtablespace.cmd
- createtablespace.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 than eXchange, then open the `createtablespace.sql` file and make the appropriate change or changes in the first line.

Note – The database user who runs the SQL scripts must have permission to create tables.

Running Database Scripts to Set Up the eXchange IntegratorDatabase

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

- Edit `setenv.cmd` so it reflects your system environment (see [Table 2–2](#)), and then run the two other command scripts (see [“Running Database Scripts to Set Up the eXchange IntegratorDatabase” on page 46](#));— or—
- Run the SQL scripts directly, supplying system information at run time; see [“Running Database Scripts to Set Up the eXchange IntegratorDatabase” on page 46](#).

▼ 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 `createtablespace.sql` if necessary.

- 2 Enter the following command:

```
createtablespace
```

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 [“Running Database Scripts to Set Up the eXchange IntegratorDatabase” on page 46](#) as needed to create other user/password entries for eXchange Integrator users.

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 Integrator 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.**

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

- 2 **Enter the following SQL*Plus command:**

```
path\sqlplus system/SYSTEMPWD@TNSNAME@createtablespace.sql
```

where:

SYSTEMPWD is the password for the system login ID

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

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

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

When this finishes, you have created new tablespaces.

- 3 **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 Integrator.

Here is an example of a valid command:

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

- 4 **In response to the system prompt for value #1, enter the username. For example: ex_admin**
- 5 **In response to the system prompt for value #2, enter the password. For example: ex_admin**
- 6 **Repeat steps “Running Database Scripts to Set Up the eXchange IntegratorDatabase” on page 46 and “Running Database Scripts to Set Up the eXchange IntegratorDatabase” on page 46 as needed to create user/password entries for eXchange Integrator users.**

- 7 After you run the createtablespace 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 Integrator administrator username and password are both *ex_admin*.

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 Integrator database. End users can create Oracle OTDs based on this database, and can use it for message tracking and other eXchange Integrator functions.

▼ To reinitialize the database

- 1 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 43](#)).

- 2 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 Integrator administrator username and password are both assumed to be *ex_admin*.

eXchange Integrator Features

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

What's in This Chapter

- “Project Tree Organization” on page 49
- “Transport Attribute Definitions” on page 50
- “Channel Manager” on page 58
- “Message Tracker” on page 64
- “B2B Protocols for X12, HIPAA, EDIFACT, AS2, and ebXML” on page 65
- “Handling Errors” on page 65

Project Tree Organization

Initial installation of eXchange Integrator 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 ⇒ eXchange — contains all of the other components used by eXchange Integrator, 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.

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 contains presupplied transport attribute definitions (TADs) and corresponding OTDs. See [Figure 3–1](#). For more information, see “[Creating and Configuring TADs](#)” on page 77.

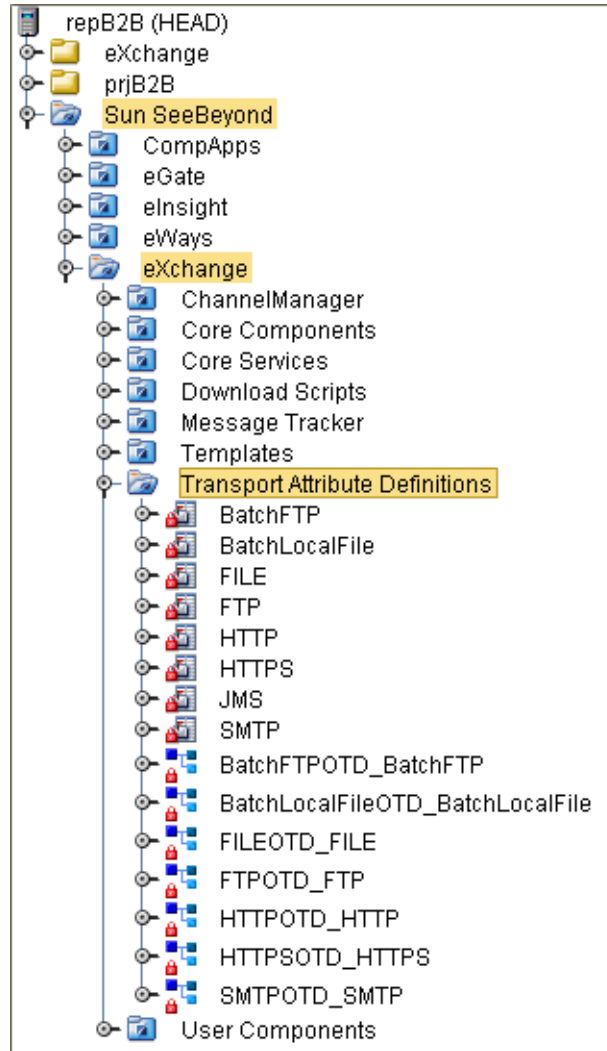


FIGURE 3-1 Sun SeeBeyond ⇒ eXchange⇒Transport Attribute Definitions Folder

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 3-1](#).

TABLE 3-1 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)

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 3-2](#) lists the attributes of the BatchFTP TAD.

TABLE 3-2 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		

TABLE 3-2 Attributes for the BatchFTP Transport Attributes Definition (Continued)

Name	Display	Type	Req?	Direction	Default	List
DirectoryListingStyle	[...]	String	No	Both		
ClientClassName		String	No	Both		
ProviderClassName		String	No	Both		
Mode		String	No	Both		
UsePASV		Boolean	No	Both		
UserName		String	No	Both		
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		

TABLE 3-2 Attributes for the BatchFTP Transport Attributes Definition *(Continued)*

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

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 3-3](#) lists the attributes of the BatchLocalFile TAD.

TABLE 3-3 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		

TABLE 3-3 Attributes for the BatchLocalFile Transport Attributes Definition *(Continued)*

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

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 3-4 lists the attributes of the FILE TAD.

TABLE 3-4 Attributes for the FILE Transport Attributes Definition

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

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 3-5 lists the attributes of the FTP TAD.

TABLE 3-5 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		
SocksUserName	SocksUserName	String	No	Both		
SocksPassword	SocksPassword	Password	No	Both		
SocksServerPort	SocksServerPort	String	No	Both		

HTTP

For Hypertext Transfer Protocol, the HTTPS eWay can use the URL attribute in the HTTP TAD to access Web pages. The HTTP TAD has the corresponding OTD: HTTPOTD_HTTP.

HTTPS

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

Table 3–6 lists the attributes of the HTTPS TAD.

TABLE 3–6 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 non-default settings: in the Integration Server, the Web server configuration must have its “Enable SSL” parameter set to “True”, and in the HTTPS external, the SSL configuration must supply a value for its “TrustStore” parameter.

JMS

The JMS transport attributes definition is used to transport data into and out of JMS topics and queues.

SMTP

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

Table 3–7 lists the attributes of the SMTP TAD.

TABLE 3–7 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		

TABLE 3-7 Attributes for the SMTP Transport Attributes Definition (Continued)

Name	Display	Type	Req?	Direction	Default	List
UserName	UserName	String	No	Both		
Password	Password	Password	No	Both		

Channel Manager

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

The Sun SeeBeyond⇒eXchange⇒ChannelManager folder contains the ChannelManagerClient OTD. See [Figure 3-2](#).

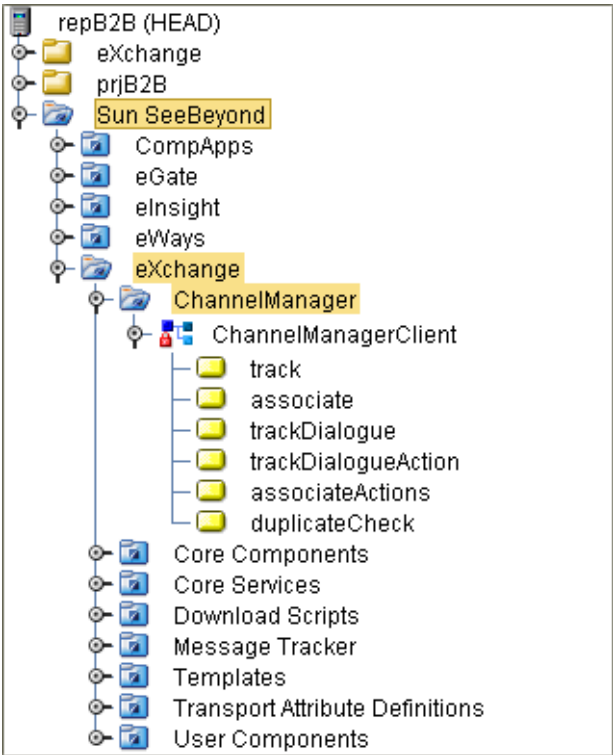


FIGURE 3-2 Sun SeeBeyond⇒eXchange⇒ChannelManager Folder

In this section

- “associate” on page 59
- “associateActions” on page 60
- “duplicatecheck” on page 60
- “track” on page 60
- “trackDialogue” on page 61
- “trackDialogueAction” on page 62

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 3–8 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.
TPIId	The database’s unique ID for the trading partner; in other words, the foreign key to <code>ex_trading_partner</code> .
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 3–9 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 3-10 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 3-11 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 Integrator database

TABLE 3-12 Input Containers for ChannelManagerClient.track

Name	Description
Protocol	Name of the protocol being used to handle the message.

TABLE 3-12 Input Containers for ChannelManagerClient.track (Continued)

Name	Description
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)
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 3-13 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 3-14 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.
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 3-15 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 messages in a business conversation (after the initial message was written by trackDialogue operation).

TABLE 3-16 Input Containers for ChannelManagerClient.trackDialogueAction

Name	Description
messageId	(deprecated) Duplicate of actionMessageId
actionName	Name of the messaging action that is processing the message.

TABLE 3-16 Input Containers for ChannelManagerClient.trackDialogueAction (Continued)

Name	Description
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.
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 3-17 Output Containers for ChannelManagerClient.trackDialogueAction

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

Message Tracker

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

The Sun SeeBeyond⇒eXchange⇒Message Tracker folder contains only one item: the tracker application itself. See [Figure 3-3](#).

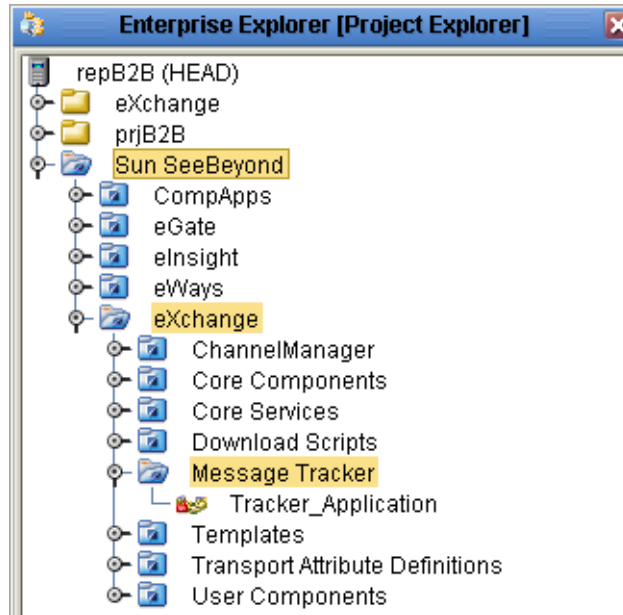


FIGURE 3-3 eXchange⇒Message Tracker Folder

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 94](#). For information on using the Message Tracker web client to view tracking information that has been written to the database, see [“Message Tracker” on page 163](#).

B2B Protocols for X12, HIPAA, EDIFACT, AS2, and ebXML

If you have licensed and installed any of the protocol manager composite applications, your Sun SeeBeyond⇒eXchange⇒ Protocol Managers folder contains 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.

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*.

Message Tracker

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. For information on linking the B2B host to a tracker application, see [“Connecting the B2B Host to Oracle and LDAP Externals”](#) on [page 94](#).

eXchange Integrator Standard Error-Handling Topics

System-level errors, and business errors that cannot be handled by message tracking, are considered faults; eXchange Integrator catches such faults and publishes the ExStdEvent — with the error information now contained inside the ExStdEvent — to a standard eXchange Integrator 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.

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*.

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 3–4](#).

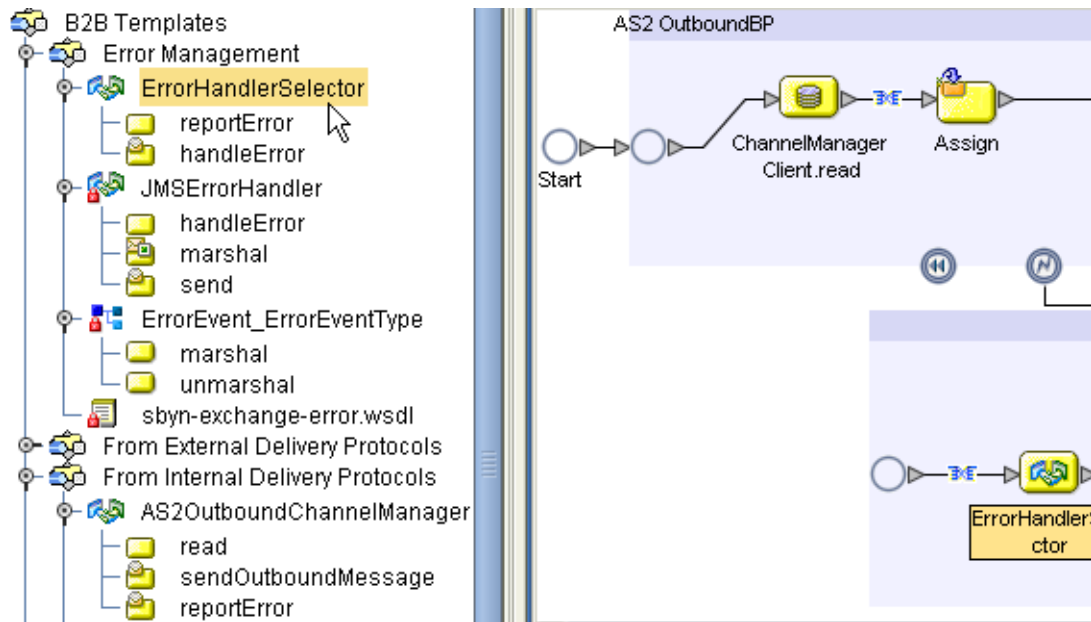


FIGURE 3-4 Template B2B Protocol Process “ErrorHandlerSelector”

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 3-5](#).

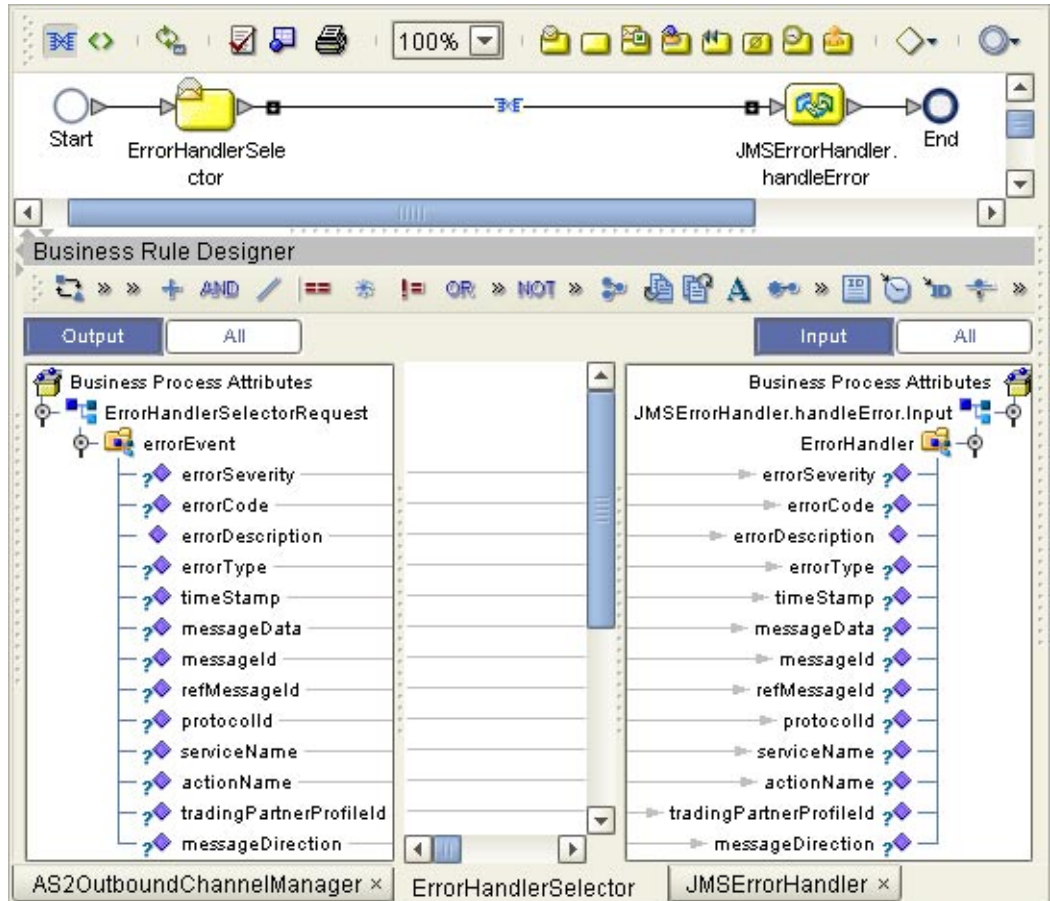


FIGURE 3-5 Mapping from ErrorHandlerSelector to ErrorHandler

The purpose of the JMSEErrorHandler 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:

▼ To create an SMTP-based error handler

- 1 Export the sbyn-exchange-error.wsdl file.
- 2 In the Error Management folder, create a new B2B protocol process and name it; for example, SMTPErrorHandler.

- 3 **Open the properties of SMTPErrorHandler and do the following:**
 - a. Use the WSDL tab to load the .wsdl file you exported.
 - b. Use the Partners tab to add a new partner named ErrorSelector.
 - c. Use 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 and connect them.**
- 5 **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 eeXchange Integrator projects in this Repository.

Using eXchange Integrator 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 Integrator.

What's in This Chapter

- “Overview” on page 71
- “Setting Up a B2B Host and Its Components” on page 74
- “Setting Up Attribute Definitions” on page 77
- “Building a B2B Host” on page 88
- “Building and Deploying the GUI Projects” on page 99
- “Building and Deploying the Error-Handling Projects” on page 103

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:
 - Populates the LDAP database with metadata for the B2B Host.
 - 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.

Transaction Profiles

eXchange Integrator 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.

Using Enterprise Designer

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 Integrator, such as X12 or HIPAA, come equipped with prebuilt BPADs. You can also create your own BPADs for custom business protocols. Similarly for transport: You can use either the standard TADs supplied with eXchange Integrator (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 is also 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 eXchange Integrator 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.

B2B Host Procedures

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.

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.

- 2 Point to New and click B2B Host.

See [Figure 4–1](#).

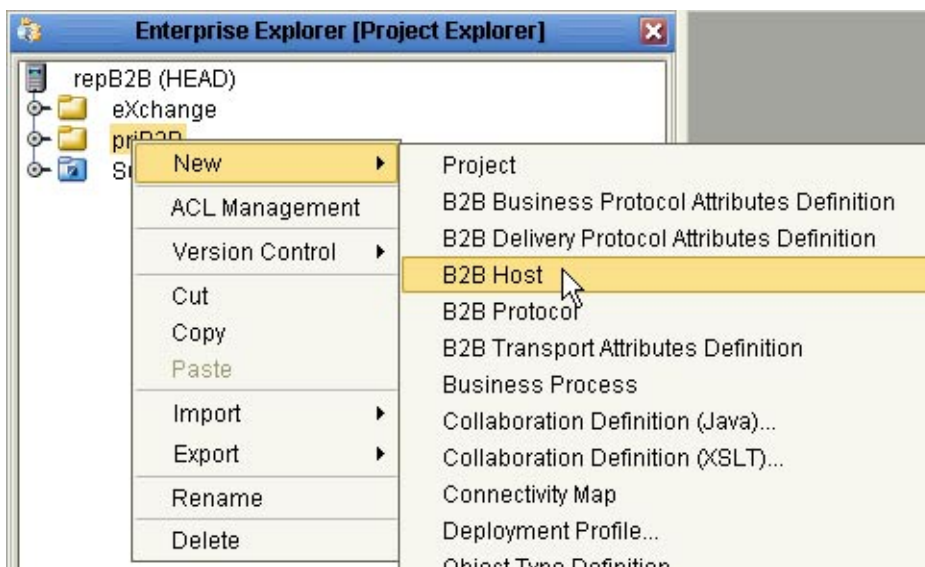


FIGURE 4-1 Creating a New B2B Host

- 3 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.

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 4–2](#).

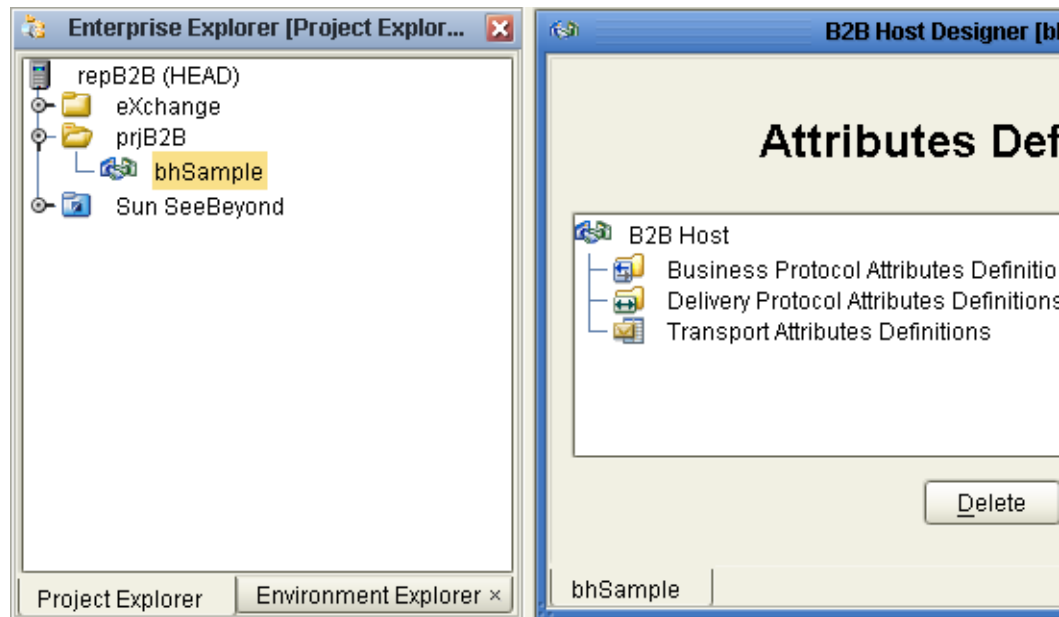


FIGURE 4-2 B2B Host Designer

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 do the following as appropriate:
 - For X12: Open the Protocol Managers⇒X12 Manager⇒OTDs folder and drag the X12 BPAD into the B2B Host Designer window.
 - For HIPAA: Open the Protocol Managers⇒HIPAA Manager⇒OTDs folder and drag the HIPAA BPAD into the B2B Host Designer window.
 - For standard transports: Open the Transport Attribute Definitions folder and drag one or more TADs into the B2B Host Designer window.

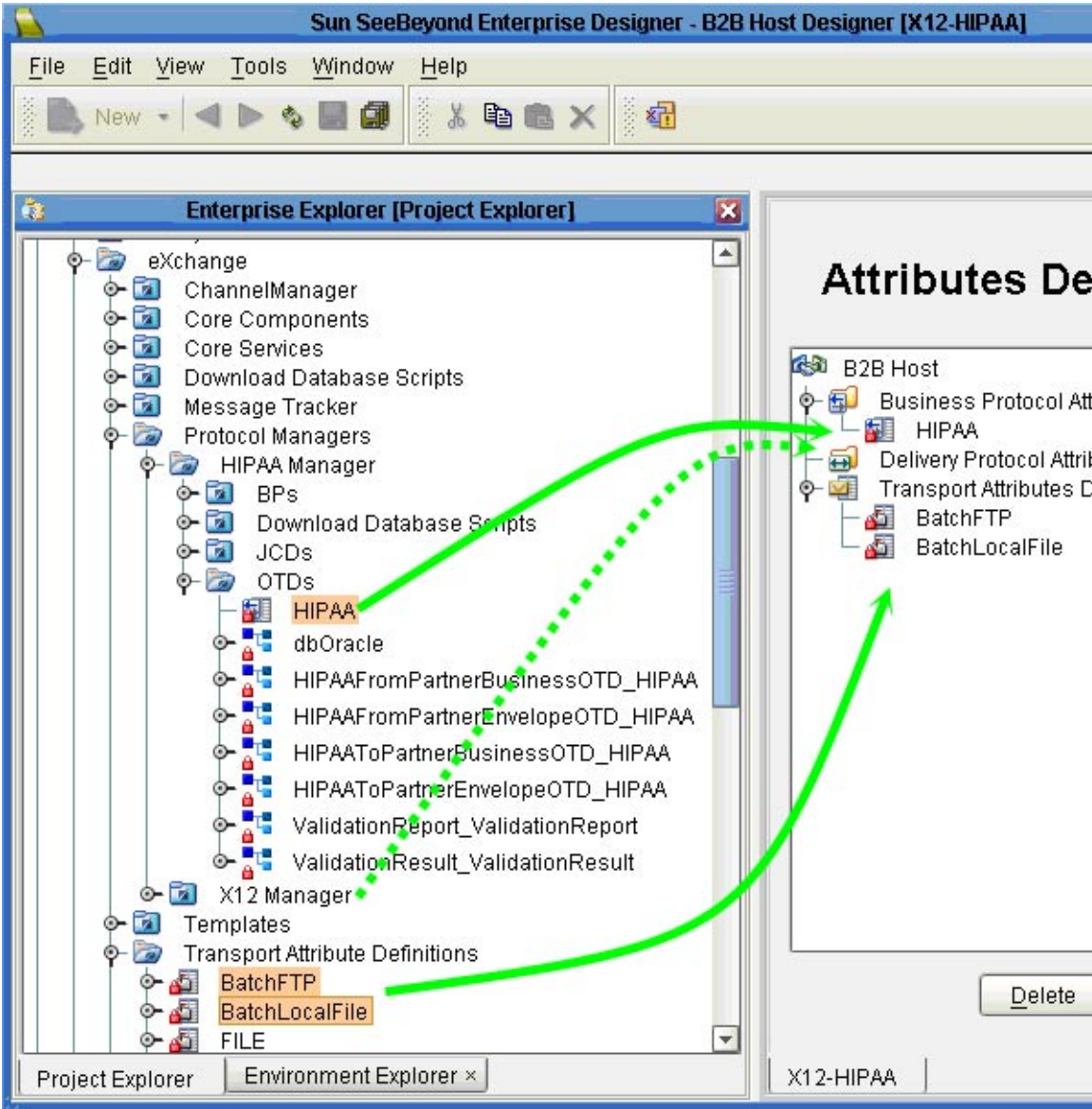


FIGURE 4-3 Business Protocols Window Showing Pre-supplied BPADs and TADs

Figure 4-3 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.

Setting Up Attribute Definitions

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

- “Creating and Configuring TADs” on page 77
- “Creating and Configuring DPADs” on page 81
- “Creating and Configuring BPADs” on page 83

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 Integrator, these parameters are specified by the standard transport attributions definition for FTP. eXchange Integrator 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

- 1 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.
- 2 On the context menu, point to New, and click B2B Transport Attributes Definition.

The project tree displays the new component, and a properties dialog allows you to edit the attribute definitions. See [Figure 4–4](#). 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.

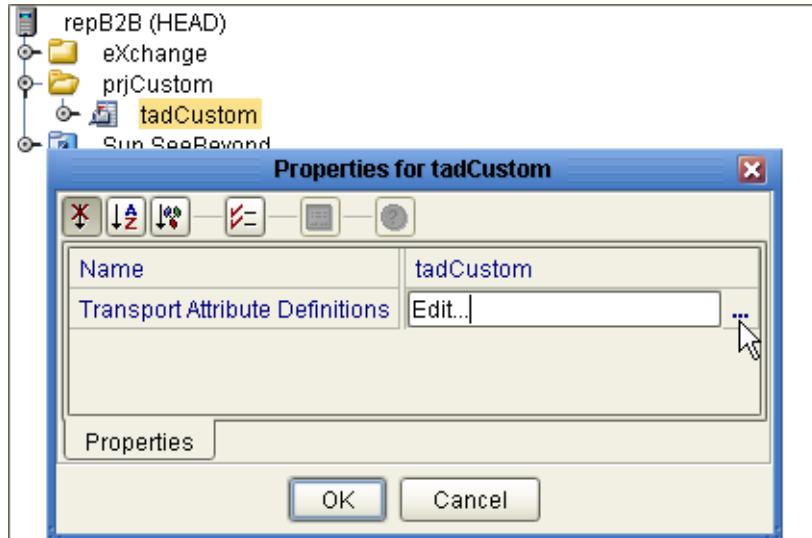


FIGURE 4-4 Custom Transport Attributes Definition

This new component is useful only by virtue of its configuration — you need to add and define attributes that govern the nodes in the OTD that are 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

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

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

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

The Attributes dialog appears. [Figure 4-5](#) shows attributes for a sample bidirectional transport attributes definition that is a modification of the basic File TAD.

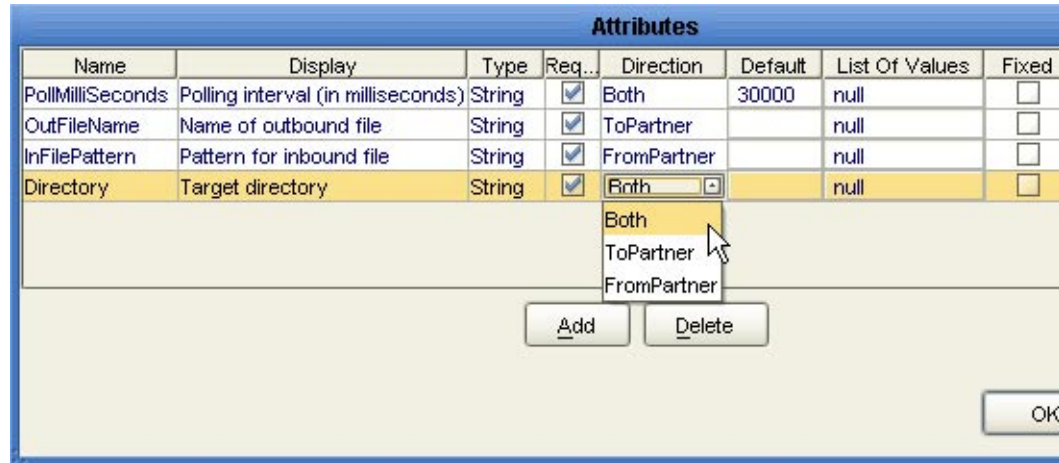


FIGURE 4-5 Custom Attribute Definitions

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:
 - *String* (the default) allows the ePM user to enter any character data
 - *Password* 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
 - *DateTime* prompts the ePM user to supply a date and/or time value, based on the formatting you provide
- 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 to appear in ePM before the user enters data or makes a selection. This is the value that is 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”, double-click the ellipsis [...] button to the far right and use the List of Values dialog box to add entries to the drop-down list that is seen by the end user:
 - Add appends a new item to the end of the list
 - Edit allows you to modify the currently selected item
 - Move Up and Move Down move it higher or lower in the list. See [Figure 4–6](#).

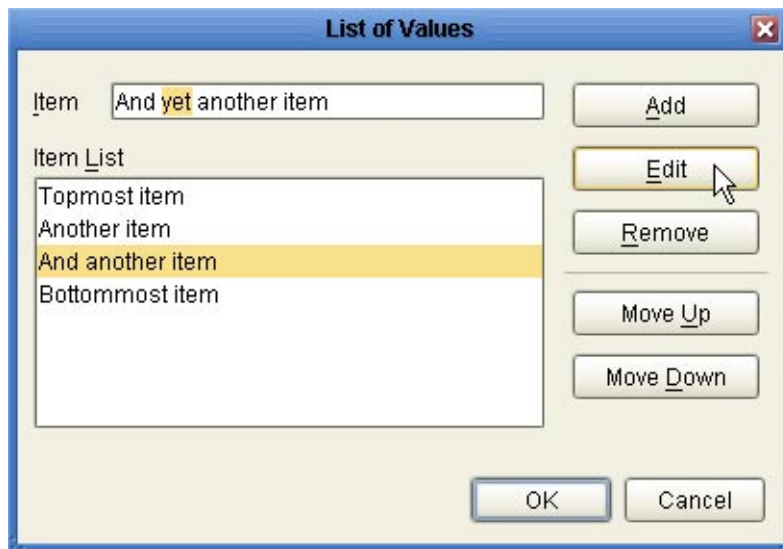


FIGURE 4–6 Attributes Definition - List of Values

- 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 *eWay Batch 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.

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

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 4–7](#) provides an example of the MAD for AS2 version 1.1.

Attributes				
Name	Display	Type	Req...	Direction
AS2_FROM	AS2_FROM	String	<input checked="" type="checkbox"/>	Both
AS2_TO	AS2_TO	String	<input checked="" type="checkbox"/>	Both
AS2_HOST	AS2_HOST	String	<input checked="" type="checkbox"/>	ToPartner
AS2_VERSION	AS2_VERSION	String	<input checked="" type="checkbox"/>	ToPartner
HTTP_FROM	AS2_HTTP_FROM	String	<input checked="" type="checkbox"/>	ToPartner
SIGNATURE_REQ	AS2_SIGNATURE_REQ	Boolean	<input checked="" type="checkbox"/>	ToPartner
ENCRYPT_REQ	AS2_ENCRYPT_REQ	Boolean	<input checked="" type="checkbox"/>	ToPartner
MDN_REQ	AS2_MDN_REQ	Boolean	<input checked="" type="checkbox"/>	ToPartner
MDN_SIGNATURE_RE	AS2_MDN_SIGNATURE_REQ	Boolean	<input checked="" type="checkbox"/>	ToPartner
MDN_RESP_TYPE	AS2_MDN_RESP_TYPE	List of Val	<input checked="" type="checkbox"/>	ToPartner
MDN_DELIVERY_URL	AS2_MDN_DELIVERY_URL	String	<input type="checkbox"/>	ToPartner
AS2_SUBJECT	AS2_SUBJECT	String	<input checked="" type="checkbox"/>	ToPartner
COMPRESSED	AS2_COMPRESSED	Boolean	<input checked="" type="checkbox"/>	ToPartner
COMPRESSED_BEFO	AS2_COMPRESSED_BEFORE_SIGNED	Boolean	<input checked="" type="checkbox"/>	ToPartner
PAYLOAD_TYPE	AS2_PAYLOAD_TYPE	String	<input checked="" type="checkbox"/>	ToPartner
MESSAGE_FORMAT	AS2_MESSAGE_FORMAT	List of Val	<input checked="" type="checkbox"/>	ToPartner
ENCODING	AS2_ENCODING	List of Val	<input checked="" type="checkbox"/>	ToPartner
REPORTING_UA	AS2_REPORTING_UA	String	<input checked="" type="checkbox"/>	FromPart...
POSITIVE_MDN_DISP	AS2_POSITIVE_MDN_DISPOSITION_MESSAGE	String	<input type="checkbox"/>	FromPart...

FIGURE 4–7 Default Messaging Attributes Definition (MAD) for AS2 v1.1

▼ 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 to New, and 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 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, and then click Properties⇒Edit.

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:
 - *String* (the default) allows the ePM user to enter any character data
 - *Password* 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
 - *DateTime* prompts the ePM user to supply a date and/or time value, based on the formatting you provide

- 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 *eWay Batch Adapter User's Guide*. The chapter on understanding OTDs has a section on using special characters.
 - 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 to appear in ePM before the user enters data or makes a selection. This is the value that is 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”, double-click the ellipsis [...] button to the far right and use the List of Values dialog box to add entries to the drop-down list that is seen by the end user:
 - a new item to the end of the list
 - Edit modifies the currently selected item
 - Up and Down move it higher or lower in the list
- 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.

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 4–8](#) 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 are displayed in ePM.

General Attributes						
Name	Display	Type	Req...	Direction	Default	List of Values
ENABLE	Enable grouping	Boolean	<input checked="" type="checkbox"/>	ToPartner	false	false, true
S006_0040	Sender Identifier	String	<input type="checkbox"/>	Both		
S006_0007	Sender identification code Qualifier	String	<input type="checkbox"/>	Both		
S007_0044	Recipient identifier	String	<input type="checkbox"/>	Both		
S007_0007	Recipient identification code qualifier	String	<input type="checkbox"/>	Both		
D0058	Application password	Password	<input type="checkbox"/>	ToPartner		

FIGURE 4-8 Default Business Attribute Definitions (BADs) for X12 and EDIFACT

Figure 4-9 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 are displayed in ePM.

Attributes						
Name	Display	Type	Req...	Direction	Default	
UNA	UNA characters	String	<input type="checkbox"/>	Both		
VERSION	Syntax version	List of Values	<input checked="" type="checkbox"/>	ToPartner	4	4, 3
S001_0001	Syntax identifier	List of Values	<input checked="" type="checkbox"/>	ToPartner	UNOA	UNO
S002_0004	Sender identifier	String	<input checked="" type="checkbox"/>	Both		
S002_0007	Sender identification code Qualifier	String	<input type="checkbox"/>	Both		
S002_0008	Sender internal identifier	String	<input type="checkbox"/>	Both		
S002_0042	Sender internal sub-identifier (v4 only)	String	<input type="checkbox"/>	Both		
S003_0010	Recipient identifier	String	<input checked="" type="checkbox"/>	Both		
S003_0007	Recipient identification code qualifier	String	<input type="checkbox"/>	Both		
S003_0014	Recipient internal identifier	String	<input type="checkbox"/>	Both		
S003_0046	Recipient internal sub-identifier (v4 only)	String	<input type="checkbox"/>	Both		
S005_0022	Recipient password	Password	<input type="checkbox"/>	ToPartner		
S005_0025	Recipient password qualifier	List of Values	<input type="checkbox"/>	ToPartner		AA,
D0026	Application reference	String	<input type="checkbox"/>	ToPartner		
D0029	Processing priority	List of Values	<input type="checkbox"/>	ToPartner		A
D0031	Acknowledge request	List of Values	<input type="checkbox"/>	Both		1, 2
D0032	Agreement identifier	String	<input type="checkbox"/>	ToPartner		
D0035	Test indicator	List of Values	<input type="checkbox"/>	ToPartner		1, 2
FORCE	Overriding syntax	List of Values	<input checked="" type="checkbox"/>	FromPartner	none	non
VLEVEL	Validation strictness	List of Values	<input checked="" type="checkbox"/>	FromPartner	Strict	Stri

FIGURE 4-9 Default Enveloping Attribute Definitions (EADs) for X12 and EDIFACT

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 context menu, point to New, and 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 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 to display the Envelope Attribute Definitions dialog box.

See [Figure 4–10](#).



FIGURE 4–10 Accessing a BPAD’s Envelope Attribute Definitions (EAD)

You use this dialog box to create and set the envelope attribute definitions (EAD) of the current BPAD.

- 3 Click the Add button as many times as needed (a four-row EAD is illustrated in [Figure 4–11](#)) 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:
 - *String* (the default) allows the ePM user to enter any character data
 - *Password* 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
 - *DateTime* prompts the ePM user to supply a date and/or time value, based on the formatting you provide
- 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 to appear in ePM before the user enters data or makes a selection. This is the value that is 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”, double-click the ellipsis [...] button to the far right and use the List of Values dialog box to add entries to the drop-down list that is seen by the end user:
 - Add appends a new item to the end of the list
 - Edit modifies the currently selected item
 - 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 *eWay Batch Adapter User’s Guide*. The chapter on understanding OTDs has a section on using special characters.

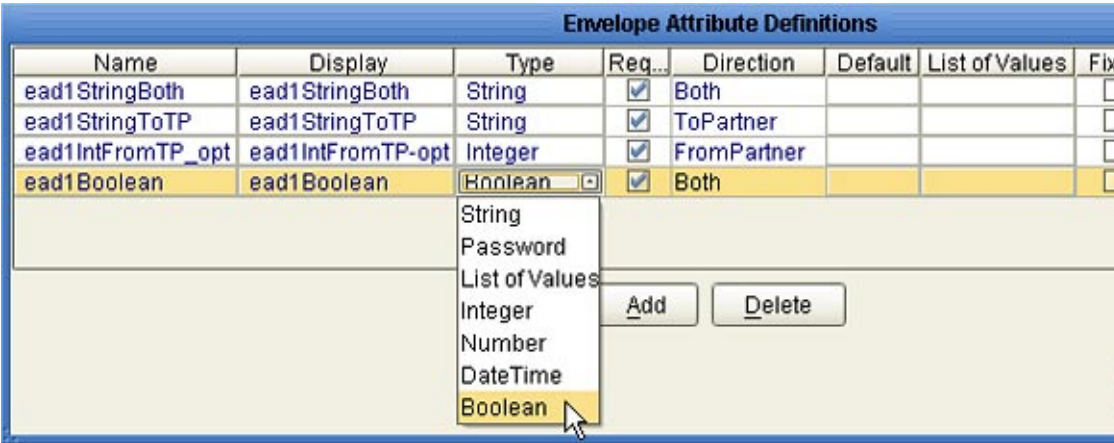


FIGURE 4-11 Custom Envelope Attribute Definitions (EAD)

- 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 causes the BPAD parameters (that is, the ones you defined for the EAD and the BAD) to appear in ePM.

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.

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.**
The explorer tree displays a new environment, and the Environment Editor opens. Optionally, you can rename the environment to something meaningful.
- 2 **On the context menu, click New Environment.**
- 3 **In the environment explorer tree, right-click the new environment.**
- 4 **On the context menu, point to 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.
- 5 **In the environment explorer tree, right-click the new Logical Host.**
- 6 **On the context menu, point to 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.
- 7 **If appropriate, right-click the Integration Server, click Properties, and configure its parameters as needed for use at your site.**
- 8 **In the environment explorer tree, right-click the new Logical Host.**
- 9 **On the context menu, point to 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.
- 10 **If appropriate, right-click the JMS IQ Manager, click Properties, and configure its parameters as needed for use at your site.**

▼ Creating and configuring the Oracle external

- 1 **In the environment explorer tree, right-click the environment.**
- 2 **On the context menu, point to New and click Oracle External System.**

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

4 Configure the Oracle external with the values for your eXchange database instance.

For details on configuring the Oracle external, see the *Oracle eWay User's Guide*. For sample settings typical of an eXchange database, see [Figure 4-12](#).

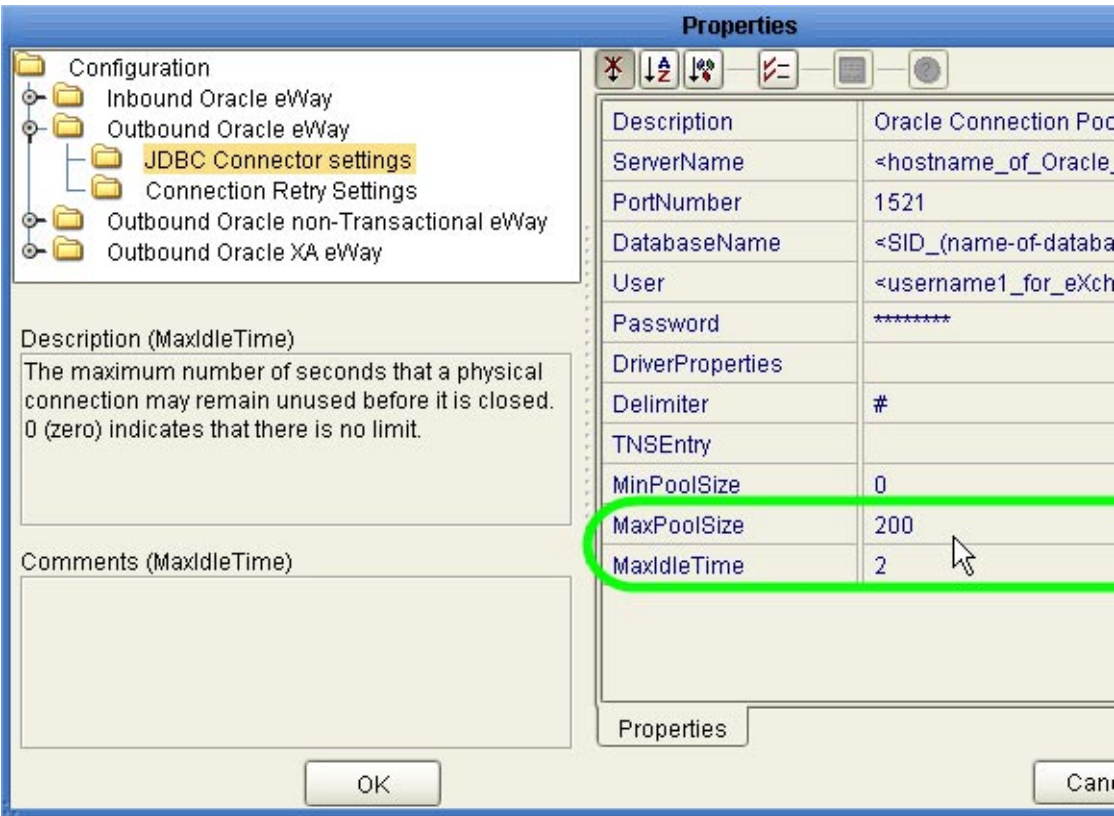


FIGURE 4-12 Environment Configuration for Oracle External System

- 5 In the environment explorer tree, right-click the environment.
- 6 On the context menu, point to New and click B2BServiceConfigurator External System.
- 7 In the Please input the name dialog, enter a meaningful name and click OK.

- 8 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.

See Figure 4–13.

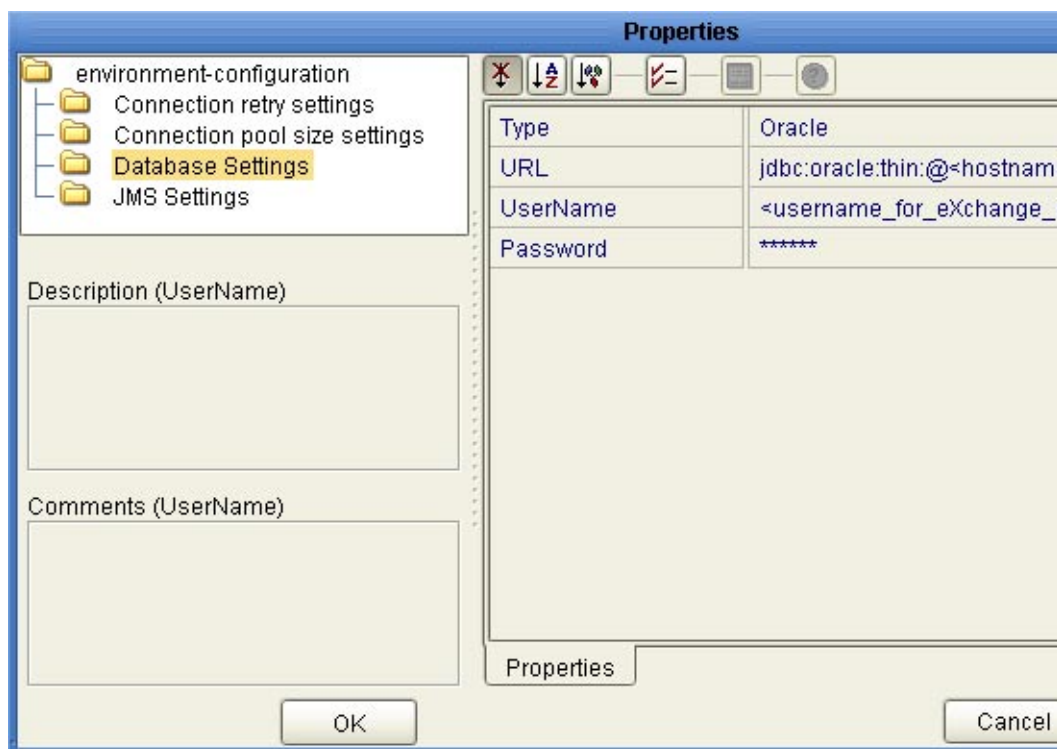


FIGURE 4–13 Environment Configuration for Oracle External System

▼ Creating and configuring LDAP and other externals

- 1 In similar fashion, create an external system for LDAP, name it appropriately, and configure it appropriately for your operating environment.
- 2 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 94](#).

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. For more information, See the *eWay HTTPS Adapter User's Guide*. For an example of settings specific to eXchange Integrator, see the [“To configure the HTTP external to use SSL” on page 92](#).
- To enable the Integration Server to communicate using HTTPS, it needs to be associated with appropriate configuration settings. For more information, see the *eGate Integrator System Administration Guide*.

▼ 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`.
For *TrustStoreType*, keep the default: JKS. See [Figure 4–14](#).

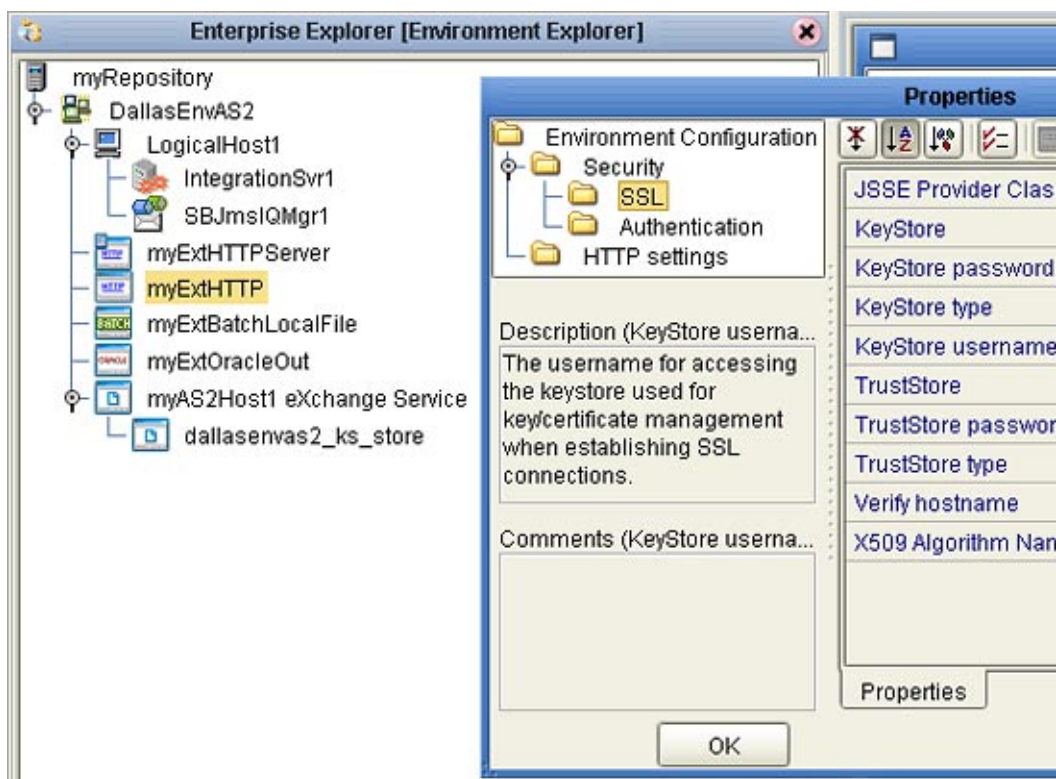


FIGURE 4-14 Configuring the SSL Properties of the HTTP External

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`
 - Also make other configuration changes as needed. For more information, see the *eWay HTTPS Adapter User's Guide*.
 - When you are finished, click OK.

Note – 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.

Connecting the B2B Host to Oracle and LDAP External

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.
- 2 On the context menu, point to 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.
- 3 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.
- 4 From the project tree, drag your B2B Host onto the left side of the canvas
- 5 From the toolbar, do the following:
 - Drag an Oracle external onto the right side of the canvas.
 - Drag and LDAP external onto the right side of the canvas.

Optionally, you can rename components to something meaningful, and then connect them. See [Figure 4–15](#).

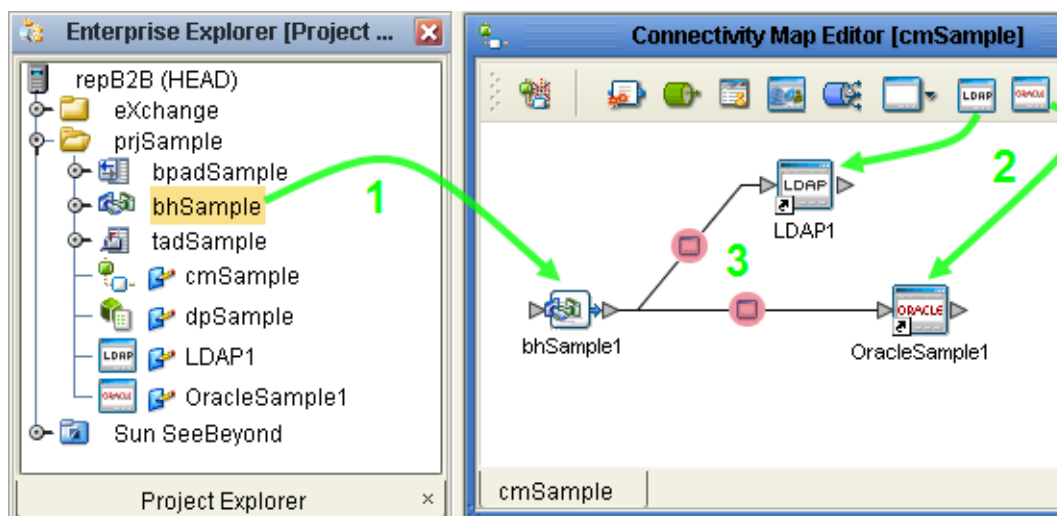


FIGURE 4-15 Connectivity Map Showing B2B Host Connected to Externals

- 6 For the B2B-LDAP connection: double-click the pink dot.
- 7 When the Properties dialog box appears, make no changes, but click OK to set the default configuration.
- 8 For the B2B-Oracle connection: double-click the pink dot and configure the Oracle eWay as outbound.

See [Figure 4-16](#).

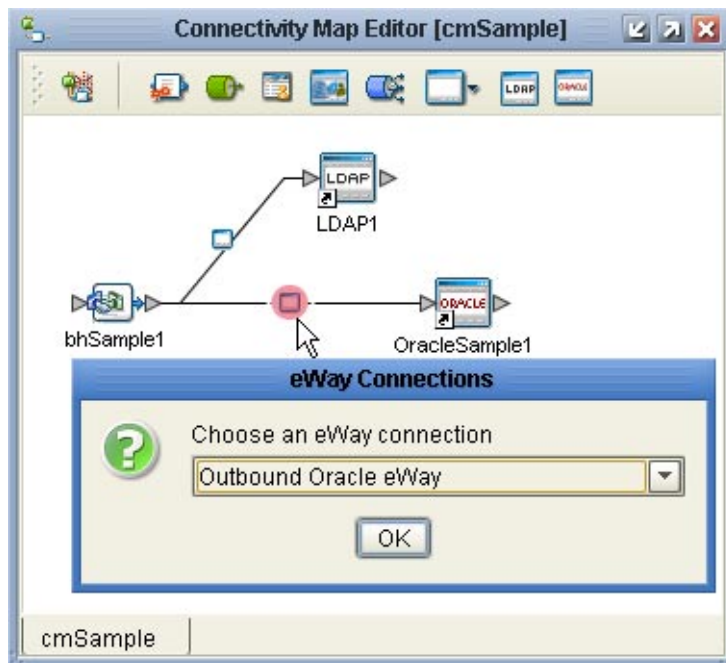


FIGURE 4-16 Configuring the Oracle External as Outbound

- 9 When the Properties dialog box appears, make no changes, but click OK to set the default configuration.

Now that the map is populated and configured, you are ready to build the B2B Host.

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.
- 2 On the context menu, point to New and click Deployment Profile.
- 3 In the Create Deployment Profile dialog box, rename the deployment profile to something meaningful, point it at the correct Environment.

Make sure that it is referencing the Connectivity Map that contains the B2B Host. See [Figure 4-17](#).

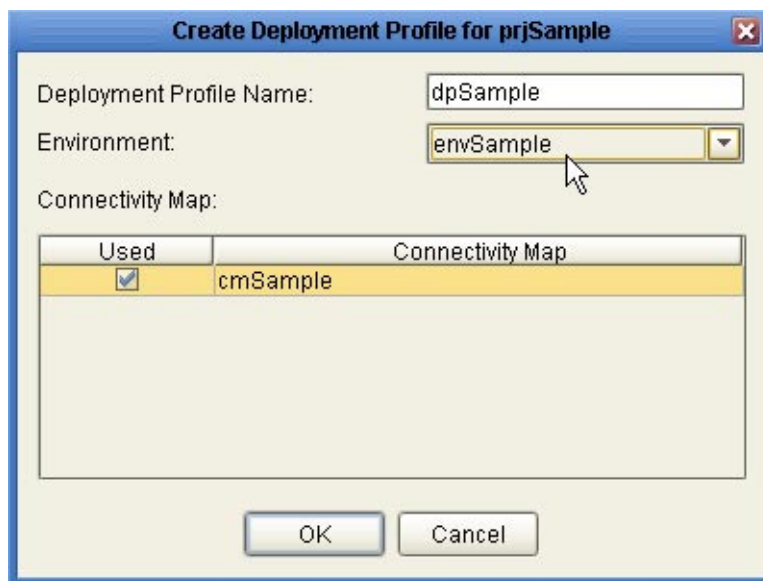


FIGURE 4-17 Creating the Deployment Profile for the B2B Host Project

- 4 In the Deployment Editor, click Automap to map the three components to their respective external systems.

See [Figure 4-18](#).

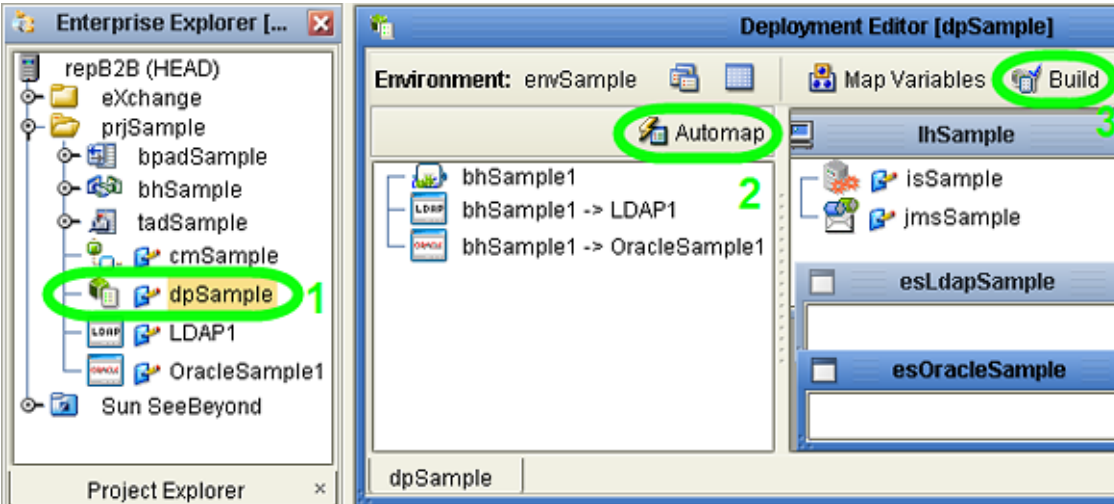


FIGURE 4-18 Deployment Profile Before Components Are Mapped to Externals

In the Environment targeted by the Deployment Profile, an eXchangeService is added that contains all the metadata of the B2B Host. See [Figure 4-19](#).

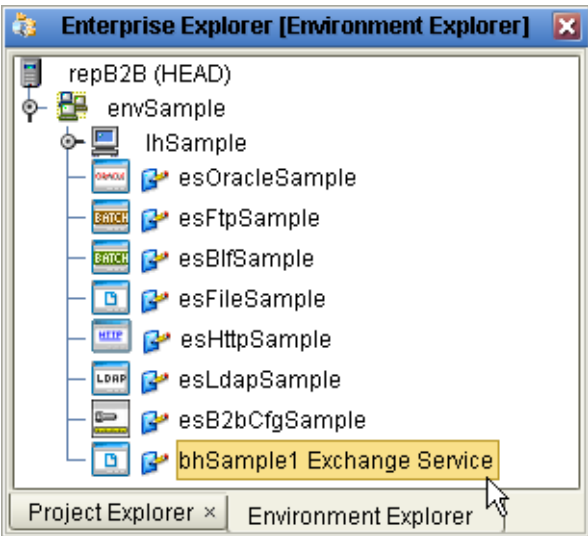


FIGURE 4-19 Environment with Newly Created eXchangeService

- 5 Click Build.
- See [Figure 4–20](#).

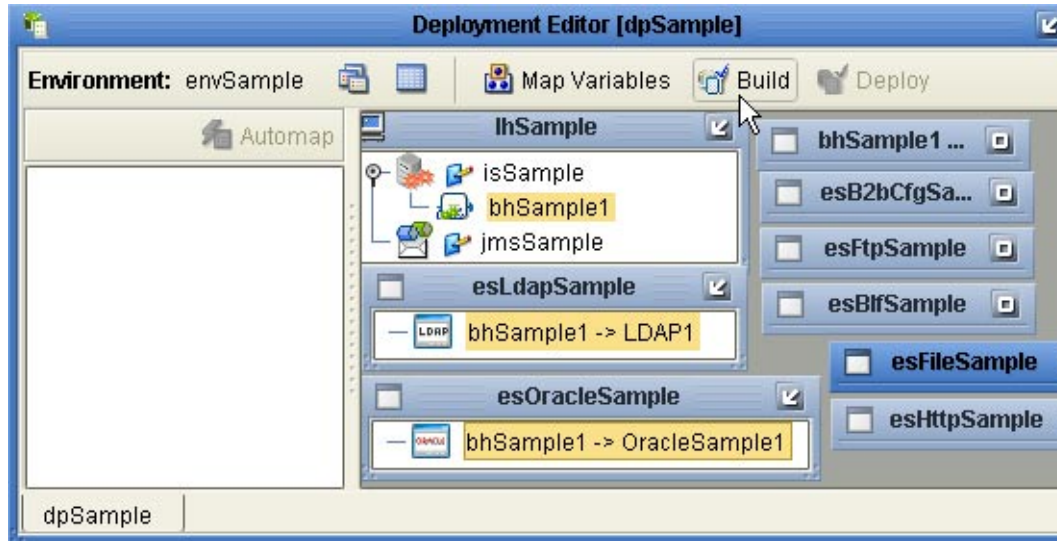


FIGURE 4–20 Deployment Profile Build in Progress

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 4–21](#).

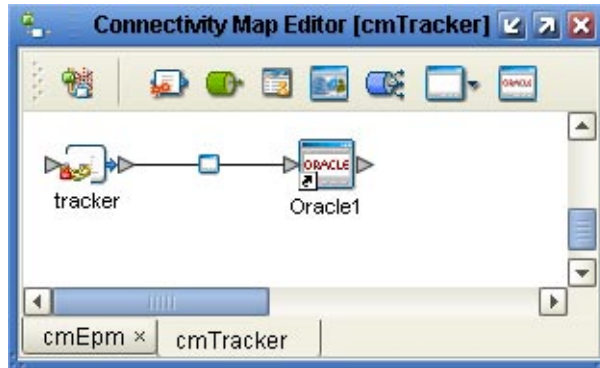


FIGURE 4-21 Connectivity Maps for ePM and Tracker

▼ Before you begin

- 1 With the Project Explorer tab active, in the project tree, open the eXchange⇒GUI project.
- 2 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.
- 3 On the context menu, point to New and click Deployment Profile.
- 4 In the Create Deployment Profile dialog box, name the Deployment Profile to dpEpm, and point it at the correct Environment.
- 5 Make sure that it is referencing cmEpm (the Connectivity Map for the ePM GUI), and then click OK.

See [Figure 4-22](#).

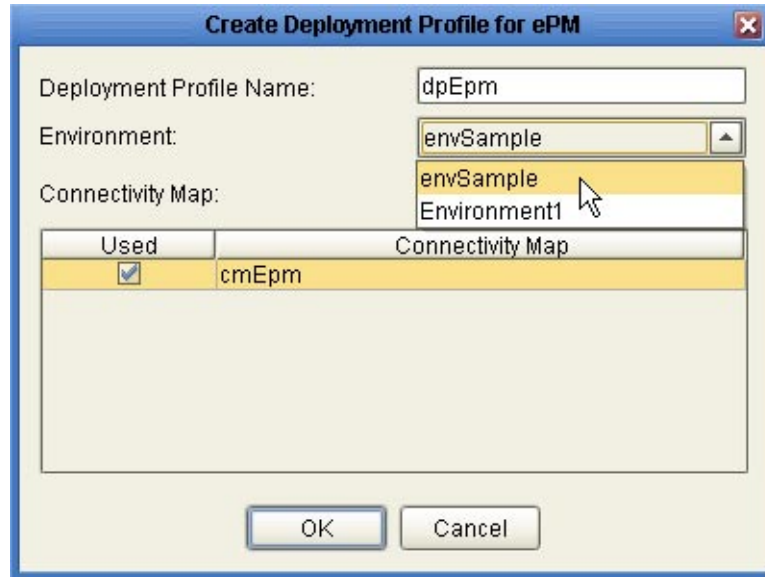


FIGURE 4-22 Deployment Profile for ePM Project

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

See Figure 4-23.

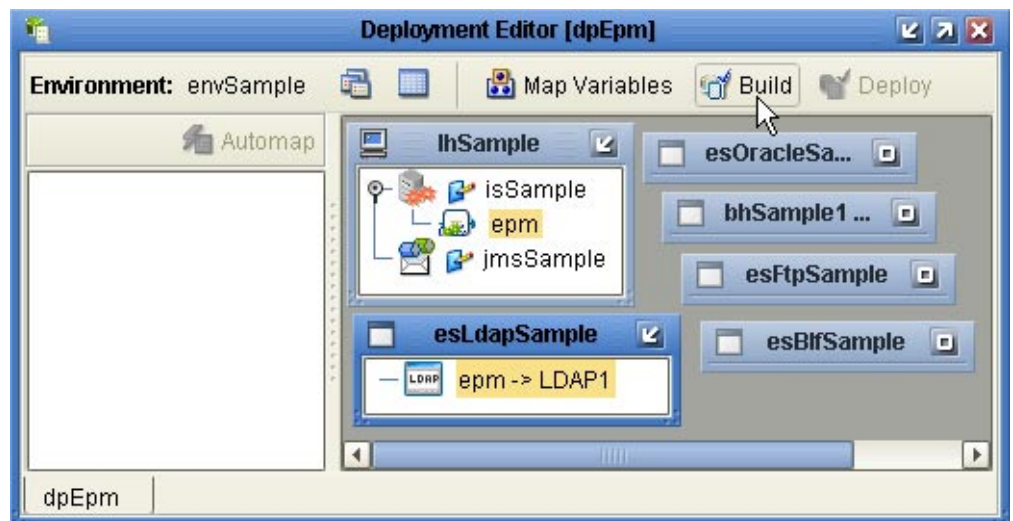


FIGURE 4-23 Deployment Profile dpEpm Being Built

- 7 After the build step completes successfully (creating the EAR file), click Deploy.

The ePM GUI project is built and deployed. You can now use eXchange Partner Manager. 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.
- 2 On the context menu, point to New and click Deployment Profile.
- 3 In the Create Deployment Profile dialog box, name the Deployment Profile to dpTracker and point it at the correct Environment.
- 4 Make sure that it is referencing cmTracker (the Connectivity Map for the Message Tracker GUI), and then click OK.
- 5 In the Deployment Editor, click Automap to map the two components to their respective external systems, and then click Build.
- 6 After the build step completes successfully (creating the EAR file), click Deploy.

See [Figure 4–24](#).

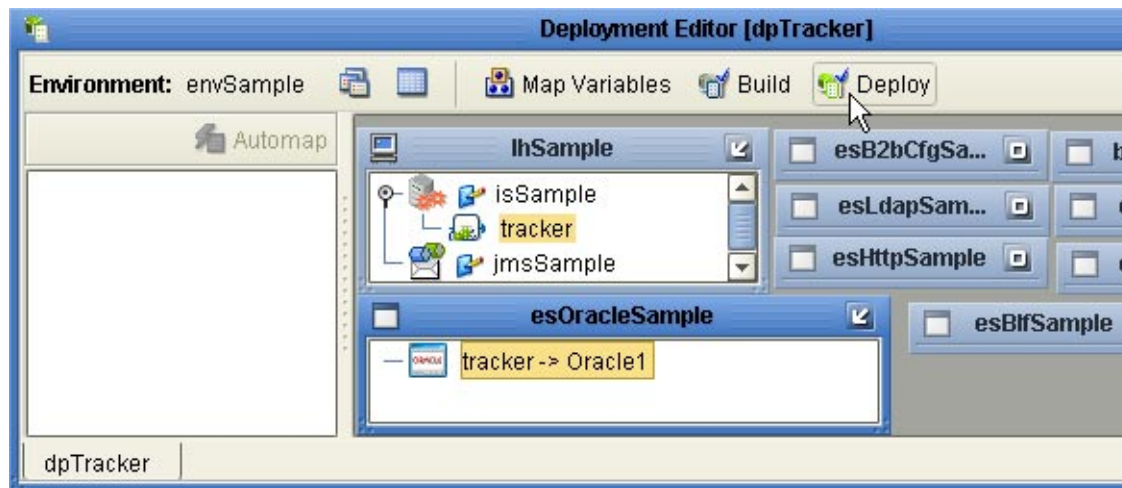


FIGURE 4–24 Deployment Profile dpTracker Being Deployed

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

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.

▼ Before you begin

- 1 With the Project Explorer tab active, in the project tree, open the eXchange⇒Errors project.
- 2 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 and make sure that its File eWay is configured appropriately for your operating environment.
- 2 Right-click the Sub_DLQ project.
- 3 On the context menu, point to New and click Deployment Profile
- 4 In the Create Deployment Profile dialog box, name the Deployment Profile to dpErrorsDLQ and point it at the correct Environment.
- 5 Make sure that it is referencing cmSub_DLQ (the Connectivity Map for the “dead letter” topic), and then click OK.

See [Figure 4–25](#).

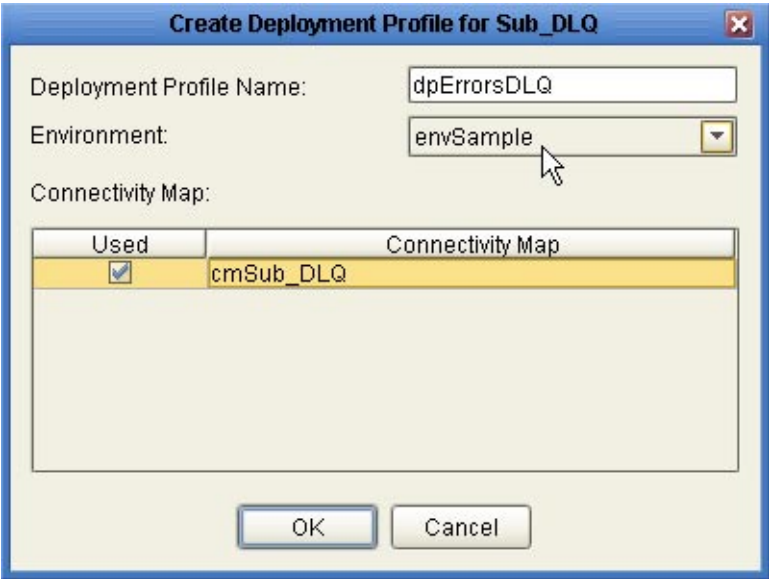


FIGURE 4-25 Deployment Profile for Sub_DLQ Project

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

See Figure 4-26.

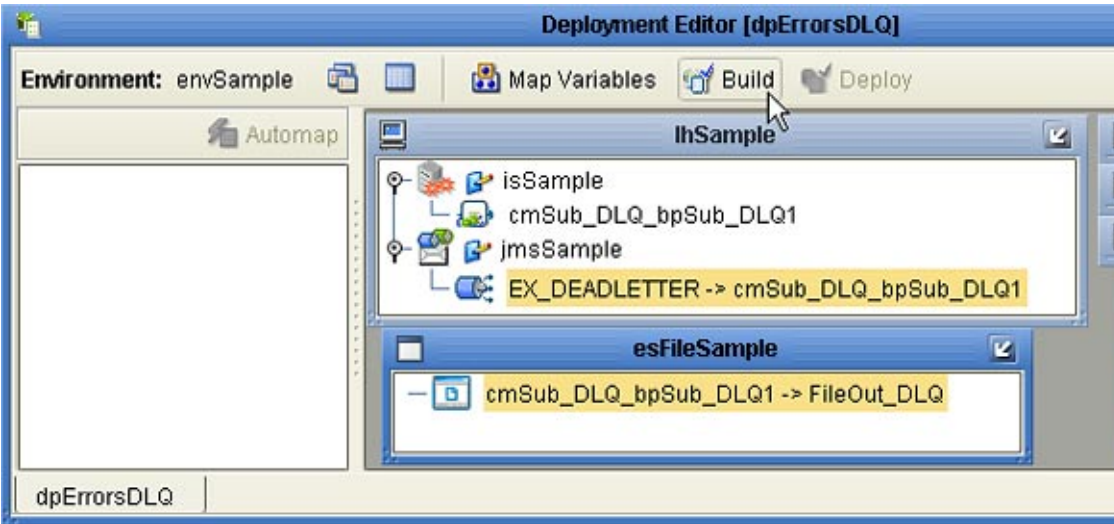


FIGURE 4-26 Deployment Profile dpErrorsDLQ Being Built

- 7 After the build step completes successfully (creating the EAR file), click Deploy.

The Sub_DLQ project is built and deployed. At runtime, any undeliverable messages that cannot be processed are 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 and make sure that its File eWay is configured appropriately for your operating environment.
- 2 Right-click the Sub_ProcErrors project.
- 3 On the context menu, point to New and click Deployment Profile.
- 4 In the Create Deployment Profile dialog box, name the Deployment Profile to dpErrorsProc and point it at the correct Environment.
- 5 Make sure that it is referencing cmSub_ProcErrors (the Connectivity Map for the “processed errors” topic), and then click OK.
- 6 In the Deployment Editor, click Automap to map the two components to their respective external systems, and then click Build.
- 7 After the build step completes successfully (creating the EAR file), click Deploy.

See [Figure 4–27](#).

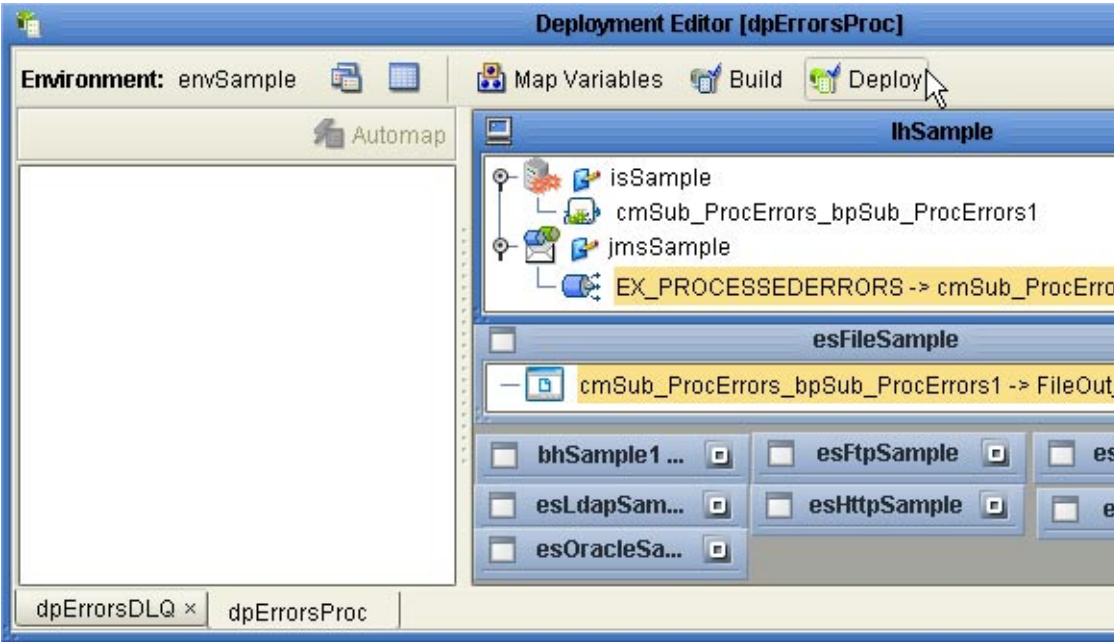


FIGURE 4-27 Deployment Profile dpErrorsProc Being Deployed

The Sub_ProcErrors project is built and deployed. At runtime, undeliverable messages written to the processed-errors topic are copied and written to a file.

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 Tracker.

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 99](#).

What’s in This Chapter

This chapter covers the following information:

General Topics

- [“Providing Access to B2B Web Facilities” on page 108](#)

Topics Specific to eXchange Partner Manager (ePM)

- [“eXchange Partner Manager \(ePM\) Overview” on page 112](#)
- [“Basic Operations in ePM” on page 115](#)
- [“Viewing and Setting Business Protocol Parameters” on page 122](#)
- [“Viewing and Setting Delivery Protocol Parameters” on page 136](#)
- [“Creating and Configuring Transaction Profiles” on page 140](#)
- [“Inheriting and Overriding Parameter Settings” on page 144](#)
- [“Creating and Configuring Trading Partners” on page 153](#)

Topics Specific to Message Tracker

- [“Message Tracker” on page 163](#)

Providing Access to B2B Web Facilities

Access to ePM and Message Tracker 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*.

▼ To add or set up user access to B2B web facilities

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.

- 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 5–1](#).

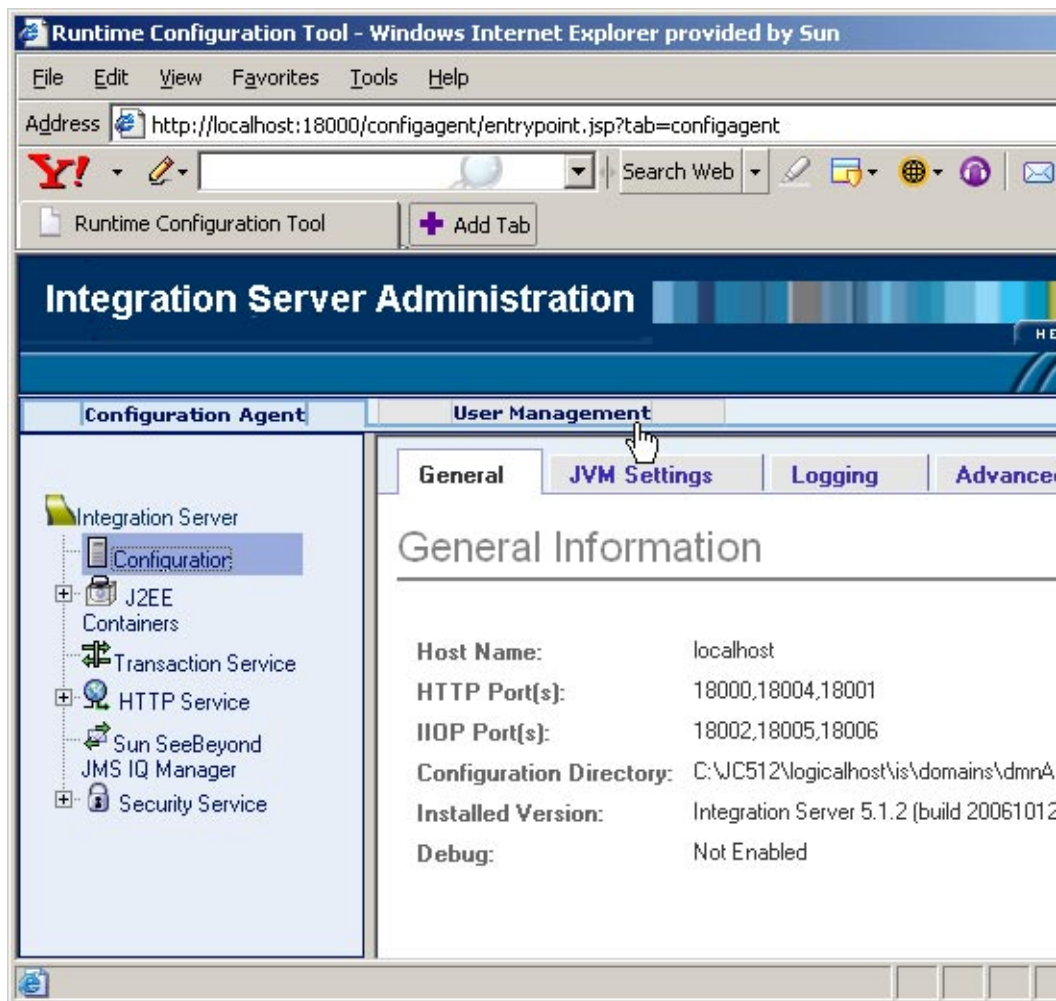


FIGURE 5-1 Integration Server Administration: User Management

- 3 In the User Management screen, either add a new user or edit an existing user.
- 4 In the Add/Edit User screen), do the following:
 - a. Supply a username.
 - b. Supply a valid password for the username.
- 5 For Group List, in addition to any other group privileges for this user, enter either or both of the following

- PartnerManager
- MessageTracking

6 When you are finished, click Submit.

See [Figure 5–2](#).

The screenshot shows a web browser window titled "Runtime Configuration Tool - Windows Internet Explorer provided by Sun". The address bar shows "http://localhost:18000/configagent/entrypoint.jsp?tab=configagent". The main heading is "Integration Server Administration". Below the heading, there are two tabs: "Configuration Agent" and "User Management", with the latter being selected and circled in green. The text "Current realm for user management is" is followed by a dropdown menu showing "file" and a "Change" button. Below this, there is a link "Users List" followed by "> Add/Edit User". The section is titled "Specify the details for this User." and contains several input fields: "User Name: *" with the value "userA", "Password: *" with masked characters, "Confirm Password: *" with masked characters, and "Group List: *" with the value "PartnerManager, MessageTracking". The "Group List" field is circled in green, and a note below it says "Separate multiple groups with commas." At the bottom, there are two buttons: "Submit" (circled in green with a mouse cursor) and "Reset".

Runtime Configuration Tool - Windows Internet Explorer provided by Sun

File Edit View Favorites Tools Help

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

Integration Server Administration

Configuration Agent **User Management**

Current realm for user management is file Change

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

Specify the details for this User.

User Name: *

Password: *

Confirm Password: *

Group List:

Separate multiple groups with commas.

FIGURE 5-2 Specifying PartnerManager and MessageTracking Privileges

- 7 Repeat this procedure for any other users who need access to ePM or Message Tracker for this domain.

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 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.

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 5-3](#).

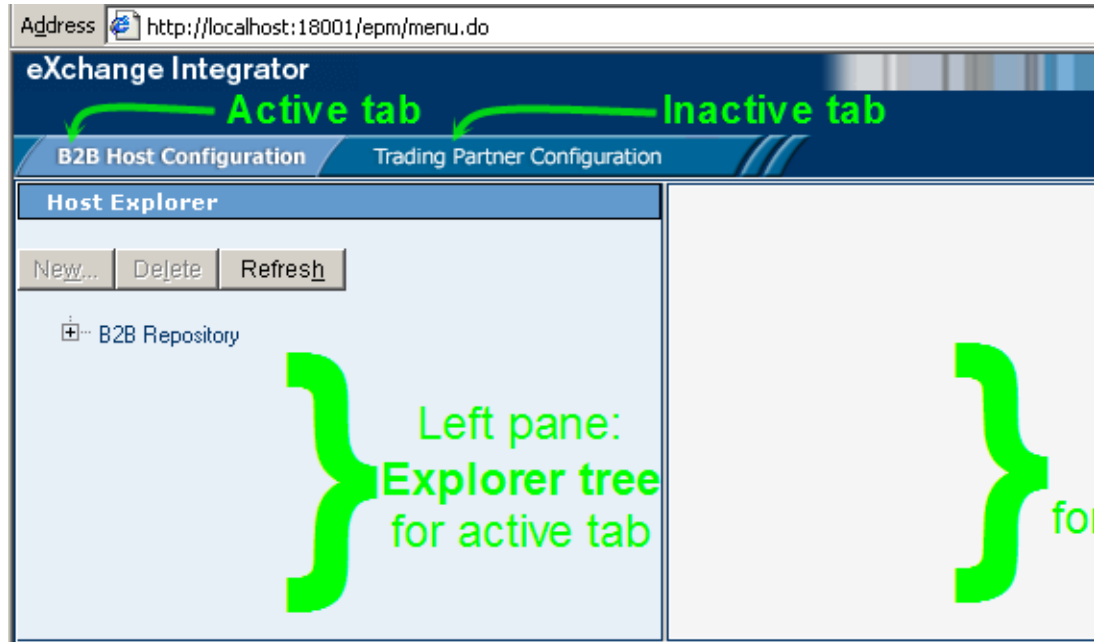


FIGURE 5-3 eXchange Partner Manager (ePM) — Initial State

- 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.

Structural Overview of the ePM GUI

The hierarchy of the Host Explorer tree in the left pane is structure as shown in [Figure 5–4](#).

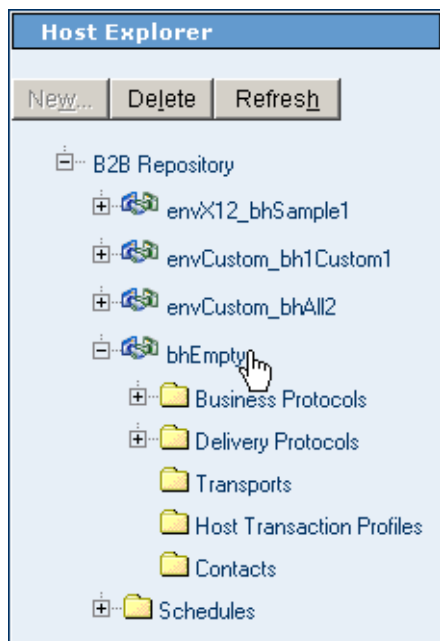


FIGURE 5–4 ePM Structure of the Host Explorer

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

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

- 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 profiles. Each transaction profile contains an outbound transaction profile 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 [“Basic B2B Host Operations in ePM” on page 118](#)). Best practice is to use this area to describe the item, and also to log every important change made to it.

Basic Operations in ePM

This section covers the following topics basic operations in ePM such as accessing, locating a B2B Host in the explorer tree, renaming, importing, and exporting a B2B Host or Trading Partner, and working with schedules:

Accessing ePM

This section describes how to access the eXchange Partner Manager.

▼ To access 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 88](#).

- You must have already built and deployed the ePM GUI project. See [“Building and Deploying the GUI Projects” on page 99](#).
- The Integration Server (domain) and the LDAP server for the associated eXchangeService must already be running.

Note – the Repository does not need to be running.

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.

After a pause, the browser displays the eXchange Partner Manager sign-in screen. See [Figure 5–5](#).

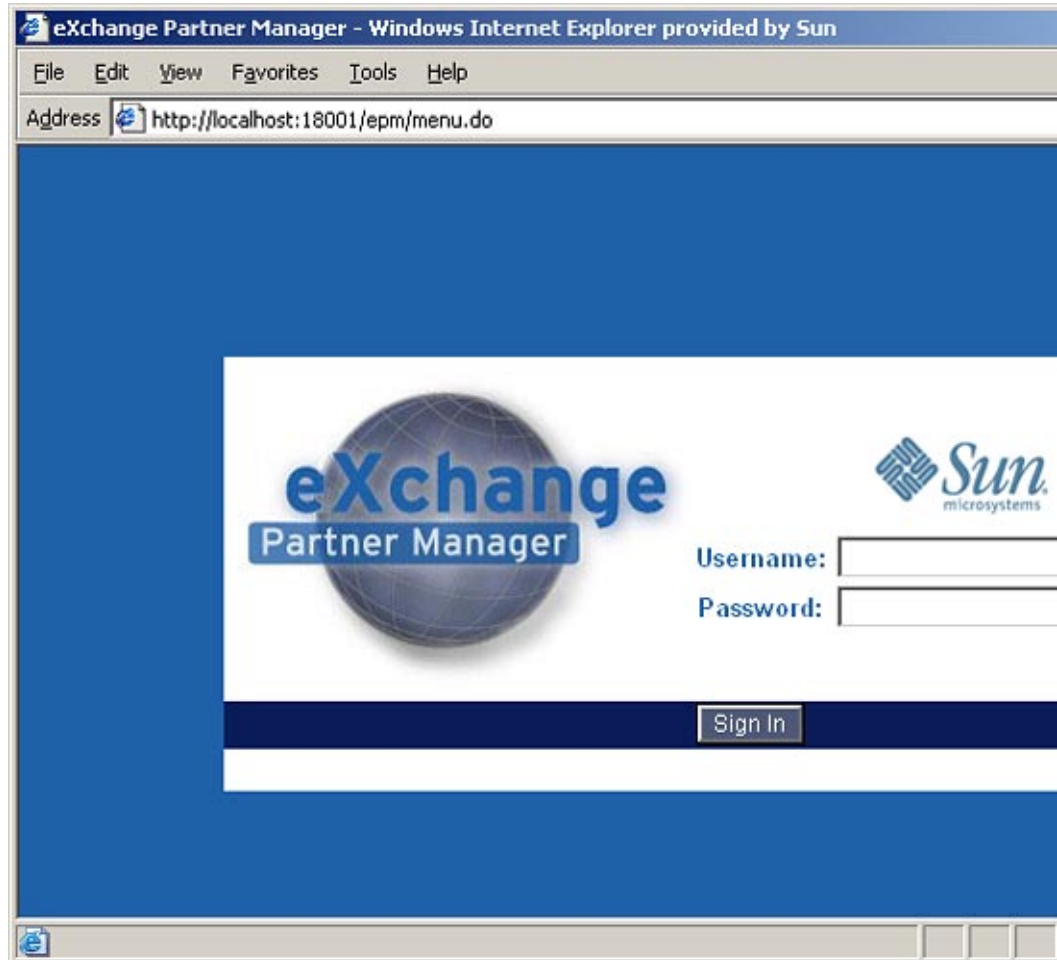


FIGURE 5-5 Sign-In Screen for eXchange Partner Manager (ePM)

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

Note – Username/password combinations are set up in Integration Server Administration; see [“Providing Access to B2B Web Facilities” on page 108](#). The only users authorized to access ePM are those that have a “PartnerManager” entry in their Group List.

After a pause, the window displays eXchange Partner Manager (ePM), a two-pane window with an explorer tree and a canvas. See [Figure 5-6](#).



FIGURE 5-6 eXchange Partner Manager (ePM) — Initial State

Basic B2B Host Operations in ePM

Some operations in ePM are not tied to any particular B2B Host or Trading Partner, for example, 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.
See [Figure 5-7](#).

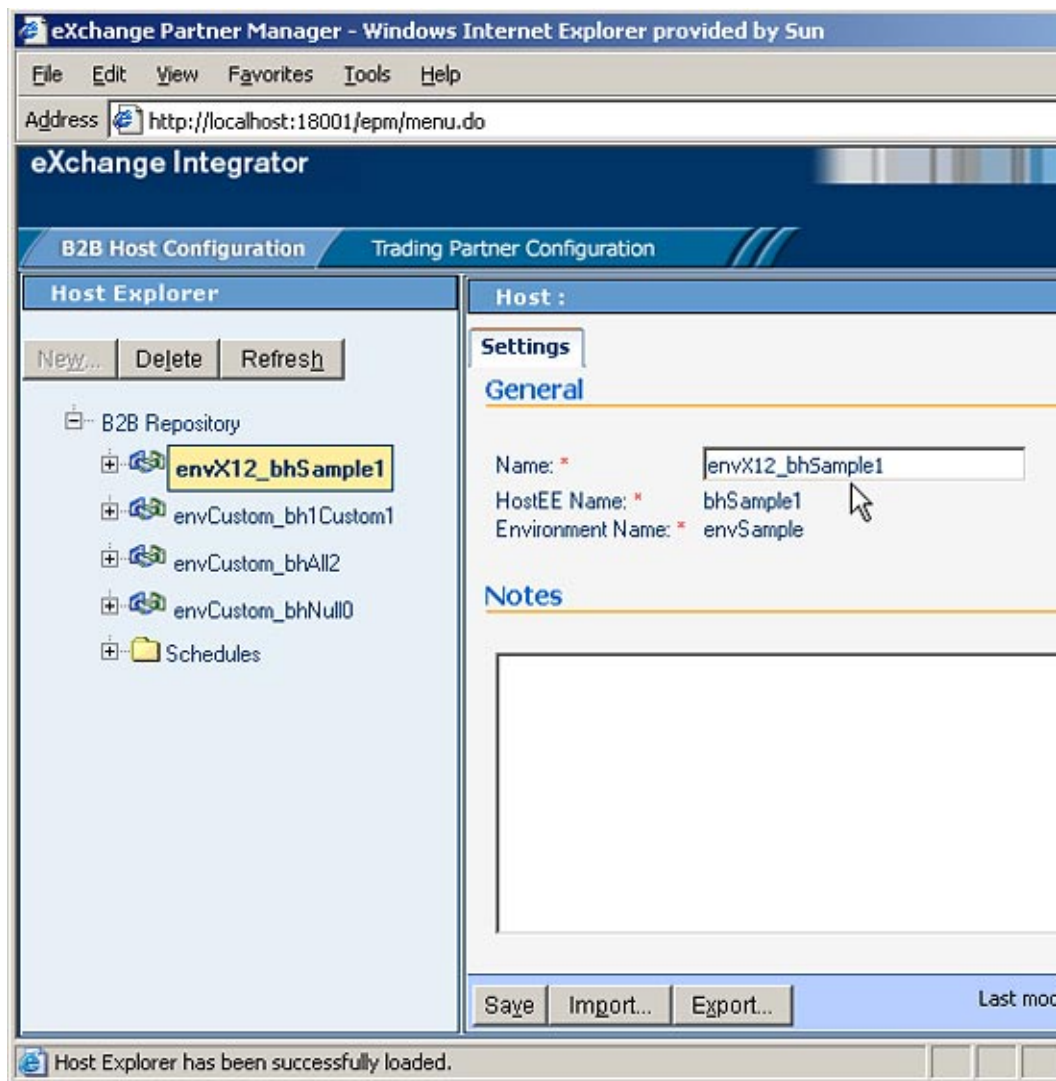


FIGURE 5-7 ePM Repository Showing B2B Hosts

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.

- 2 In the right pane, under the Settings tab, in the General parameters section, edit the value of Name.
- 3 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.
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, click Browse.
- 4 Navigate to and open the .exp file you want to import, and then click Import.

See [Figure 5–8](#)

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 Browse...

Import... Cancel

FIGURE 5–8 Importing a B2B Host

5 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 5–8](#)

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

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.

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 126](#).
- *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 130](#).
- *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 133](#).
- *Batch - Inbound* and *ACK - Batch-Inbound*. Parameters determined by enveloping attribute definitions (EADs) whose direction is FromPartner or Both.

Note – Although the PassThrough business protocol does not supply any batch-inbound parameters, other business protocols might do so.

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 5–9](#).

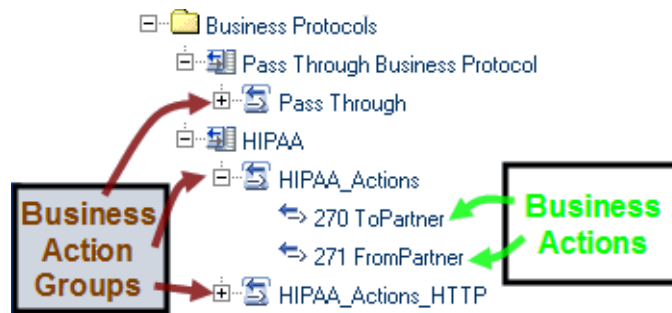


FIGURE 5–9 Business Action Groups and Business Actions

▼ 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 5–10](#).

Host Business Action Group :

Settings

General

Host Business Action Group Name: * 850-855

Notes

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

Save

FIGURE 5-10 Creating a New Business Action Group

4 Click Save.

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 are 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 5–11.

Host Business Action :

Settings

General

Host Business Action Name: * X12_850_Out

Direction: * FromPartner

ToPartner

FromPartner

Notes

Outbound 850 (Purchase Order Request).

Save

FIGURE 5–11 Creating a New Business Action

- 4 Click Save.
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 are 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.

Protocol-Outbound (ToPartner) Parameter and ACK Settings

This section lists and describes business protocol-outbound=ToPartner parameter settings and corresponding acknowledgment settings. See [Figure 5-12](#).

Host Business Protocol :	
Settings General Business Protocol: * Pass Through Business Protocol Properties	
<div>Protocol - Outbound ToPartner</div> <div> Business - Duplication Checking: * <input type="text" value="No"/> </div> <div> Business - Use Encryption: * <input type="text" value="No"/> </div> <div> Business - Use Signature: * <input type="text" value="No"/> </div> <div> Business - Use Compression: * <input type="text" value="No"/> </div> <div> Business - Expect Acknowledgements: * <input type="text" value="No"/> </div> <div> Business - Character Set Encoding: <input type="text"/> </div> <div> Business - Message Encoding: <input type="text"/> </div> <div> Business - Content Type: <input type="text"/> </div> <div> Business - Maximum Number of Resends: <input type="text"/> </div> <div> Business - Minutes Between Resends: <input type="text"/> </div>	
<div>Batch - Outbound ToPartner</div>	
<div>Protocol - Inbound FromPartner</div>	
<div>ACK - Protocol - Outbound ToPartner</div> <div> Business - Use Encryption: * <input type="text" value="No"/> </div> <div> Business - Use Signature: * <input type="text" value="No"/> </div> <div> Business - Use Compression: * <input type="text" value="No"/> </div> <div> Business - Character Set Encoding: <input type="text"/> </div> <div> Business - Message Encoding: <input type="text"/> </div> <div> Business - Content Type: <input type="text"/> </div>	
<div>ACK - Batch - Outbound ToPartner</div>	
<div>ACK - Protocol - Inbound FromPartner</div>	
<input type="button" value="Save"/>	Last modified:

FIGURE 5-12 Protocol-Outbound (ToPartner) Parameter and ACK Settings
Chapter 5 • Using B2B Web Facilities

TABLE 5-1 Business-Duplication Checking

Description	Settings	Default
Whether or not outbound messages are checked for duplication.	Yes or No	No

TABLE 5-2 Business-Use Encryption

Description	Settings	Default
Whether or not outbound messages are encrypted	Yes or No	No

TABLE 5-3 Business-Use Signature

Description	Settings	Default
Whether or not outbound messages are electronically signed for authentication	Yes or No	No

TABLE 5-4 Business-Use Compression

Description	Settings	Default
Whether or not outbound messages are compressed	Yes or No	No

TABLE 5-5 Business-Expect Acknowledgments

Description	Settings	Default
Whether or not the host is expecting acknowledgement for the request/response	Yes or No	No

TABLE 5-6 Business - Character Set Encoding

Description	Settings	Default
The character set used to encode the text of the outgoing message; optional Leave blank to use the default encoding	A string representing a valid character set encoding, such as ASCII or UTF-8) (or empty)	(empty)

TABLE 5-7 Business - Content Type

Business - Content Type

Description	Settings	Default
(reserved)	(reserved)	(empty)

TABLE 5-8 Business - Maximum Number of Resends

Description	Settings	Default
<p>The maximum number of resend tries permitted (optional)</p> <p>Does not apply to acknowledgments</p>	<p>Nonnegative integer; however, if you are expecting a response or acknowledgment, set this to a positive value</p> <p>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 causes an exception to be thrown</p> <p>To turn off business responses only, make sure the acknowledgement value is set to any nonnegative integer, then set the reply time out to 0.</p>	(empty)

TABLE 5-9 Business - Minutes Between Resends

Description	Settings	Default
<p>Number of minutes to wait between resends (optional)</p> <p>Does not apply to acknowledgments</p>	<p>Nonnegative integer; however, if you are expecting a response or acknowledgment, set this to a positive value</p> <p>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 causes an exception to be thrown</p>	(empty)

Protocol-Inbound (FromPartner) Parameter and ACK Settings

This section lists and describes business protocol-inbound FromPartner parameter settings and corresponding acknowledgment settings. See [Figure 5-13](#).

Host Business Protocol :	
Settings General Business Protocol: * Pass Through Business Protocol Properties	
<div> <div>+</div> <div>Protocol - Outbound ToPartner</div> </div>	
<div> <div>+</div> <div>Batch - Outbound ToPartner</div> </div>	
<div> <div>-</div> <div>Protocol - Inbound FromPartner</div> </div> <div> Business - Duplication Checking: * <input type="text" value="No"/> </div> <div> Business - Use Decryption: * <input type="text" value="No"/> </div> <div> Business - Verify Signature: * <input type="text" value="No"/> </div> <div> Business - Use Decompression: * <input type="text" value="No"/> </div> <div> Business - Send Acknowledgements: * <input type="text" value="No"/> </div> <div> Business - Character Set Encoding: <input type="text"/> </div> <div> Business - Message Encoding: <input type="text"/> </div> <div> Business - Content Type: <input type="text"/> </div>	
<div> <div>+</div> <div>ACK - Protocol - Outbound ToPartner</div> </div>	
<div> <div>+</div> <div>ACK - Batch - Outbound ToPartner</div> </div>	
<div> <div>-</div> <div>ACK - Protocol - Inbound FromPartner</div> </div> <div> Business - Duplication Checking: * <input type="text" value="No"/> </div> <div> Business - Use Decryption: * <input type="text" value="No"/> </div> <div> Business - Verify Signature: * <input type="text" value="No"/> </div> <div> Business - Use Decompression: * <input type="text" value="No"/> </div> <div> Business - Character Set Encoding: <input type="text"/> </div> <div> Business - Message Encoding: <input type="text"/> </div> <div> Business - Content Type: <input type="text"/> </div>	
<input type="button" value="Save"/>	Last mod

FIGURE 5-13 Protocol-Inbound (FromPartner) Parameter and ACK Settings

TABLE 5-10 Business-Duplication Checking

Description	Settings	Default
Whether or not inbound messages are checked for duplication.	Yes or No	No

TABLE 5-11 Business-Use Decryption

Description	Settings	Default
Whether or not inbound messages are decrypted	Yes or No	No

TABLE 5-12 Business-Verify Signature

Description	Settings	Default
Whether or not inbound messages are authenticated for a valid electronic signature	Yes or No	No

TABLE 5-13 Business-Use Decompression

Description	Settings	Default
Whether or not inbound messages are decompressed	Yes or No	No

TABLE 5-14 Business-Expect Acknowledgments

Description	Settings	Default
Whether or not inbound messages are acknowledged	Yes or No	No

TABLE 5-15 Business-Character Set Encoding

Description	Settings	Default
The character set used to encode the text of the incoming message (optional)	A string representing a valid character set encoding (such as ASCII or UTF-8), or empty	(empty)
Leave blank to use the default encoding		

TABLE 5-16 Business - Content Type

Description	Settings	Default
(reserved)	(reserved)	(empty)

Batch-Outbound (ToPartner) Parameter and ACK Settings

This section lists and describes business batch-outbound ToPartner parameter settings and corresponding acknowledgment settings. See [Figure 5-14](#).

Host Business Protocol :

Settings

General

Business Protocol: * Pass Through Business Protocol

Properties

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

Save

Last modified at

FIGURE 5-14 Batch-Outbound (ToPartner) Parameter and ACK Settings

TABLE 5-17 Business-Release Quantity

Description	Settings	Default
Specifies the threshold beyond which a batch send is triggered (required)	Nonnegative integer	1
If you want to send messages on a schedule using Release Scheduler String, set Release Quantity to a very high value (for example, 99999999)		

134

IBM Sterling Exchange Integrator User's Guide • December 2007

TABLE 5-18 Business-Release Size

Description	Settings	Default
(reserved)	(reserved)	(empty)

TABLE 5-19 Business-Cron Schedule

Description	Settings	Default
<p>An expression specifying when and how often to run (optional)</p> <p>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 135.</p>	Either a valid cron expression, or empty	(empty)

TABLE 5-20 Business-Reliability Timeout

Description	Settings	Default
Specifies the maximum number of minutes to wait for a reply before attempting a re-send (required)	Nonnegative integer	0

TABLE 5-21 Business-Reliability Maximum Retry Count

Description	Settings	Default
Specifies maximum number of times to retry sending before giving up; required	Nonnegative integer	0

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”.

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.
- 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 140.

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 5–15](#).

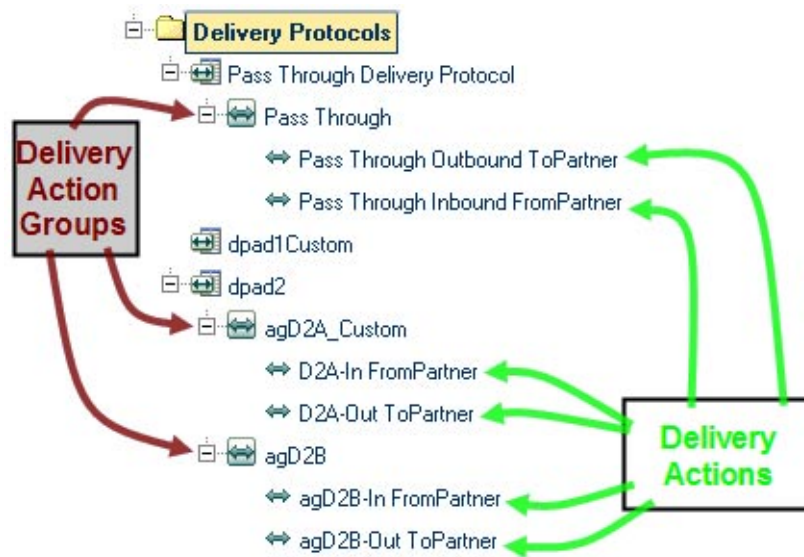


FIGURE 5–15 Delivery Action Groups and Delivery Actions

▼ To create and configure a delivery action group

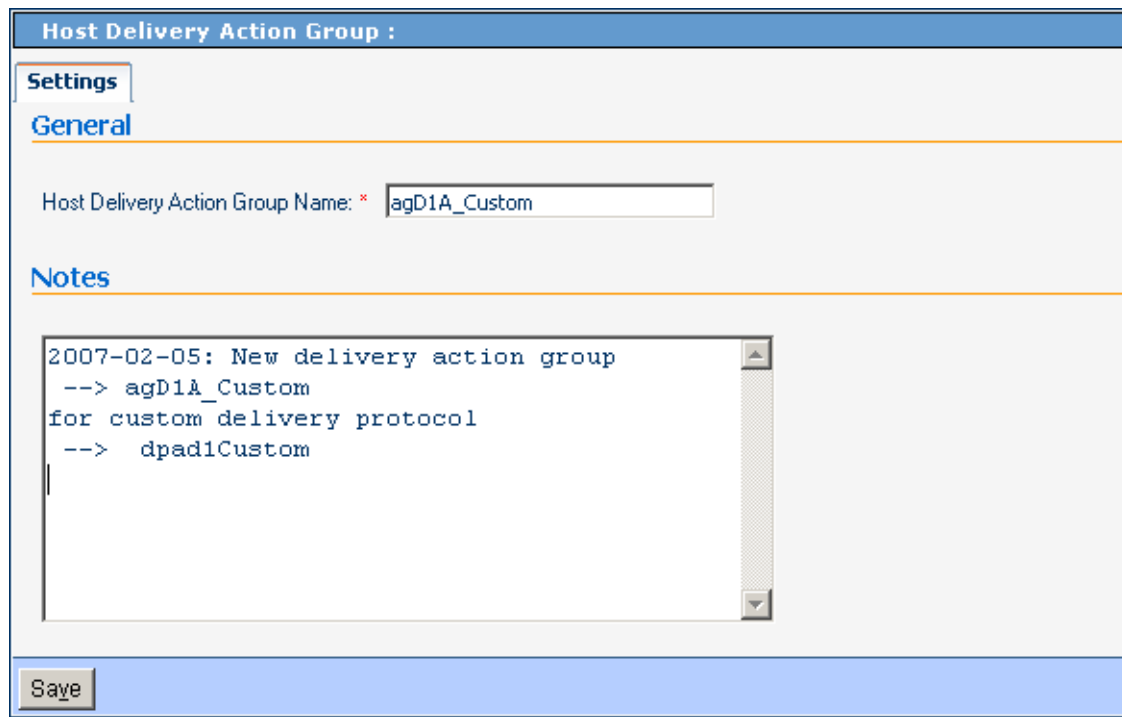
- 1 In the Host Explorer tree, click the delivery protocol under which you want to create an action group.

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 5-16](#).



The screenshot shows a window titled "Host Delivery Action Group :". It has a "Settings" tab selected, with a "General" sub-tab. Under "General", there is a text field labeled "Host Delivery Action Group Name: *" containing the text "agD1A_Custom". Below this is a "Notes" section with a text area containing the following text:

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

At the bottom of the window is a "Save" button.

FIGURE 5-16 Creating a New Delivery Action Group

- 4 Click Save.

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 are 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.
- 4 For Direction, choose either FromPartner or ToPartner.

See [Figure 5–17](#).

Host Delivery Action :

Settings

General

Host Delivery Action Name: *

Direction: *

Notes

FIGURE 5–17 Creating a New Delivery Action

- 5 **Click Save.**
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.
- 6 **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 are inherited by transaction profiles that use this delivery action.
- 7 **Click Save.**
- 8 **Repeat as needed to create additional delivery actions in the same action group.**

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 the following:

- [“Protocol-Outbound \(ToPartner\) Parameter and ACK Settings” on page 126.](#)
- [“Protocol-Inbound \(FromPartner\) Parameter and ACK Settings” on page 130.](#)
- [“Batch-Outbound \(ToPartner\) Parameter and ACK Settings” on page 133 and Table 5–22.](#)

TABLE 5–22 Delivery - Batch Tracking Code

Description	Settings	Default
List of options for tracking message batches; tsure tracking of outbound and inbound batches, retain the default setting	No Tracking or Header Only or Header and Message	Header and Message

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

- 1 In the Host Explorer tree, open the Host for the transaction profile, click the Host Transaction Profiles folder and click New.

See Figure 5–18.

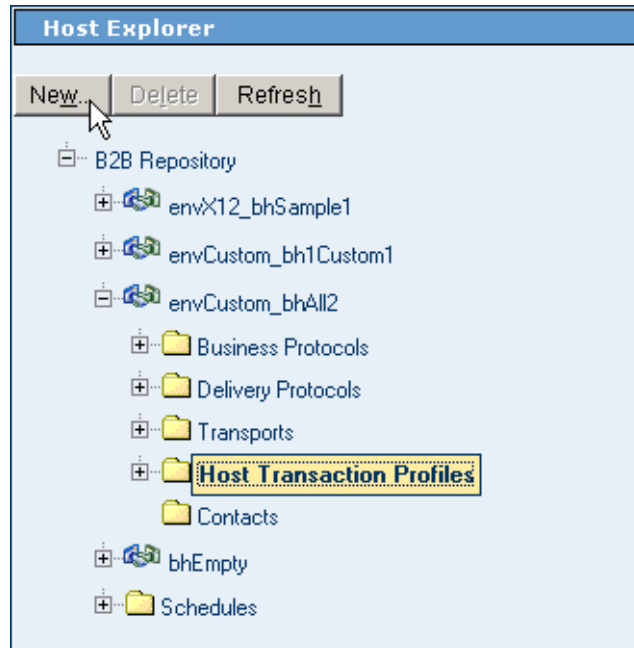


FIGURE 5–18 Creating a New Transaction Profile

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

See Figure 5–19.

New Host Transaction Profile

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

Name: *

Business Action Group: *

- /Pass Through Business Protocol/Pass Through
- /bpad1Custom/850-855
- /bpad1Custom/agB1_Custom
- /bpad2/agB2_Custom

FIGURE 5-19 New Transaction Profile - Name and Business Action Group

- 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 5-20](#).

New Host Transaction Profile, cont'd

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

Business Action	Delivery Action
Pass Through Inbound FromPartner	/Pass Through Delivery Protocol/Pass Through/Pass Through Inbound FromPartner
Pass Through Outbound ToPartner	/Pass Through Delivery Protocol/Pass Through/Pass Through Outbound ToPartner

FIGURE 5-20 New Transaction Profile - Delivery Actions and Transports

Note – You can click Back to change any choices previously made.

5 When you are satisfied with your choices, click Finish.

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

6 Click Save.

Note – You can 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.

7 Repeat as needed to create additional transaction profiles for the same host.

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.

The following applies to 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, where “←” 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

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.

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.

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 5–21](#) and [Figure 5–22](#) show 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.

Host Delivery Protocol :

Settings

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:

FIGURE 5-21 Host Parameters at the Protocol Level

Trading Partner Delivery Protocol :			
Settings		Overrides	
Protocol - Outbound ToPartner			
Property	Current Value	Override?	Inherits From
mad2Int42: *	42	<input type="checkbox"/>	Host Delivery Protocol
mad2Boolean: *	No	<input type="checkbox"/>	Host Delivery Protocol
mad2NumberPi: *	-3.14159	<input checked="" type="checkbox"/>	Host Delivery Protocol
Delivery - Duplication Checking: *	Yes	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Use Encryption: *	Yes	<input checked="" type="checkbox"/>	Host Delivery Protocol
Delivery - Use Signature: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Use Compression: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Expect Acknowledgements: *	Yes	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Character Set Encoding:	UTF-8	<input checked="" type="checkbox"/>	Host Delivery Protocol
Delivery - Message Encoding:		<input type="checkbox"/>	Host Delivery Protocol
Delivery - Content Type:		<input type="checkbox"/>	Host Delivery Protocol

FIGURE 5-22 TP Parameters at the Protocol Level

Figure 5-23 and Figure 5-24 show 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. In this example, 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.

Host Delivery Action Group :				
<div>Settings</div> <div>Overrides</div>				
Protocol - Outbound ToPartner				
Property	Current Value	Override?	Inherits From	Original Value
mad2Int42: *	42	<input type="checkbox"/>	Host Delivery Protocol	42
mad2Boolean: *	No	<input type="checkbox"/>	Host Delivery Protocol	No
mad2NumberPi: *	0.0	<input checked="" type="checkbox"/>	Host Delivery Protocol	3.1415
Delivery - Duplication Checking: *	No	<input checked="" type="checkbox"/>	Host Delivery Protocol	Yes
Delivery - Use Encryption: *	No	<input type="checkbox"/>	Host Delivery Protocol	No
Delivery - Use Signature: *	No	<input type="checkbox"/>	Host Delivery Protocol	No
Delivery - Use Compression: *	No	<input type="checkbox"/>	Host Delivery Protocol	No
Delivery - Expect Acknowledgements: *	Yes	<input type="checkbox"/>	Host Delivery Protocol	Yes
Delivery - Character Set Encoding:		<input type="checkbox"/>	Host Delivery Protocol	
Delivery - Message Encoding:		<input type="checkbox"/>	Host Delivery Protocol	
Delivery - Content Type:		<input type="checkbox"/>	Host Delivery Protocol	

FIGURE 5-23 Host Parameters at the Action Group Level

Trading Partner Delivery Action Group :			
Settings		Overrides	
Protocol - Outbound ToPartner			
Property	Current Value	Override?	Inherits From
mad2Int42: *	4242	<input checked="" type="checkbox"/>	Host Delivery Protocol
mad2Boolean: *	No	<input type="checkbox"/>	Host Delivery Protocol
mad2NumberPi: *	-3.14159	<input type="checkbox"/>	Trading Partner Delivery
Delivery - Duplication Checking: *	No	<input type="checkbox"/>	Host Delivery Action Gro
Delivery - Use Encryption: *	Yes	<input type="checkbox"/>	Trading Partner Delivery
Delivery - Use Signature: *	Yes	<input checked="" type="checkbox"/>	Host Delivery Protocol
Delivery - Use Compression: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Expect Acknowledgements: *	Yes	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Character Set Encoding:	UTF-8	<input type="checkbox"/>	Trading Partner Delivery
Delivery - Message Encoding:		<input type="checkbox"/>	Host Delivery Protocol
Delivery - Content Type:		<input type="checkbox"/>	Host Delivery Protocol

FIGURE 5-24 TP Parameters at the Action Group Level

Figure 5-25 and Figure 5-26 show 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. In this example, 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.

Host Delivery Action :			
<div>Settings</div> <div>Overrides</div>			
Protocol - Outbound ToPartner			
Property	Current Value	Override?	Inherits From
mad2Int42: *	-42	<input checked="" type="checkbox"/>	Host Delivery Protocol
mad2Boolean: *	Yes	<input checked="" type="checkbox"/>	Host Delivery Protocol
mad2NumberPi: *	3.14	<input checked="" type="checkbox"/>	Host Delivery Action Group (ag)
Delivery - Duplication Checking: *	No	<input type="checkbox"/>	Host Delivery Action Group (ag)
Delivery - Use Encryption: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Use Signature: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Use Compression: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Expect Acknowledgements: *	Yes	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Character Set Encoding:		<input type="checkbox"/>	Host Delivery Protocol
Delivery - Message Encoding:		<input type="checkbox"/>	Host Delivery Protocol
Delivery - Content Type:		<input type="checkbox"/>	Host Delivery Protocol

FIGURE 5-25 Host Parameters at the Action Level

Host Delivery Action :			
Settings		Overrides	
Protocol - Outbound ToPartner			
Property	Current Value	Override?	Inherits From
mad2Int42: *	-42	<input checked="" type="checkbox"/>	Host Delivery Protocol
mad2Boolean: *	Yes	<input checked="" type="checkbox"/>	Host Delivery Protocol
mad2NumberPi: *	3.14	<input checked="" type="checkbox"/>	Host Delivery Action Gro
Delivery - Duplication Checking: *	No	<input type="checkbox"/>	Host Delivery Action Gro
Delivery - Use Encryption: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Use Signature: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Use Compression: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Expect Acknowledgements: *	Yes	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Character Set Encoding:		<input type="checkbox"/>	Host Delivery Protocol
Delivery - Message Encoding:		<input type="checkbox"/>	Host Delivery Protocol
Delivery - Content Type:		<input type="checkbox"/>	Host Delivery Protocol

FIGURE 5-26 TP Parameters at the Action Level

Figure 5-27 and Figure 5-28 show 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, is overridden and set to Yes — the same value (see Figure 5-21). 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 does not change.

Host Transaction Profile :			
Settings		Overrides	
+... 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
mad2Int42: *	-424242	<input checked="" type="checkbox"/>	Host Delivery Action (D2A-Out)
mad2Boolean: *	Yes	<input type="checkbox"/>	Host Delivery Action (D2A-Out)
mad2NumberPi: *	3.14	<input type="checkbox"/>	Host Delivery Action (D2A-Out)
Delivery - Duplication Checking: *	Yes	<input checked="" type="checkbox"/>	Host Delivery Action Group (ag)
Delivery - Use Encryption: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Use Signature: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Use Compression: *	No	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Expect Acknowledgements: *	Yes	<input type="checkbox"/>	Host Delivery Protocol
Delivery - Character Set Encoding:		<input type="checkbox"/>	Host Delivery Protocol
Delivery - Message Encoding:	binary	<input checked="" type="checkbox"/>	Host Delivery Protocol
Delivery - Content Type:		<input type="checkbox"/>	Host Delivery Protocol

FIGURE 5-27 Host Parameters at the Transaction Profile Level

Trading Partner Transaction Profile :			
Settings		Overrides	
+ 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
mad2Int42: *	-4242	<input type="checkbox"/>	Trading Partner Delivery
mad2Boolean: *	Yes	<input type="checkbox"/>	Host Delivery Action (D2
mad2NumberPi: *	-3.14159	<input type="checkbox"/>	Trading Partner Delivery
Delivery - Duplication Checking: *	Yes	<input type="checkbox"/>	Host Transaction Profile
Delivery - Use Encryption: *	Yes	<input type="checkbox"/>	Trading Partner Delivery
Delivery - Use Signature: *	Yes	<input type="checkbox"/>	Trading Partner Delivery
Delivery - Use Compression: *	Yes	<input checked="" type="checkbox"/>	Host Delivery Protocol
Delivery - Expect Acknowledgements: *	Yes	<input checked="" type="checkbox"/>	Host Delivery Protocol
Delivery - Character Set Encoding:	UTF-8	<input type="checkbox"/>	Trading Partner Delivery
Delivery - Message Encoding:	base64	<input type="checkbox"/>	Trading Partner Delivery
Delivery - Content Type:		<input type="checkbox"/>	Host Delivery Protocol

FIGURE 5-28 TP Parameters at the Transaction Profile Level

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 example:

- 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 144.

Basic Operations for Trading Partners

This section covers creating, finding, selecting, and adding a B2B Host to a Trading Partner.

▼ To create a new trading partner

- 1 With the Trading Partner Configuration tab active, in the explorer pane, click Create.

See [Figure 5–29](#).

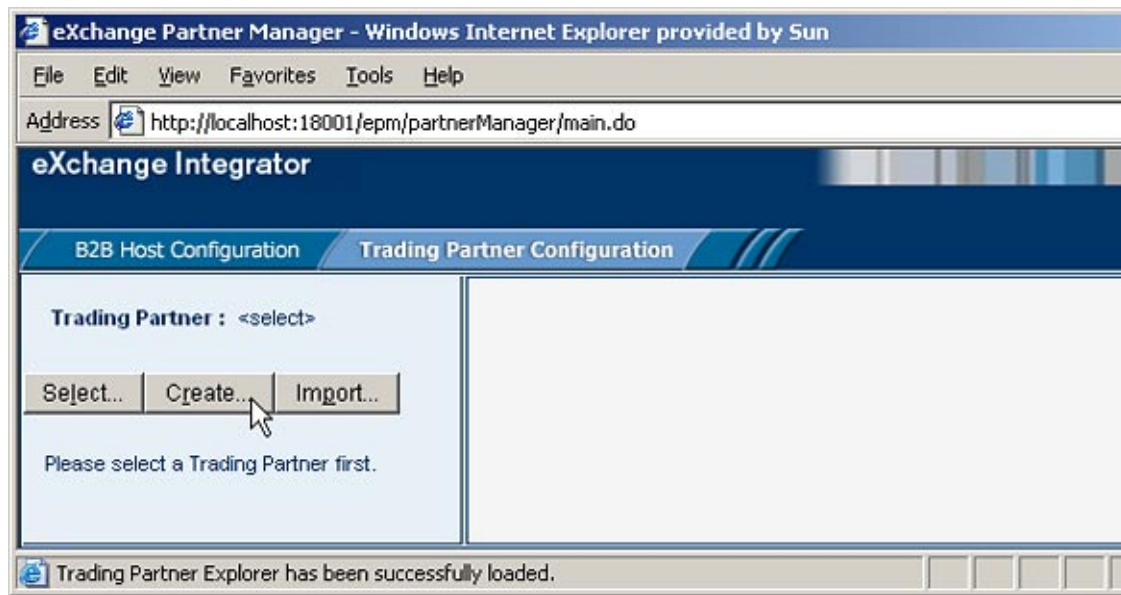
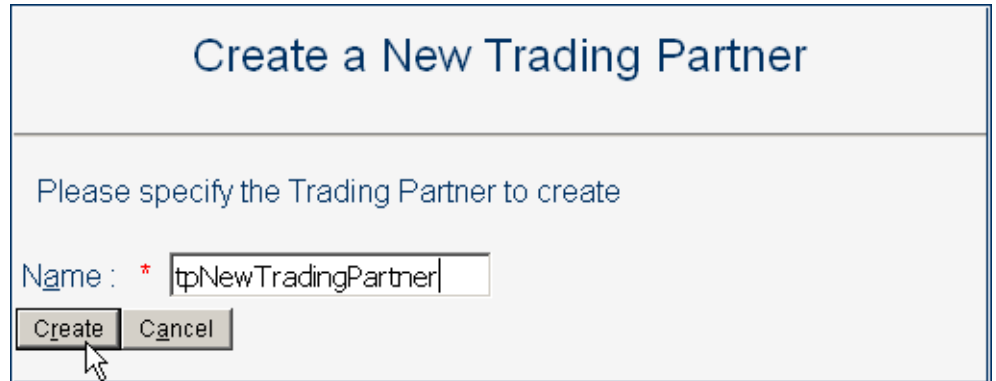


FIGURE 5–29 Creating a New Trading Partner

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

See [Figure 5–30](#).



Create a New Trading Partner

Please specify the Trading Partner to create

Name : *

Create Cancel

FIGURE 5–30 Naming a New Trading Partner

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

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**.
- 2 In the canvas, optionally after entering the first few characters of a possible name, click **Search**. See [Figure 5–31](#).

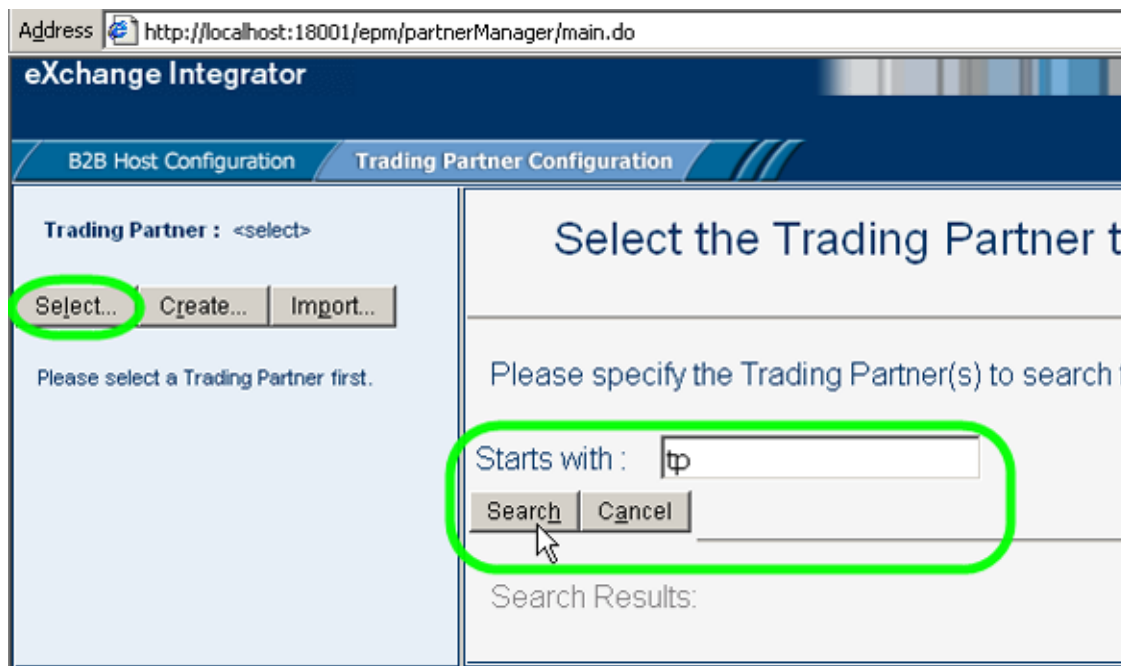


FIGURE 5-31 Searching for a Trading Partner

- 3 From the Search Results list, select the trading partner you want.

See [Figure 5-32](#).

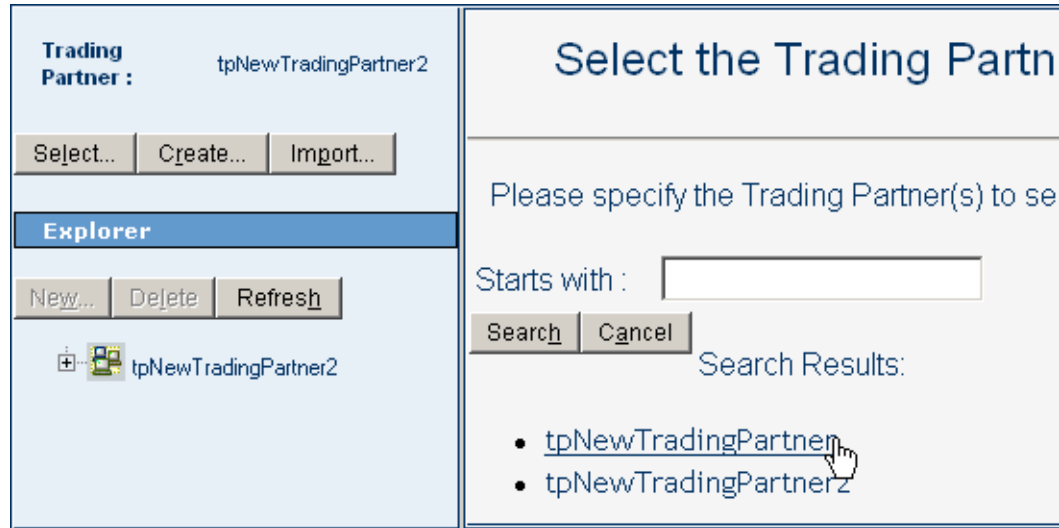


FIGURE 5–32 Selecting a Trading Partner

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.
- 2 Then, in the canvas (under the Settings tab), select a B2B Host from the pull-down list.
See [Figure 5–33](#).

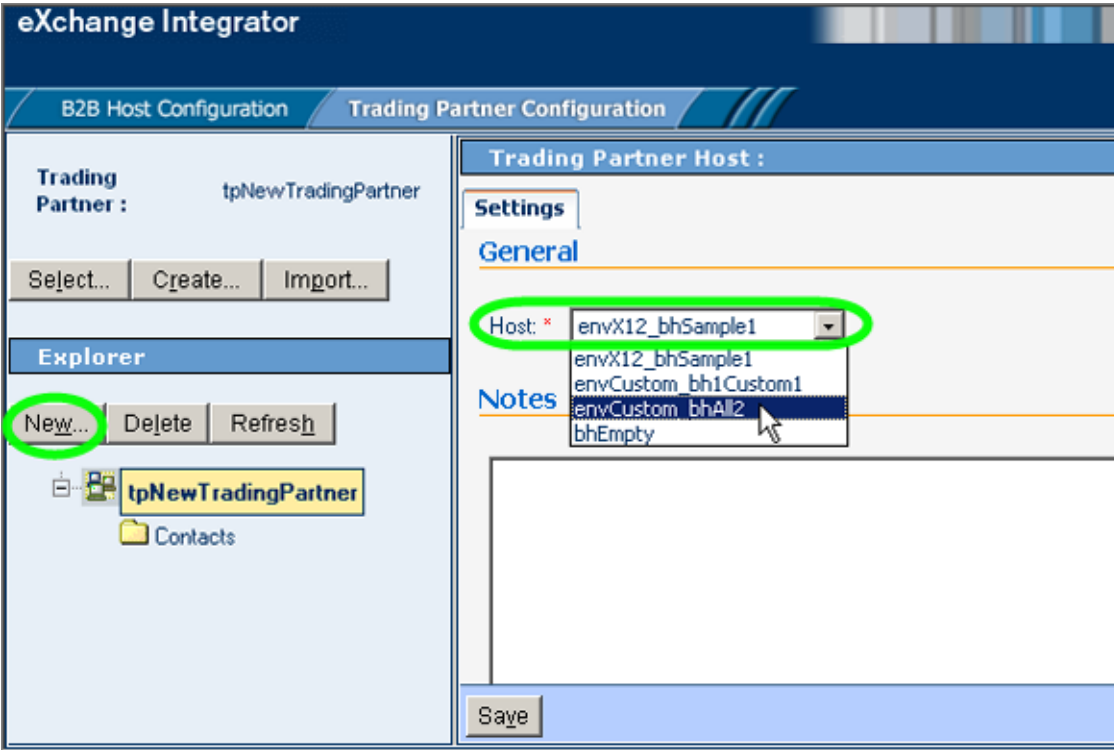


FIGURE 5-33 Configuring a Trading Partner With a New B2B Host

3 Click Save.

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.

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.

▼ **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.

The new TP transaction profile appears in the TP tree in the left pane.

Note – If the TP tree does not display the new TP transaction profile, click Refresh.

- 3 Optional. You can configure one or more TP transaction profiles by following the steps below, repeating this step as needed for additional TP transaction profiles.
- 4 In the TP explorer tree, open the TP transaction profile and select the inbound TP transaction profile.
- 5 Use the canvas to override TP transaction profile parameter settings as needed, and click Save.
- 6 Within the same TP transaction profile, select the outbound TP transaction profile.
- 7 Use the canvas to override TP transaction profile parameter settings as needed, and click Save.
- 8 Repeat the procedure as needed for additional TP transaction profiles.
- 9 Optional. 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 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 Optional. To create a new TP business protocol:
 - a. Click the host's Business Protocols folder.
 - b. 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:
- 3 In the TP explorer tree, select the TP business protocol.
- 4 Use the canvas to override TP business protocol parameter settings as needed, and click Save.

- 5 With the TP business protocol selected, click New.
- 6 To create a new TP business action group, select a host business action group and click Save.
- 7 In the TP explorer tree, select the TP business action group.
- 8 Use the canvas to override TP action group parameter settings as needed, and click Save
- 9 With the TP business action group selected, click New.
- 10 To create a new TP business action, select a host business action and click Save.
- 11 In the TP explorer tree, select the TP business action.
- 12 Use the canvas to override TP business action parameter settings as needed, and click Save.
- 13 Optional. To create a new TP delivery protocol:
 - a. Click the host's Delivery Protocols folder,
 - b. Make a selection from the Host Business Protocol pull-down list, and click Save.
- 14 In the TP explorer tree, select the TP delivery protocol.
- 15 Use the canvas to override TP delivery protocol parameter settings as needed, and click Save
- 16 With the TP delivery protocol selected, click New.
- 17 To create a new TP delivery action group, select a host delivery action group and click Save.
- 18 In the TP explorer tree, select the TP delivery action group.
- 19 Use the canvas to override TP action group parameter settings as needed, and click Save.
- 20 With the TP delivery action group selected, click New.
- 21 To create a new TP delivery action, select a host delivery action and click Save.
- 22 In the TP explorer tree, select the TP delivery action.
- 23 Use the canvas to override TP delivery action parameter settings as needed, and click Save

- 24 Optional. To create a new TP transport:
 - a. Click the host's Transports folder.
 - b. Make a selection from the Delivery Channel pull-down list and click Save
 - c. If the TP tree does not display the new TP Transport, click Refresh.
- 25 Optional. To configure a TP transport:
 - a. Select the transport in the TP explorer tree.
 - b. Use the canvas to override TP transport parameter settings as needed, and click Save.
- 26 Optional. To create a new TP contact based on a host contact:
 - a. Click the host's Contacts folder
 - b. Make a selection from the Trading Partner Host Contact pull-down list, and click Save.
 - c. Select the contact in the TP explorer tree.
 - d. 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 5–34](#).

Trading Partner : tpNew3

Select... Create... Import...

Explorer

New... Delete Refresh

tpNew3

envCustom_bhAll2

bhEmpty

Contacts

Trading Partner Contact :

Settings

General

Name: *

Email: *

Primary Contact: * Yes

Properties

Mailing Address

Address 1: *

Address 2:

City: *

State: *

Country: *

Zip Code: *

Phone/Fax Numbers

Notes

Save

FIGURE 5-34 Creating a New TP-Specific Contact

- 3 Supply information for all required fields and, as needed, any optional fields, and click Save.
- The name of the new contact appears in the TP tree, under the TP's Contacts folder.

Message Tracker

Included with eXchange Integrator 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 Tracker. 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 Tracker.

Accessing Message Tracker

▼ To access Message Tracker

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 Tracker to be useful, there must be one or more messages that have already been picked up by the current domain's Integration Server.

1 Start a browser session.

2 Point your browser at the following URL:

`http://logicalhost:port+1/objname`

Where:

- *logicalhost* is the hostname or IP address of the machine where your Project is deployed; the machine name of the Logical Host.
- *port+1* is the port number, 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 **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* is the name of the Message Tracker instance as it appears on the Connectivity Map. By default, the name that appears in cmTracker is tracker.

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

`http://localhost:18001/tracker/msgTrack/`

Using Message Tracker

When you first run Message Tracker, the **Message Tracker** 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 5–35](#).

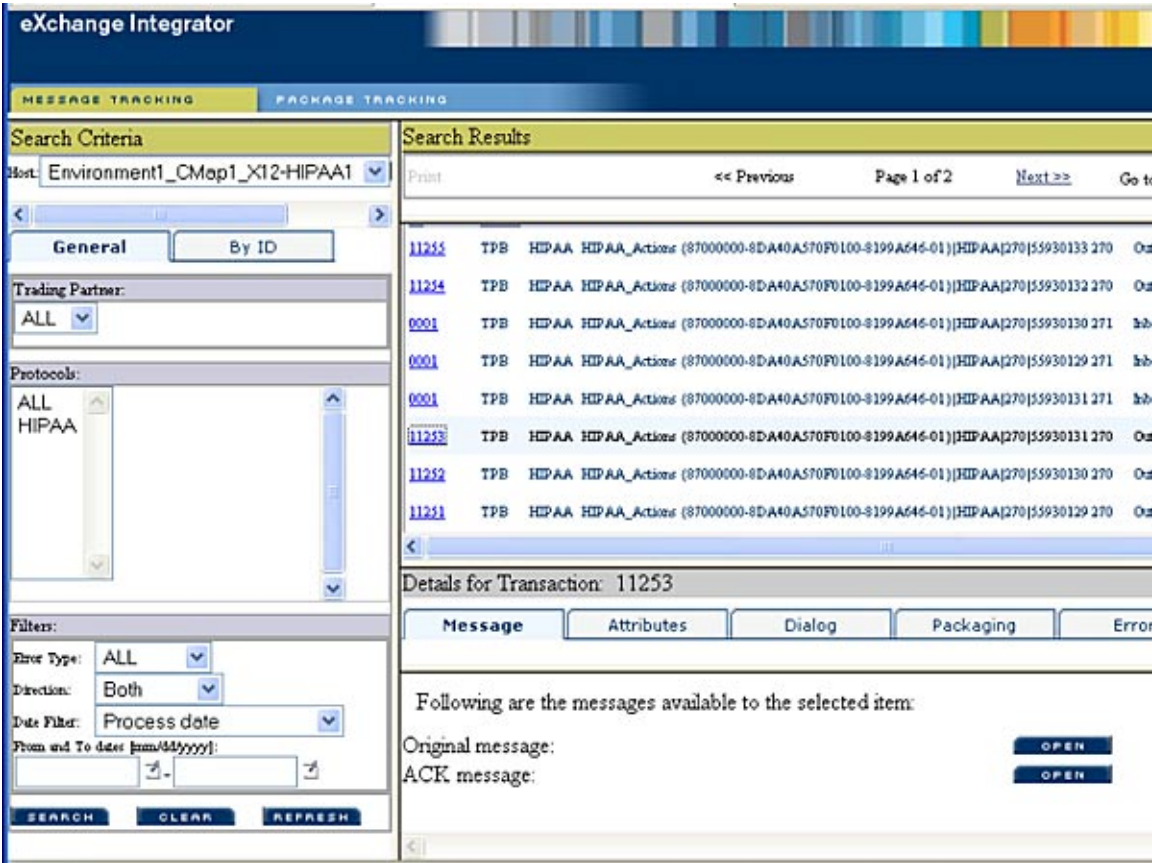


FIGURE 5–35 Message Tracker Window

▼ 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**.
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**.
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

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**.
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**

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.**

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

Designing Business Processes

You can use eXchange Integrator 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 Integrator Developer's Guide*. For detailed information on creating a project with a business process, see the *eInsight Business Process Manager User's Guide*.

What's in This Chapter

- “Using the eXchange Protocol Designer GUI” on page 167
- “Modeling Elements” on page 168
- “Using Business Processes in a Connectivity Map” on page 174

Using the eXchange Protocol Designer GUI

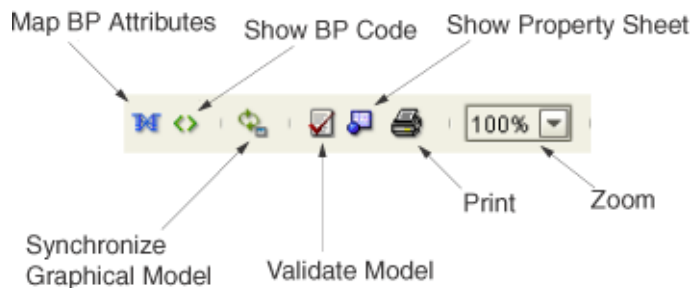


FIGURE 6-1 Toolbar Options

- *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.

Modeling Elements

The eXchange Protocol Designer is where the user creates the Business Process flow. It provides a palette of modeling elements for designing your Business Process. 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 Business Process you create consists of basic elements as described in the following sections:

- [“Activity Elements” on page 168](#)
- [“Branching Activities” on page 171](#)
- [“Intermediate Events” on page 172](#)
- [“Scope” on page 173](#)
- [“While” on page 173](#)

Activity Elements

Adding an Activity

You can include several different kinds of activities and subprocesses in a Business Process.

▼ To add an activity

- 1 **Do one of the following:**
 - Drag a modeling element from the toolbar and drop it on the canvas.
 - Drag a web service operation from the Project Explorer and drop it on the canvas.

- 2 **Click the default activity name and type a new name.**

The activity appears on the modeling canvas.

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.

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 6–1](#).

Linking Modeling Elements

eXchange Integrator 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

- 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 6–2](#).



FIGURE 6–2 Selected Activity

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

Activity Elements

TABLE 6-1 Activity Elements










Button	Command	Function
 Start	Start Node	<p>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.</p> <p>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).</p>
 	Link Link with Business Rule	<p>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.</p> <p>eXchange Integrator 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.</p>
 End	End Node	<p>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.</p>
	Receive Activity	<p>The Receive activity indicates the invocation of a BP or a wait state pending the arrival of an inbound message.</p> <p>The Receive activity represents the actual method by which a BP is initiated. For example:</p> <ul style="list-style-type: none"> ■ 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	<p>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.</p>

TABLE 6-1 Activity Elements (Continued)

Button	Command	Function
	Reply Activity	<p>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.</p> <p>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.</p>
	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 as well as eXchange Integrator. It is used when assigning, escalating, or otherwise using human intervention to complete eInsight business process tasks.

Branching Activities

TABLE 6-2 Branching Activities

	Decision	<p>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 Business Process. It also contains the expression and connection names.</p> <p>Decisions allow you to define expressions that are evaluated to determine the proper Business Process flow. Expressions are built using the mapping interface and Business Process attributes.</p>
	Event Based Decision	Multiple possible messages can be juxtaposed against a timeout condition to allow the type of message received to determine the appropriate Business Process path.
	Flow	Allows you to specify one or more activities to be performed concurrently.

▼ **To add an intermediate event**


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

- 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.

Intermediate Events

TABLE 6-3 Intermediate Events

	Compensation Handler	Used when something in a Business Process fails and requires a rollback or upstream activities (like money has to be returned to the customer). On an automatic basis in the Business Process , upstream steps in the Business Process 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 Business Process . The components within that scope throws the exceptions when things go wrong and the exception handler automatically initiates the appropriate process to handle the problem.
	Catch All Exceptions	This exception handler is configured to handle all exceptions that occur in a scope.
	Message Event	This is similar to a Receive Activity, but it occurs only in the middle of a Business Process . Each of these elements can be a different message.
	Timer Event	A timeout condition is set upon Activities, sets of Activities, or a Business Process as a whole, to ensure that processes complete within given amount of time. Timeout conditions also allow you to design the Business Process 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 Business Process .

While

While

This allows you to create a looping process within a Business Process (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 Business Process continues.

Validating a Business Process

After creating a Business Process , 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 Business Process 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” on page 173](#) for information on repairing errors. Repairing the error may entail such items as adding logic to Decisions or adding attributes to activities.

Saving an Unfinished Business Process

Even if a Business Process 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**

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 Business Process . eXchange Integrator also uses the information in the connectivity map to establish and maintain connections to systems for the connect step in a Business Process .

▼ **To include a Business Process as a service on a connectivity map**

- 1 In the Connectivity Map Editor, drag a Business Process onto the canvas.
- 2 Add and connect other components and external systems as needed.

See [Figure 6–3](#).

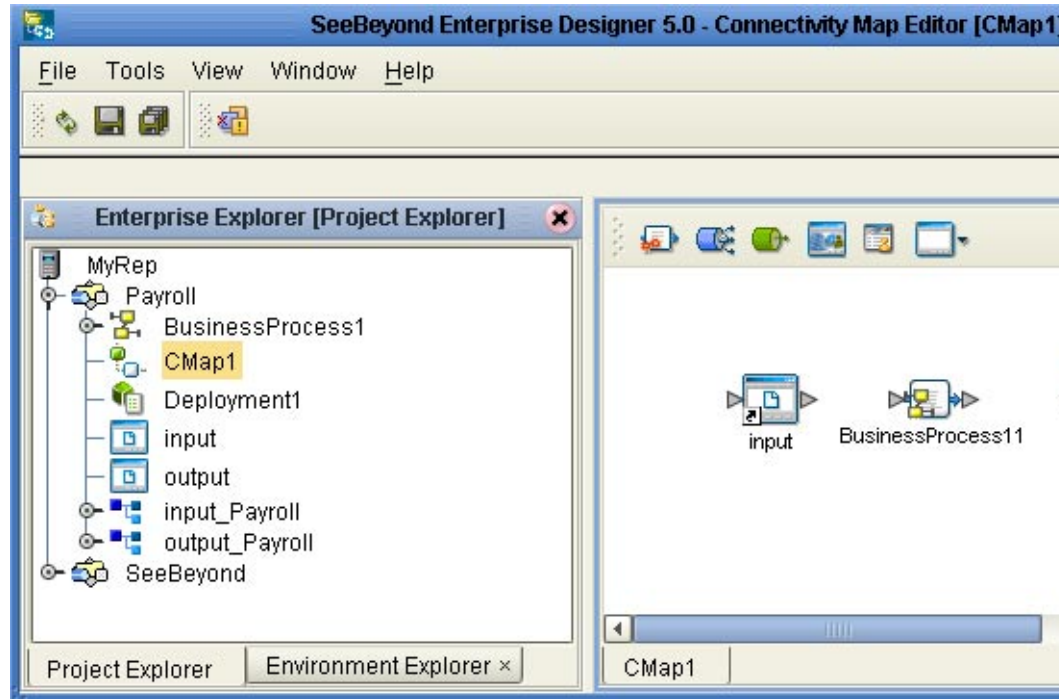


FIGURE 6-3 Connectivity Map with Business Process

▼ To connect the Business Process activities to the externals

- 1 In the map, double-click the Business Process 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 6-4](#).

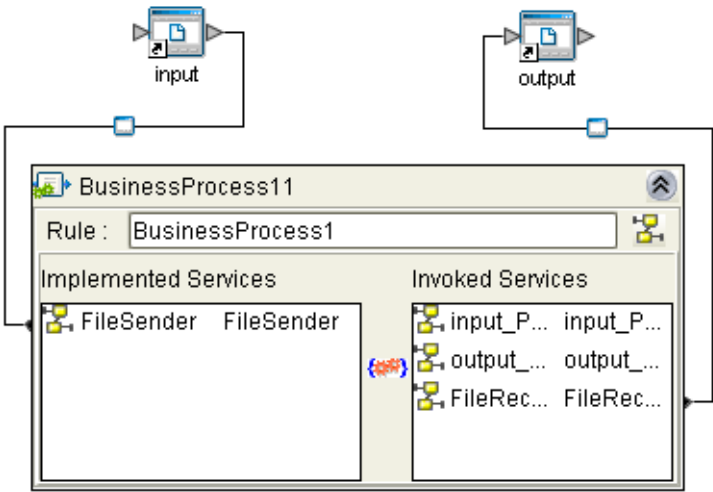


FIGURE 6-4 Connectivity Map: B2B Protocol Binding

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

- “Overview” on page 177
- “Scope” on page 179
- “Compensation” on page 180
- “Validating the B2B Protocol” on page 180

Overview

Exception handling is the identification of failed components or systems. In eXchange Integrator, exception handling allows one or more components to throw an exception that is caught by eXchange Integrator within a scope. Using the scope element, you can configure eXchange Integrator 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.

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. The following represents one example:

▼ Exception Handling 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.

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 contains the run-time name of the thrown fault.

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

- Select the configuration control for the Exception Handler– the properties pane appears to select the Fault name and container.
- Fault Name — The run-time value for the exception that is passed from the component to the engine at run time.

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.

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.

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.

eXchange Troubleshooting Tips

This chapter provides troubleshooting and tuning information for eXchange Integrator.

What’s in This Chapter

- “Identifying and Resolving Problems” on page 181
- “Performance Considerations” on page 185

Identifying and Resolving Problems

The Sun Java System B2B Release Notes PDF contains a list of known issues for eXchange Integrator and other B2B products. This document is available at <http://docs.sun.com/app/docs>.

For help resolving some commonly encountered difficulties, see Table 8–1.

TABLE 8–1 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: <i>product</i> ”	One or more prerequisite .sar files have not been installed.	Ensure you are licensed to install the product. Install its prerequisite .sar files 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.

TABLE 8-1 Troubleshooting: Symptoms, Causes, and Remedies (Continued)

Symptom	Possible Causes	Suggested Remedies
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 <i>jc512</i> locations: <ul style="list-style-type: none"> ■ edesigner\bin\log4j.properties ■ ESRs\log4j.properties ■ monitor\config\log4j.properties ■ Repository\server\conf\log4j.properties
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.	<ul style="list-style-type: none"> ■ 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. 	<ul style="list-style-type: none"> ■ 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: <ul style="list-style-type: none"> ■ 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 <i>logicalhost</i> , except bootstrap and jre; and then re-bootstrap to regenerate the keystores.

TABLE 8-1 Troubleshooting: Symptoms, Causes, and Remedies (Continued)

Symptom	Possible Causes	Suggested Remedies
<p>Changes made to the design, environment, or trading partner profile are not picked up at run time. For example:</p> <ul style="list-style-type: none"> ■ 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. 	<p>Stale information is being retained by the Repository or Logical Host or eXchange database.</p>	<ul style="list-style-type: none"> ■ 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 rebuild 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 rebuild the trading partner.
<p>Message Tracker reports "No messages found".</p>	<ul style="list-style-type: none"> ■ The statement is accurate, and no messages have been processed. ■ The wrong host is specified and/or the filters are too stringent. 	<ul style="list-style-type: none"> ■ 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.
<p>The Business Process starts, and ChannelManager retrieves the delivery channel profile, but nothing happens.</p>	<ul style="list-style-type: none"> ■ There is no input being staged. ■ There is no binding for the internal delivery channel to the location where input exists. 	<ul style="list-style-type: none"> ■ 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, resave.
<p>Errors occur upon trying to bootstrap the Logical Host; log file text reads as follows: "DCPAccessException: No Host with Network Id ... at ...DCPDBViewV2.lookup"</p>	<p>You are trying to run a project that performs a TP lookup, but no trading partner profile has been saved in ePM.</p>	<p>Create and save one or more trading partner profiles. See "Creating and Configuring Transaction Profiles" on page 140.</p>

TABLE 8-1 Troubleshooting: Symptoms, Causes, and Remedies (Continued)

Symptom	Possible Causes	Suggested Remedies
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 Integrator Database” on page 41 .
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 185 . For example, when dealing with 100-200 concurrent messages, the following settings might be adjusted. For Oracle: <ul style="list-style-type: none"> Change Oracle number of processes to 500. For the Logical Host: <ul style="list-style-type: none"> Change heap size from 512 to 1024 (or higher, if the machine has sufficient memory). For LH >IS>eInsight Engine: <ul style="list-style-type: none"> 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 See the <i>eInsight Business Process Manager User's Guide</i> .
At runtime, batched messages are not handled correctly, or messages are not correlated correctly. The server.log file contains messages such as: “SQL Error: 1461, SQLState: 72000, and ORA-01461: Can bind a LONG value only for insert into a LONG column”.	In the environment referenced by the B2B Host, the value supplied for the User (and/or Password) in the Oracle external is different from the value for UserName (and/or Password) in the B2BService Configurator.	In the environment referenced by the B2B Host, edit the B2BService Configurator properties so that the Oracle UserName and Password values match the User and Password values supplied for the Oracle external. Next, rebuild the B2B Host. Finally, rebuild and redeploy all projects that reference this B2B Host.

TABLE 8-1 Troubleshooting: Symptoms, Causes, and Remedies (Continued)

Symptom	Possible Causes	Suggested Remedies
<p>If an inbound message's GS01 (<i>FUNC_ID_CODE</i>) does not match the corresponding information for the Trading Partner as specified in ePM, the B2B Suite will create a dead letter. The deadletter and the server.log will contain the following message:</p> <p>Could not retrieve ExStdEvent</p>	<p>The value for the inbound GS01 does not match the value specified in ePM for the Trading Partner.</p> <p>Example log message:</p> <p>PM GS01 set to HS and inbound sent with GS01 = HA</p>	<p>Check to see if any of the following values do not match the expected values set in ePM, where <i>var=value</i>:</p> <pre>ISA_SENDER_QUAL = 01 ISA_SENDER_ID = AtlantaIID ISA_RECEIVER_QUAL = 01 ISA_RECEIVER_ID = BerlinIID ISA_VERSION = 00401 ST_TRANS_ID = 270 FUNC_ID_CODE = HA GS_SENDER_ID = AtlantaAcid GS_RECEIVER_ID = BerlinAcid GS_VERSION = 004010 </pre> <p>Correct the mismatched information.</p>
<p>Incorrect configuration of database users can result in runtime database errors.</p>	<p>This problem can occur if different users are assigned via Enterprise Designer to the Oracle eWay and the <i>B2BServiceConfiguration</i>. For example, if the Oracle eWay's user is set to <i>ex_adminA</i> and the <i>B2BServiceConfiguration</i> Oracle user is set to <i>ex_admin</i>, a database error will occur. The B2B Suite does not support using two different database users for the same environment setup.</p>	<p>To correct the error, set the <i>B2BServiceConfiguration</i> Oracle user to <i>ex_adminA</i> and rebuild and redeploy eXchange.</p>

Performance Considerations

For B2B Suite 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 186
- “Increasing eInsight Engine Performance” on page 187
- “Increasing the JMS IQ Manager Segment Size Property” on page 188
- “Increasing the Oracle Number of Processes” on page 189

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.

See [Figure 8–1](#).

- 2 In the **Heap Size** box, enter the new heap size.

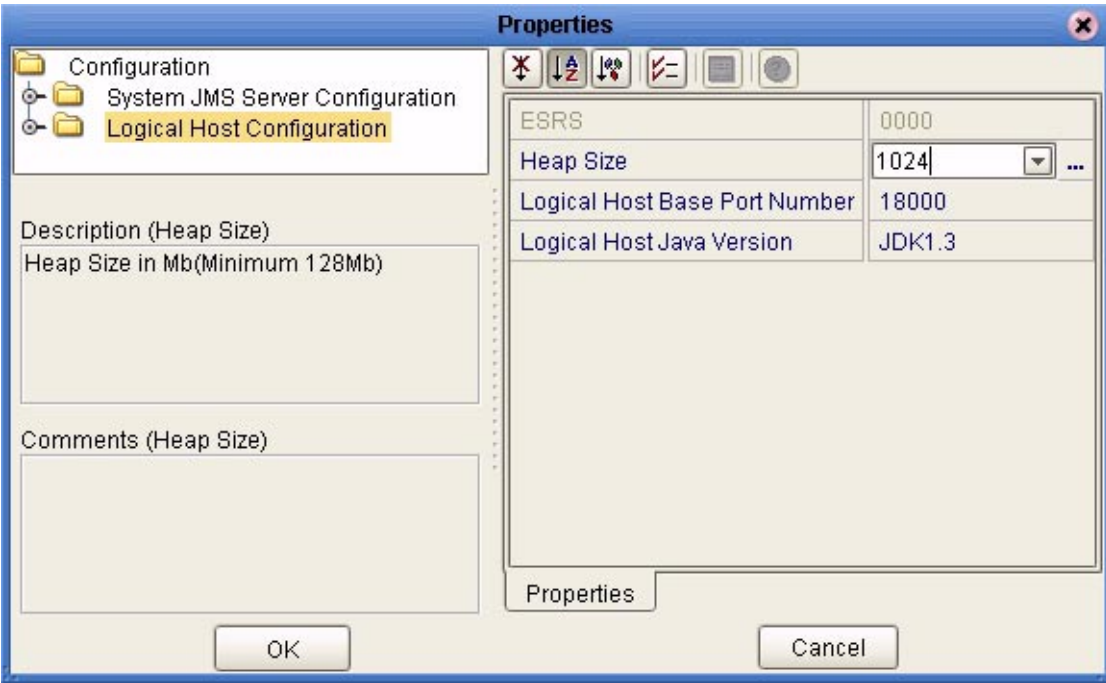


FIGURE 8–1 Increasing Logical Host Heap Size

- 3 Click **OK**.

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, and more information about these properties, see 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 In the Properties dialog, expand the Configuration⇒eInsight Engine Configuration folders.
- 3 Increase the settings for the Max Concurrent Instances and Max Runtime Thread Pool Size properties.

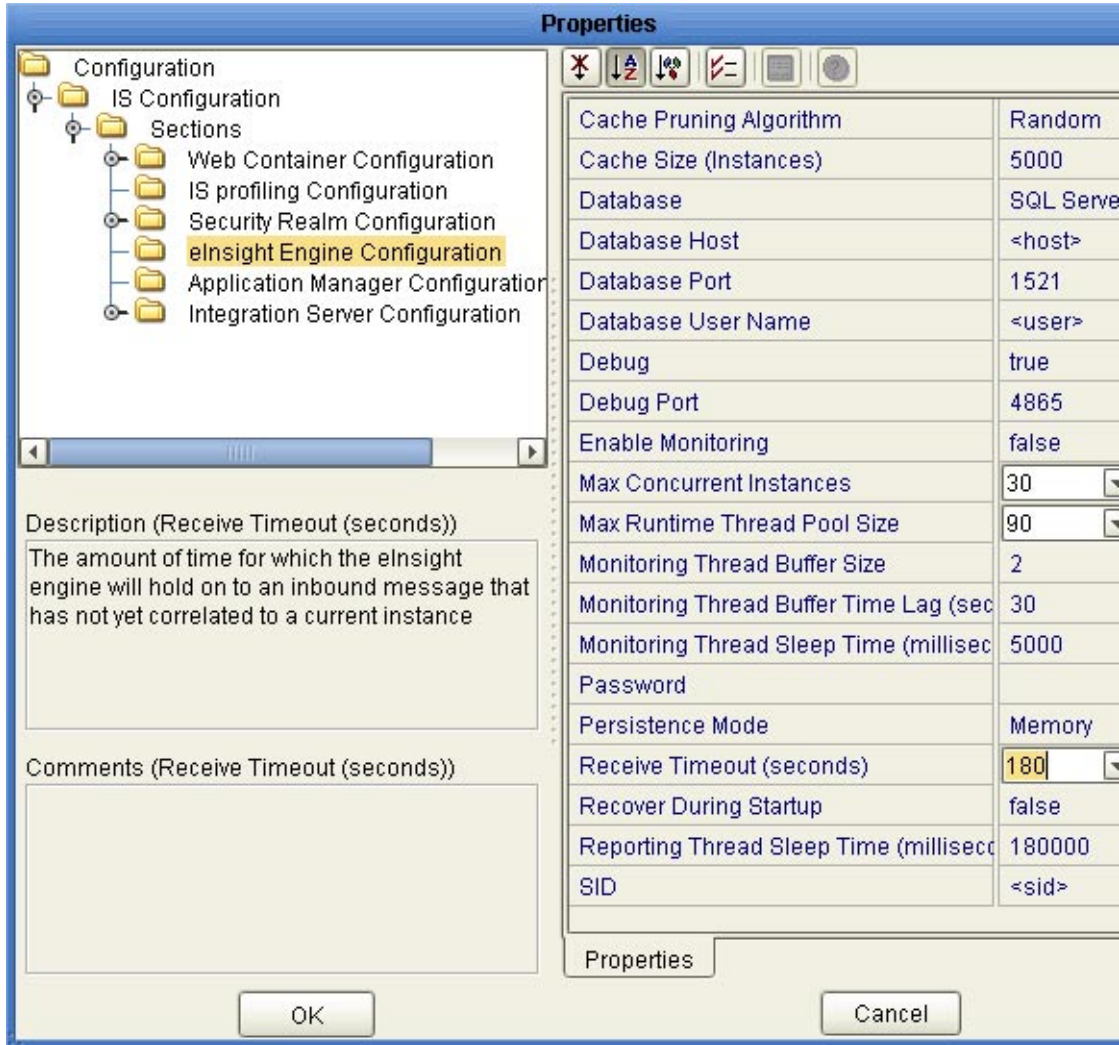


FIGURE 8-2 Increasing eInsight Engine Performance

- 4 Click OK.

Increasing the JMS IQ Manager Segment Size Property

If a Project that uses the SeeBeyond JMS IQ Manager needs 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*.

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

AD, *AD, xAD	In eXchange Integrator an Attributes Definition defines the metadata attributes of parameters used in a business protocol, delivery protocol, or transport. Examples of xADs include: BPAD=BAD+EAD; DPAD=MAD+PAD; and TAD.
AS2	Applicability Statement 2 (AS2) is an Internet Draft security standard defined by the IETF (Internet Engineering Task Force), designed to allow business transactions to move securely over the Internet.
B2B	Business-to-business (B2B) interactions are those that occur between business partners in the context of e-commerce.
BAD	In eXchange Integrator, 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 Integrator, 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 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 Integrator, 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 Integrator, 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 <code>http:// hostname : port /.../....</code> . Specifications for HTTP include RFCs 2068, 2616, 2617, 2660, and 3310.
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 Integrator, 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 the B2B Suite, 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 Integrator, 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.
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.
SME	In the B2B Suite, Secure Messaging Extensions (SME) uses advanced cryptographic techniques to ensure security, verifiability, and non-repudiation of messages exchanged electronically.
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 Integrator, 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 Integrator, a Trading Partner (TP) has one or more Trading Partner Profiles (TPPs) that contain information identifying the values of messaging, enveloping, and/or transport parameters to be used for sending and receiving B2B information.
URL	A Uniform Resource Locator (URL) is a string that identifies information, such as a particular piece of information shared by a particular host.

XML

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

Index

Numbers and Symbols

"bootstrap already running", 182
"Could not install" (error message), 181
"Error while decrypting", 182
"installation failed" (error message), 181
"InstallJavaCollab - Error is", 181
"instance of bootstrap already running, 182
"Max Instances Exceeded", 184
"maximum open cursors exceeded", 181, 184
"No Host with Network Id ... Found", 183
"RSA_not supported", 182

A

associate (Channel Manager operation), 59
associateActions (Channel Manager operation), 60
attribute definitions
 as components of a DCP, 72
 business.
 See business attribute definitions.
 enveloping.
 See enveloping attribute definitions.messaging
 attribute definitions.
 messaging.
 See messaging attribute definitions.
 transport.
 See transport attribute definitions.

B

B2B host
 activating, 88-99, 99-103, 103-106
B2B Host Designer, (illustrated), 74-75
B2B Hosts, renaming, 74
B2B protocols
 ErrorHandlerSelector, 67
 JMSErrHandler, 68
B2B Protocols⇒ ebXML folder, (illustrated), 65
batched messages not handled correctly, 184
BatchFTP transport attributes definition, 52, 54
bootstrap.lock file, 182
BPADs, 81
 for HIPAA, 75
 for X12, 75
BPELInterpreter exception, 184
business attribute definitions, 83
business protocol attribute definitions, 81

C

Catch All Exceptions, 179
Catch Named Exception, 179
Channel Manager folder, (illustrated), 58
ChannelManagerClient operations
 associate, 59
 associateActions, 60
 lookupTPFromPartner, 60
 track, 60
 trackDialogue, 61
 trackDialogueAction, 62

- ClassNotFoundException, 181
- correcting problems, 181
- createdb.sql
 - running, 48
- createtablespace.sql, running, 47
- createuser.sql, running, 47
- custom messaging attribute definitions, 75
 - creating, 85
- custom transport attribute definitions, creating, 77

D

- database for eInsight engine
 - supported levels, 28, 40
- database for eXchange
 - creating, 40, 46-48
 - supported levels, 28
 - updating for 5.0.x, 46-48
- DCPAccessException, 183
- DCPDBViewV2.lookup, 183
- designing B2B protocols, 167-175

E

- ebXML protocols, location in project tree, 65
- editors
 - B2B Host Designer, 74-75
 - ePM, 112-115
 - eXchange Partner Manager (ePM), 112-115
 - eXchange Protocol Designer, 167-168
- eInsight engine
 - supported databases, 28, 40
- enveloping attribute definitions, 83
- ePM
 - (illustrated), 112, 117
 - accessing, 115-117
- error handling
 - default, 67
 - nondefault, 68
- Error Management folder
 - ErrorHandlerSelector, 67
 - JMSEErrorHandler, 68

- error messages
 - "(InstallJavaCollab) Error is", 181
 - "bootstrap already running", 182
 - "Could not install", 181
 - "installation failed", 181
 - "Max Instances Exceeded and ...", 184
 - "maximum open cursors exceeded", 181, 184
 - "No Host with Network Id ...
Found"DCPAccessException
No Host with Network Id ... at
...DCPDBViewV2.lookup, 183
 - (terseness/verbosity), 182
 - ClassNotFoundException, 181
 - encryption/decryption, 182
 - NBSystemResourceLoader, 181
- ErrorHandlerSelector
 - (location in project tree), 67
 - mapping to ErrorHandler, 67
- EX_DEADLETTER topic, 65, 66
- EX_ERROR topic, 65, 66
- EX_PROCESSEDERRORS topic, 65, 66
- exceptions, in B2B protocols
 - Catch All Exceptions, 179
 - Catch Named Exception, 179
 - scope, 179
- eXchange database
 - init.ora file, 41, 184
 - open_cursors parameter, 41, 184
 - system requirements, 28
 - updating for 5.0.x, 46-48
- eXchange folder
 - B2B Protocols⇒ebXML folder, 65
 - Channel Manager, 58
 - DBScripts, 43, 44
 - Message Tracker folder, 64
 - Transport Attribute Definitions, 50
- eXchange Partner Manager., *See ePM.*
- eXchange Protocol Designer, 167-168
- eXchange support
 - for Oracle databases, 28
 - for platforms, 27

F

FILE transport attributes definition, 55
 FTP, transport attributes definition, 56

G

GUIs

Attributes dialog, 78
 B2B Host Designer, 74-75
 ePM, 112-115
 eXchange Partner Manager (ePM), 112-115
 eXchange Protocol Designer, 167-168

H

heap size, adjusting heap memory size, 186
 HIPAA BPAD, 75
 HTTP transport attributes definition, 56
 HTTPS
 configuring environment to use, 92
 JSSE Provider Class for, 93
 X509 Algorithm Name for, 93
 HTTPS transport attributes definition, 57

I

identifying problems, 181
 init.ora file, 41, 184
 integration server, keystore files, 93

J

jar files, for encryption/decryption, 182
 java.lang.SecurityException, 182
 java.sql.SQLException, 181, 184
 JMS topics for error handling
 EX_DEADLETTER, 65, 66
 EX_ERROR, 65, 66
 EX_PROCESSEDERRORS, 65, 66
 JMS transport attributes definition, 57
 JMSErrorHandler, purpose, 68

JSSE Provider Class, for HTTPS, 93

K

keystore files, 93

L

local_policy.jar, 182
 log4j.properties file, 182
 lookupTPFromPartner (Channel Manager), 60

M

Message Tracker, using, 163
 Message Tracker folder, (location in project tree), 64
 messages no correlated correctly, 184
 messaging attribute definitions
 custom, 75, 85

N

NBSystemResourceLoader, 181

O

open_cursors parameter, 41, 184
 operating systems supported by eXchange, 27
 Oracle database for eXchange
 supported levels, 28
 updating for 5.0.x, 46-48
 Oracle database parameters, 41, 184

P

performance tips, 185
 PKCS7 format, 182
 platforms supported by eXchange, 27
 policy.jar files, 182

problems, identifying and correcting, 181

R

renaming, B2B Hosts, 74

runtime.BPELInterpreter exception, 184

S

sbyn-exchange-error.wsdl, 68

sbyn.keystore file, 93

scope, in B2B protocols, 179

SecurityException, 182

shutdown script, 182

SMTP transport attributes definition, 57

SQLException, 181, 184

T

TADs., *See* **transport attribute definitions**

tips

- for performance, 185

- for troubleshooting problems, 181

track (Channel Manager), 60

trackDialogue (Channel Manager), 61

trackDialogueAction (Channel Manager), 62

trading partner lookup operations,

- lookupTPFromPartner, 60

trading partner profiles, fetching from database, 60

transaction profile, defined and explained, 72

transport attribute definitions

- BatchFTP, 52, 54

- custom, 77

- FILE, 55

- FTP, 56

- HTTP, 56

- HTTPS, 57

- JMS, 57

- metadata for all protocols, 52

- SMTP, 57

Transport Attribute Definitions folder, (illustrated), 50

troubleshooting, 181

U

US_export_policy.jar, 182

using Message Tracker, 163

W

WSDL files, sbyn-exchange-error.wsdl, 68

X

X12 BPAD, 75

X509 Algorithm Name, for HTTPS, 93