SeeBeyond ICAN Suite

eXchange Integrator User's Guide

Release 5.0

SEEBEYOND®

SeeBeyond Proprietary and Confidential

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

SeeBeyond, e*Gate, and e*Way are the registered trademarks of SeeBeyond Technology Corporation in the United States and select foreign countries; the SeeBeyond logo, e*Insight, and e*Xchange are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2003 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

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

Version 20031015103123.

Contents

Chapter 3

Chapter 4

Chapter 5

List of Figures

List of Tables

Chapter 1

Introduction

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

In this chapter

1.1 Overview

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

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

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

1.2 Contents of This Guide

This guide is arranged as follows:

- **Chapter 1, "Introduction"** provides an overview of this document's purpose, contents, writing conventions, and supported documents.
- **Chapter 2, "Overview"** discusses general features and architecture of eXchange.
- **Chapter 3, "Installing eXchange"** provides step-by-step instructions for installing the eXchange product and setting it up for use.
- **Chapter 4, "Using eXchange in Enterprise Designer"** provides step-by-step procedures for working with eXchange at design time and deploying projects.
- **Chapter 5, "Designing B2B Protocol Pipelines"** provides step-by-step procedures for designing and deploying protocol pipelines using eXchange Protocol Designer.
- **Chapter 6, "Exception Handling"** explains the use of protocol pipelines for handling exceptions.
- **Chapter 7, "Using eXchange in Enterprise Manager"** provides step-by-step procedures for working with eXchange's Web-based GUIs—Partner Manager (ePM) and Message Tracking—as well as using Monitor for B2B protocol pipelines.
- **Appendix A, "Method Palette"** lists and describes the tools available to you in eXchange Protocol Designer.

1.3 Writing Conventions

The following writing conventions are observed throughout this document.

Table 1 Writing Conventions

Text	Convention	Example
Names of buttons, files, icons, parameters, variables, methods, menus, and objects	**Bold** text	▪ Click **OK** to save and close. ▪ From the **File** menu, select **Exit**. ▪ Select the **logicalhost.exe** file. ▪ Enter the **timeout** value. ▪ Use the **getClassName()** method. ▪ Configure the **Inbound** File eWay.
Command-line arguments, code samples	`Fixed font`. Variables are shown in ***bold italic***.	`bootstrap -p` ***password***
Hypertext links	**Blue** text	For more information, see **"Writing Conventions" on page 10**.

Additional Conventions

Windows Systems

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

1.4 Supporting Documents

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

Title	Filename
SeeBeyond ICAN Suite Primer	Primer.pdf
eGate Integrator Release Notes	eGate_Release_Notes.pdf
eGate Tutorial	eGate_Tutorial.pdf
eGate Integrator Installation Guide	eGate_Install_Guide.pdf
eGate User's Guide	eGate_User_Guide.pdf
eGate Integrator JMS Reference Guide	eGate_JMS_Reference.pdf
Oracle eWay Intelligent Adapter User's Guide	Oracle_eWay.pdf
SeeBeyond ICAN Suite Deployment Guide	Deployment_Guide.pdf
SeeBeyond ICAN Suite Upgrade Guide, Chapter 3 (“Upgrading e*Xchange”)	eGate_Upgrade_Guide.pdf
eXchange Integrator Release Notes	eXchange_Release_Notes.pdf
	readme.txt

1.5 Online Documents

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

http://www.adobe.com

1.6 The SeeBeyond Web Site

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

http://www.seebeyond.com

Chapter 2

Overview

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

In this chapter

2.1 SeeBeyond Integrated Composite Application Network

Businesses face tremendous challenges in managing dynamic partner relationships and the many processes that control inventory, absorb shifting transaction volumes, and help to transform raw data into capital. Among other things, organizations and their trading partners must manage disparate component applications and align proprietary software requirements while conforming to common data exchange and security standards.

The SeeBeyond® Integrated Composite Application Network (ICAN) Suite merges traditional Enterprise Application Integration (eAI) and Business-to-Business (B2B) interactions into a unified eBusiness infrastructure that connects, integrates, and optimizes data flow across the extended enterprise for customers, suppliers, and partners alike.

ICAN enables you to:

- Leverage your existing technology and applications.
- Create an application consisting of component applications that are managed by your organization for exchanging messages with your trading partners.
- Rapidly execute business strategies.
- Create and manage virtual organizations across the entire value chain.
- Rapidly implement industry-standard business protocols.
- Quickly and easily establish new business partners, or update existing ones.

- Secure transmissions sent through the public domain.

The ICAN Suite also provides:

- Extensive back-office connectivity.
- Powerful data transformation and mapping.
- Content-based routing.
- Unparalleled scalability based on a fully distributed architecture.

2.1.1. ICAN Suite Components

The ICAN Suite includes eGate™ Integrator components, together with servers and business integration applications.

Projects

SeeBeyond ICAN Suite components are used in Projects. A Project represents a logical solution for a business problem.

Environments

Project components are hosted by a collection of physical resources and their configurations called an environment.

eGate Integrator Components

eGate Integrator ("eGate") is an Enterprise Application Integration (eAI) platform that runs on a distributed and open architecture. Depending on the communications protocols and adapters you choose, eGate can communicate with and link multiple applications and databases across a variety of operating systems.

Enterprise Designer

Enterprise Designer is the primary application for configuring eGate to create Projects. It includes subsidiary applications such as a Connectivity Map Editor and Collaborations that you use to develop Projects.

eGate Enterprise Manager

eGate Enterprise Manager is a client-side web application that communicates with the Integration Server and is used to manage service status (stopped, running, starting, or stopping). Enterprise Manager generates "alert," "event" and "trace" notifications.

Enterprise Manager also includes the JMS Administrator, which enables you to manage messages for both the SeeBeyond Message Server and the Nonstop Server for Java Message Service (NSJMS). With JMS Administrator, you can add, delete, refresh, connect to, or disconnect from message servers. You can also bind objects and show bindings.

Diagnostic tools

Diagnostic tools such as Impact Analyzer and Version Control are applications accessed through Enterprise Designer that can be used to examine individual Project components.

Servers

Servers handle back-end processing and storage of various kinds for ICAN Suite components.

SeeBeyond Integration Server

The SeeBeyond Integration Server is a J2EE-compliant software platform that provides application server features such as transaction services, persistence, load balancing, and external connectivity. Integration Server houses the business logic container used to run Collaborations, which are encoded business rules and the information that enables them to interact with other components.

SeeBeyond Security Server

SeeBeyond Security Server is a standalone server that enables disparate security frameworks to be integrated across an enterprise.

SeeBeyond Message Server for eGate

The message server is a JMS-compliant, guaranteed delivery, store, forwarding, and queueing service.

Business Integration Applications

Business integration applications are eGate add-ons that extend ICAN Suite functionality.

eWay Intelligent Adapters

eWay Intelligent Adapters (“eWays”) connect applications and databases with eGate, communicating with both external applications and message servers. When integrating different systems, the appropriate eWay on each end of the route provides the adaptation necessary for seamless flow of data.

eXchange Integrator

eXchange Integrator (“eXchange”) runs on the Integration Server, where it tracks messages and manages trading partner profiles through support for such business message format and enveloping protocols as AS2, and ebXML.

eTL Integrator

eXtract Transform and Load Integrator (“eTL”) is an integrated Collaboration editor used to perform high-volume, high-speed extractions, transformations, and loads, primarily from databases and flat files in tabular format. eTL Collaborations are deployed through eGate’s Enterprise Designer.

eInsight Business Process Manager

eInsight Business Process Manager ("eInsight") facilitates the automation and administration of business process flow across business activities. Real-time graphical modeling and monitoring enables users to assess the state of a business process instance and identify any bottlenecks in that process.

eInsight ESB

eInsight ESB facilitates the automation and administration of business process flow across Web services (services written in the WSDL format of XML) to the exclusion of all other types of objects. eInsight ESB does not recognize eWays other than the SOAP (Simple Object Access Protocol) eWay.

eVision Studio

eVision Studio ("eVision") is SeeBeyond's Web application development product. eVision applications enable users to interact with eInsight activities through the ePortal Composer framework. eVision applications also enable users to send data to or receive data from eGate.

eView Studio

eView Studio ("eView") is a plug-in for eGate, and an application-building tool that enables you to design, configure, and create a master index for uniquely identifying and cross-referencing the business objects (customers, vendors, members, hardware parts, and so on) stored in your system databases. Using eView, you can define the structure of your business objects and the logic that determines how business object data is updated, standardized, weighted, and matched in the master index database.

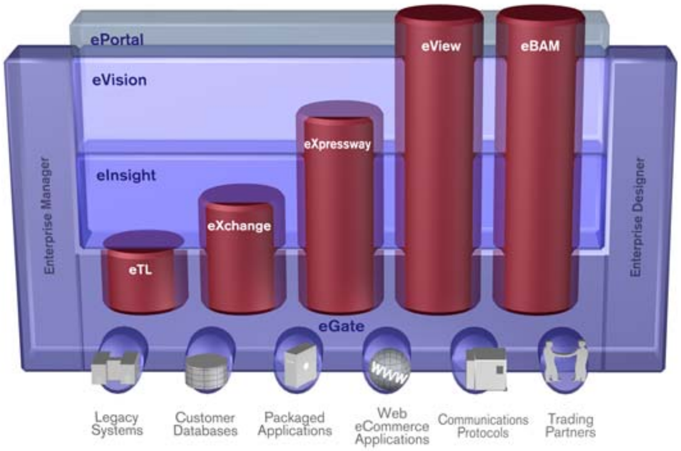
eIndex Global Identifier

eIndex Global Identifier ("eIndex") is a highly customized instance of eView with added plug-in modules to facilitate person matching. eIndex automates identification and matching of member data across disparate source systems to simplify the process of sharing that data. eIndex creates a single view of all member data by providing a common identification process regardless of the system from which data originates.

ePortal Composer

ePortal Composer ("ePortal") is a plug-in for eGate. ePortal enables people to interact with designated business processes over the Intranet or Internet.

Figure 1 SeeBeyond Integrated Composite Application Network (ICAN) Suite



For more information about the ICAN Suite, see the documentation associated with each products.

2.2 About eXchange Integrator

eXchange Integrator is an application that runs on the SeeBeyond Integration Server in distributed computing environments. eXchange manages interactions with trading partners by facilitating the reception, validation, transmission, and tracking of messages in supported formats.

Design-time, run-time, and administrative tools

eXchange provides three major sets of tools, for design-time, run-time, and monitoring.

- Integrated with the Enterprise Designer framework, eXchange's design-time tools, libraries, and prebuilt B2B protocol pipelines supplement and extend the eGate tool set with the following editors and components:
    - B2B Host Designer, along with Channel Manager, allow you to create and manage internal and external delivery channels tailored to the exact needs of each trading partner.
    - Protocol attribute definitions allow you to extend or modify standard transport and enveloping protocols.
    - The eXchange Protocol Designer is a powerful graphical modeling tool that lets you apply business logic to message flow through a B2B protocol pipeline.
    - The Dialog Designer, for editing B2B business protocols, helps you choreograph message-exchange interactions between your enterprise and trading partners.
- Running on the SeeBeyond Integration Server, eXchange's B2B Host, protocol pipelines, and message-tracker application enable your enterprise to:
    - Receive, validate, process, and route inbound and outbound messages.

- Look up trading partner configurations and access custom-tailored delivery channels that use agreed-upon protocols to envelope/de-envelope, encrypt/decrypt, and compress/decompress messages.
- Generate and reconcile acknowledgments; handle and report errors; and store message history, acknowledgments, and errors in a database.

- On the Web, plugged into the Enterprise Manager framework, eXchange's administrative and monitoring tools allow you to define and manage trading partner profiles, track message access, view message history, and record/monitor the status of completed and running B2B protocol pipelines.

2.3 Architectural Overview

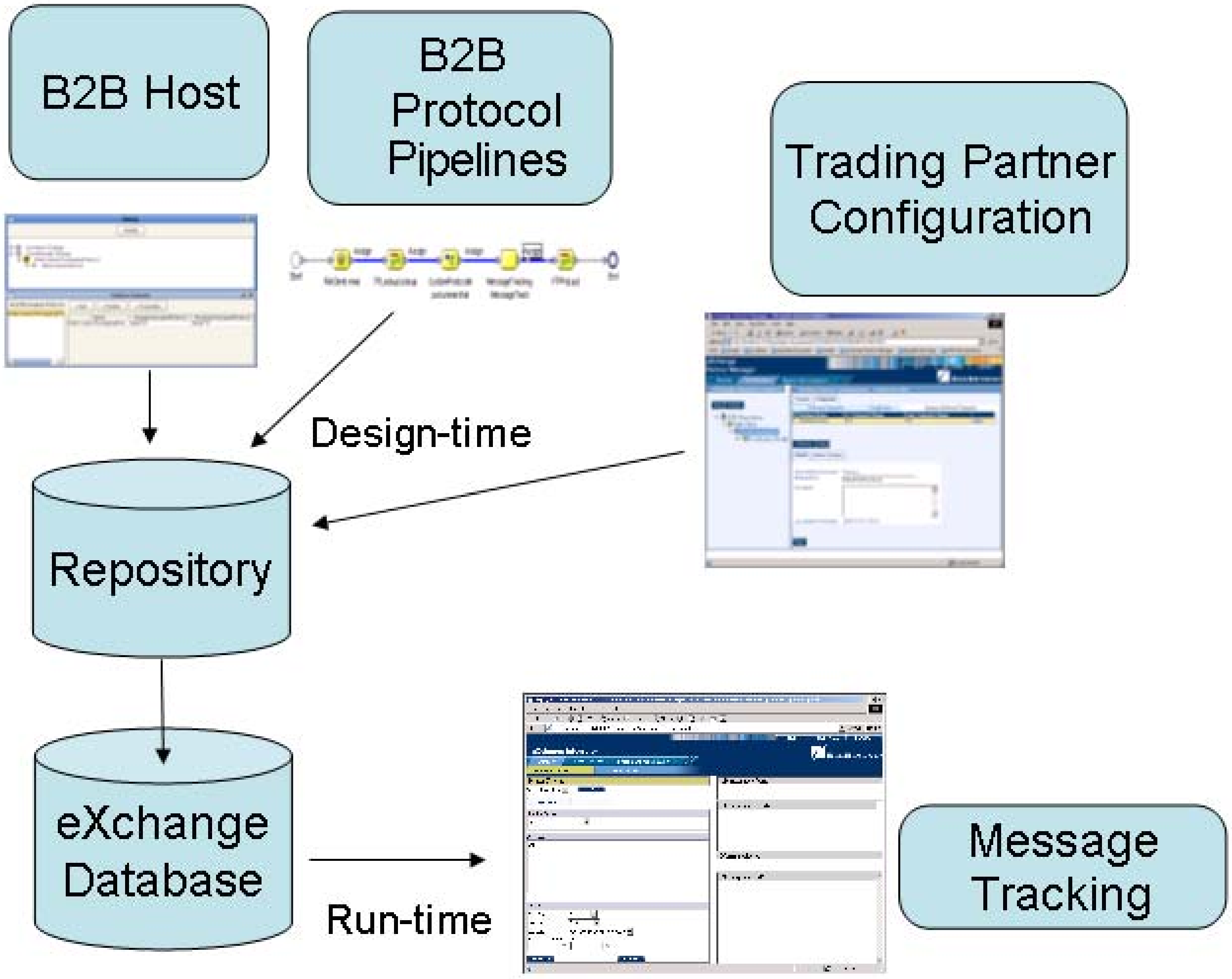
eXchange uses the following key components:

- **B2B Host Designer** — eXchange provides an additional editor within **Enterprise Designer** for setting up B2B environments. Each B2B Host provides one or more protocol-specific delivery channels that are exposed to the eXchange database via the Repository. Delivery channels provided by the B2B Host can then be accessed when specifying trading partner configurations. See **Figure 3 on page 19**.
- **Enveloping Protocols and Transport Protocols** — eXchange supplies prebuilt B2B protocol pipelines for industry-standard protocols such as AS2 and ebXML, and it also provides the flexibility of allowing the enterprise to create and configure custom protocols and protocol pipelines. See **Figure 4 on page 19**.

  eXchange also supplies **Channel Manager**, a special eWay that provides trading partner–specific integration to the enterprise. Industry-standard transport protocols (FTP, HTTP, HTTPS, SMTP) are supported by Channel Manager and other eWays.
- **Trading Partner Configuration** — eXchange provides a Web-based GUI, eXchange Partner Management (ePM), for configuring and managing B2B trading partners. Each trading partner has one or more delivery channels that specify protocols to be used, with corresponding transport mechanisms—parameters such as encryption and signature, acknowledgment-handling preferences, and so forth. See **Figure 5 on page 20**.
- **eXchange database** — eXchange uses an Oracle database to mediate retrieval of trading partner information and store run-time information for message tracking.
- **Message Tracking**— eXchange provides a MessageTracker application that can be combined with other processes in a project by dragging it into the connectivity map, as well as a Web-based message tracking GUI with powerful filtering and searching capabilities. See **Figure 6 on page 20**.

The interaction of these key components is illustrated in Figure 2.

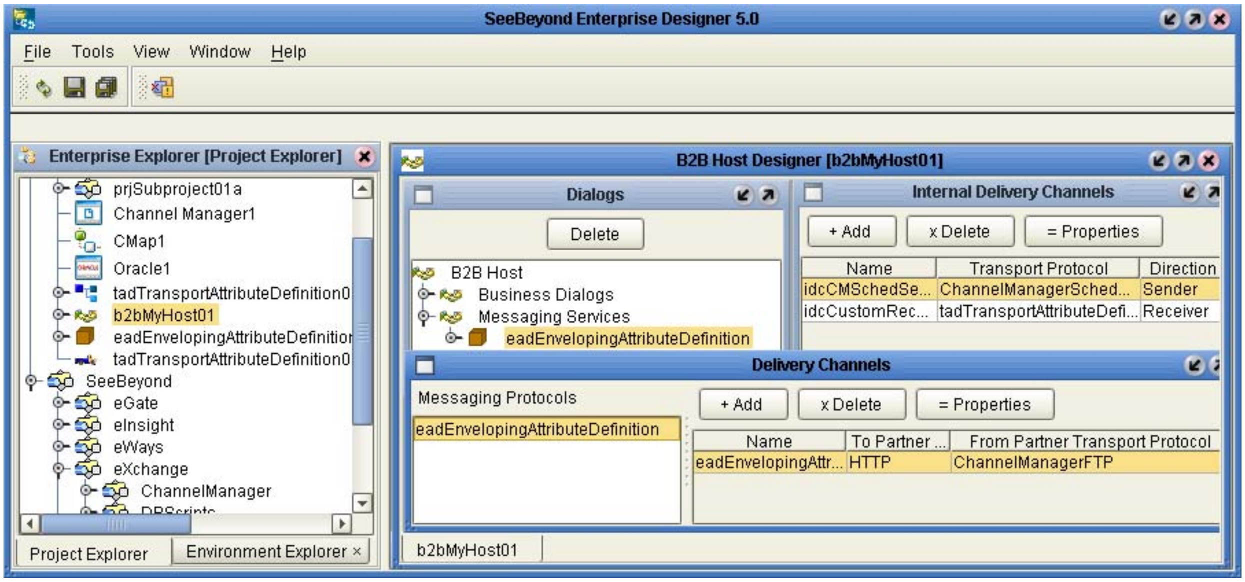
Figure 2 eXchange Architecture



2.4 Overview of Features

The illustrations below indicate some of the features provided by the various GUIs.

Figure 3 B2B Host Designer in Enterprise Designer


SeeBeyond Enterprise Designer 5.0
File Tools View Window Help
Enterprise Explorer [Project Explorer]
prjSubproject01a
Channel Manager1
CMap1
Oracle1
tadTransportAttributeDefinition0
b2bMyHost01
eadEnvelopingAttributeDefinitior
tadTransportAttributeDefinition0
SeeBeyond
eGate
eInsight
eWays
eXchange
ChannelManager
DBScripts
Project Explorer
Environment Explorer ×
B2B Host Designer [b2bMyHost01]
Dialogs
Delete
B2B Host
Business Dialogs
Messaging Services
eadEnvelopingAttributeDefinition
Internal Delivery Channels
+ Add
x Delete
= Properties
Delivery Channels
Messaging Protocols
eadEnvelopingAttributeDefinition
+ Add
x Delete
= Properties
b2bMyHost01

Name	Transport Protocol	Direction
idcCMSchedSe...	ChannelManagerSched...	Sender
idcCustomRec...	tadTransportAttributeDefi...	Receiver

Name	To Partner ...	From Partner Transport Protocol
eadEnvelopingAttr...	HTTP	ChannelManagerFTP

Figure 4 Prebuilt B2B Protocol Pipeline (for AS2 Inbound) in Enterprise Designer

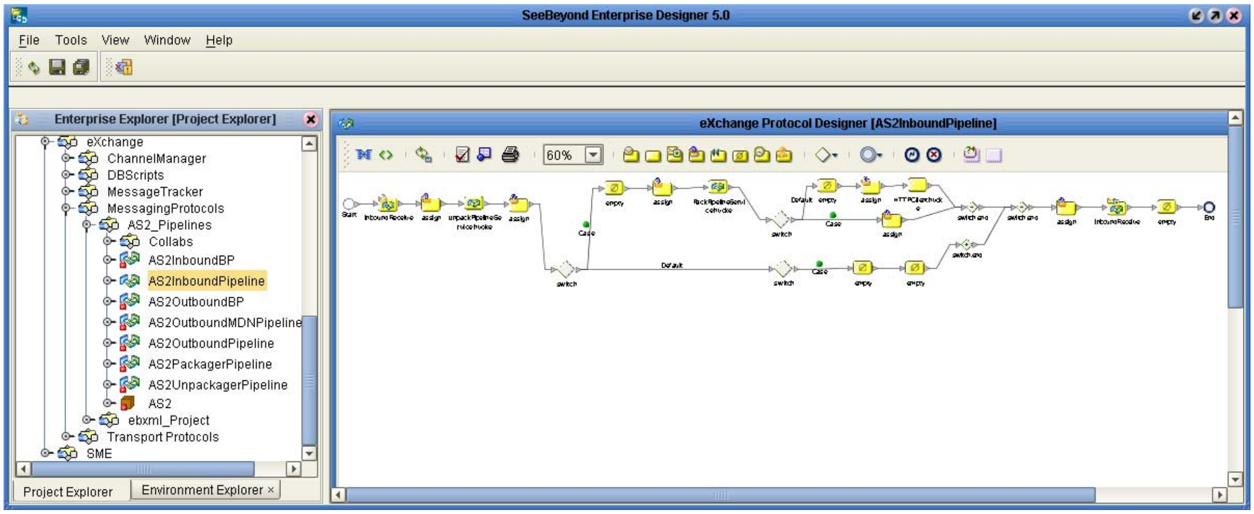
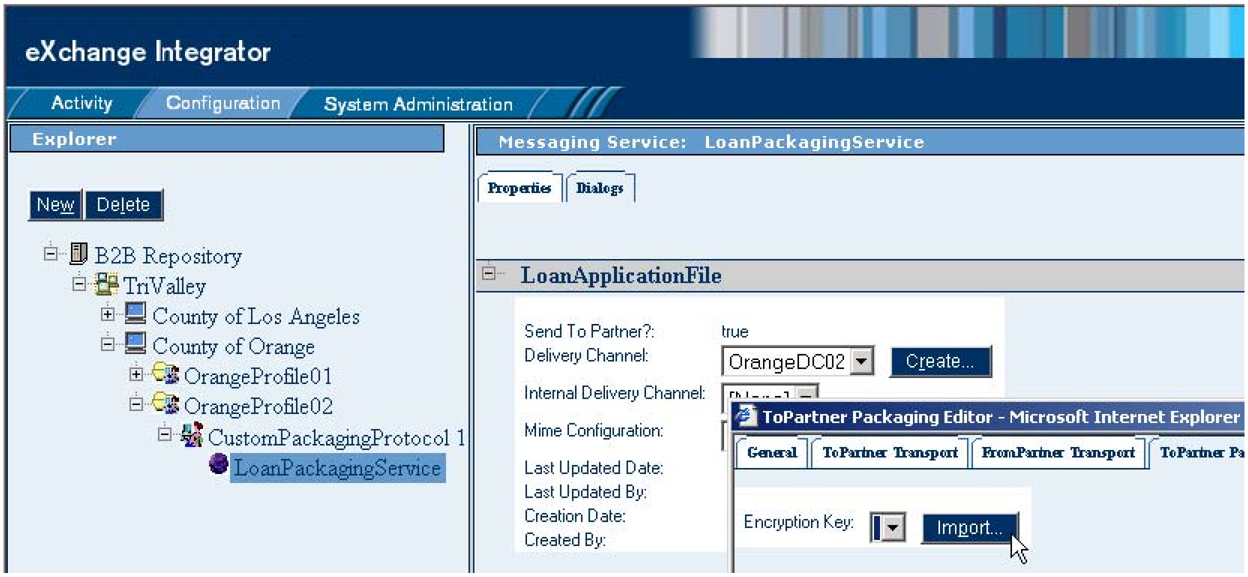
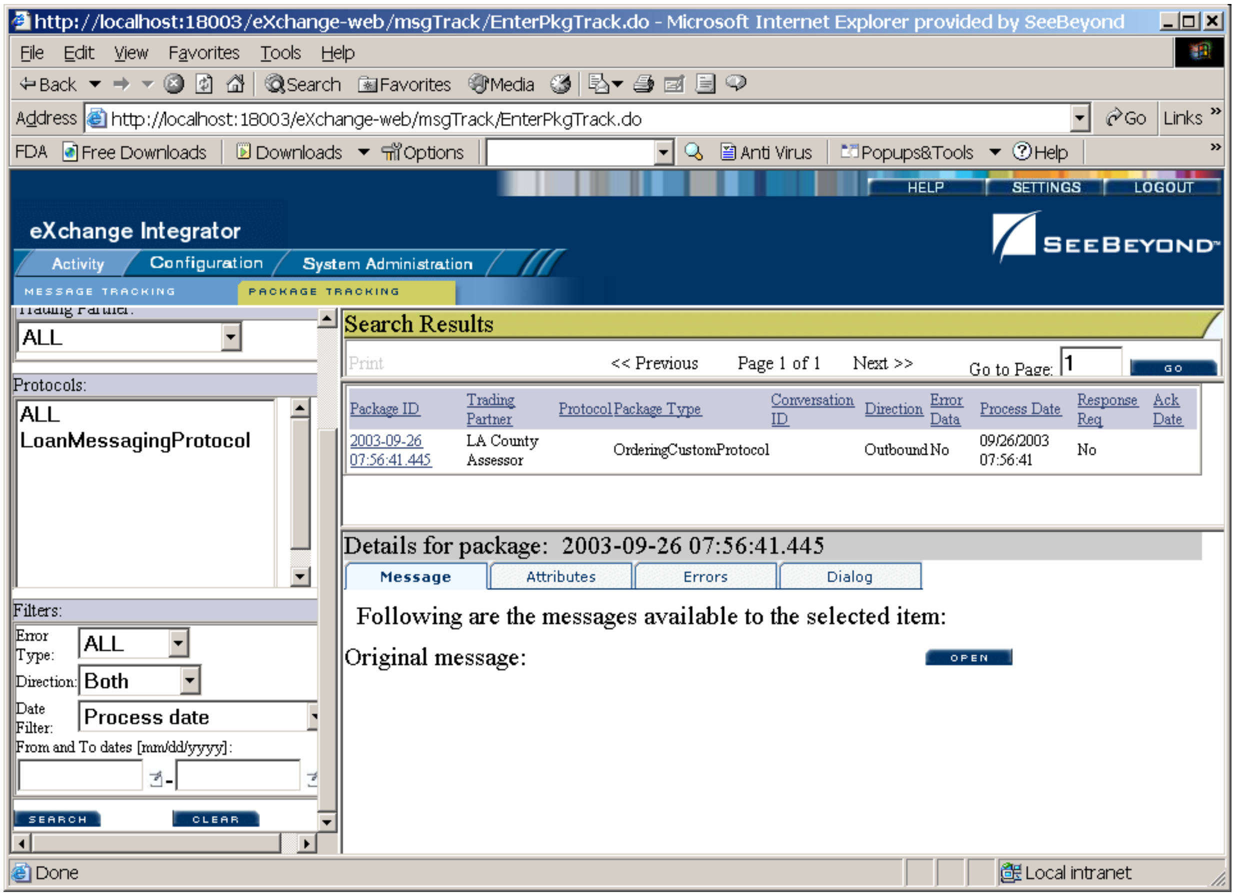


Figure 5 eXchange Trading Partner Configuration



Send To Partner?: true
Delivery Channel: OrangeDC02
Internal Delivery Channel:
Mime Configuration:
Last Updated Date:
Last Updated By:
Creation Date:
Created By:
Encryption Key:

Figure 6 eXchange Message Tracking



Address: http://localhost:18003/eXchange-web/msgTrack/EnterPkgTrack.do

Trading Partner: ALL

Protocols: ALL, LoanMessagingProtocol

Filters:
Error Type: ALL
Direction: Both
Date Filter: Process date
From and To dates [mm/dd/yyyy]:

Search Results

Page 1 of 1 Go to Page: 1

Package ID	Trading Partner	Protocol	Package Type	Conversation ID	Direction	Error Data	Process Date	Response Req	Ack Date
2003-09-26 07:56:41.445	LA County Assessor		OrderingCustomProtocol		Outbound	No	09/26/2003 07:56:41	No	

Details for package: 2003-09-26 07:56:41.445

Following are the messages available to the selected item:

Original message:

2.5 Summary of Features

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

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

The trading partner management facility is provided via a Web interface. For easy interoperability, trading partners can be configured by importing Collaboration Protocol Agreements (CPAs); or trading partner profiles can be configured manually. Each trading partner profile is identified by a unique ID determined by the enterprise, and delivery channels can be configured for acknowledgments, compression, industry-standard encryption and decryption, and nonrepudiation.

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

Chapter 3

Installing eXchange

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

3.1 System Requirements

This section lists system requirements and database requirements. The Readme.txt file on the product media contains the most up-to-date operating system requirements for each supported platform.

3.1.1. Platform Support

eXchange supports the following operating systems:

- Microsoft Windows Server 2003, Windows XP SP1a, and Windows 2000 SP3
- Sun Solaris 8 and 9 with required patches
- IBM AIX 5.1 and 5.2 with required maintenance level patches
- HP-UX 11.0 and 11i with required patches
- HP Tru64 V5.1A with required patches
- Red Hat Linux 8 (Intel) and Linux Advanced Server 2.1 (Intel version)

3.1.2. Database Support

Database for eXchange Partner Management and Message Tracking

The eXchange database provide a run-time persistent store for trading partner management and message tracking. eXchange supports the following databases:

- Oracle 8.1.7
- Oracle 9.01
- Oracle 9.2

Database for eInsight Engine

In addition, eXchange uses the eInsight engine (supplied with eXchange) to collect and persist data from your B2B protocol pipelines, so that Enterprise Manager can monitor business processes and B2B protocol pipelines. The eInsight engine supports the following databases:

- SQL Server 2000
- Sybase 12.5
- Oracle 8.1.7
- Oracle 9i

3.2 Installation Steps

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

3.2.1. Uploading eXchange to the Repository

Before you begin

- A Repository server must be running on the machine where you will be uploading the eXchange product files.
- The following ICAN products must have already been uploaded to this Repository:
  - eGate Enterprise Designer (eGate.sar)
  - Batch eWay adapter (BatcheWay.sar)
  - File eWay adapter (FileeWay.sar)
  - HTTP eWay adapter (HTTPeWay.sar)
  - Oracle eWay adapter (OracleeWay.sar)

To upload eXchange product files to the Repository

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

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

   where

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

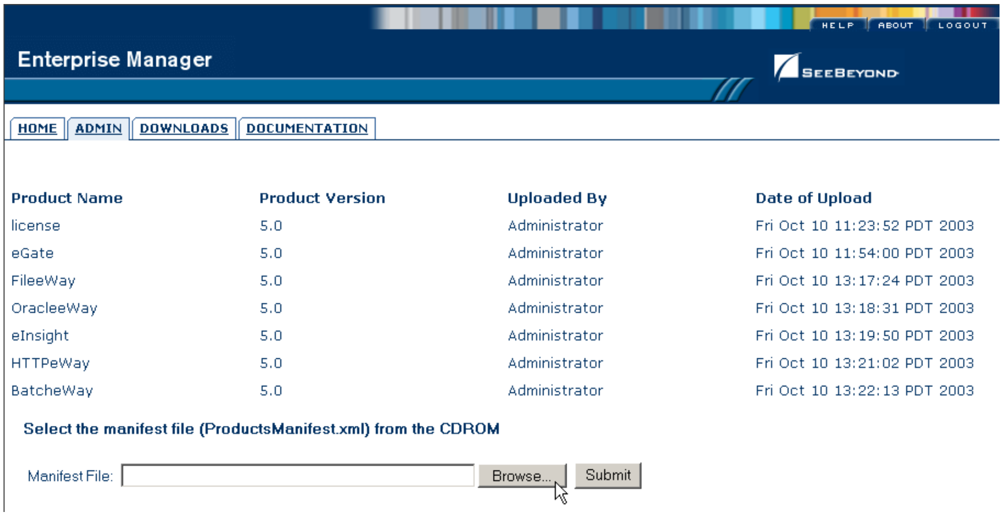
   For example, the URL you enter might look like either of the following:

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

2 On the Enterprise Manager **SeeBeyond Customer Login** page, enter your username and password.

3 When Enterprise Manager responds, click the **ADMIN** tab. See **Figure 7 on page 24**.

Figure 7 Enterprise Manager ADMIN page



Product Name	Product Version	Uploaded By	Date of Upload
license	5.0	Administrator	Fri Oct 10 11:23:52 PDT 2003
eGate	5.0	Administrator	Fri Oct 10 11:54:00 PDT 2003
FileeWay	5.0	Administrator	Fri Oct 10 13:17:24 PDT 2003
OracleeWay	5.0	Administrator	Fri Oct 10 13:18:31 PDT 2003
eInsight	5.0	Administrator	Fri Oct 10 13:19:50 PDT 2003
HTTPeWay	5.0	Administrator	Fri Oct 10 13:21:02 PDT 2003
BatcheWay	5.0	Administrator	Fri Oct 10 13:22:13 PDT 2003

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

Manifest File:

4 In the ADMIN page, click **Browse**.

5 In the **Choose file** dialog, click **ProductsManifest.xml**, and then click **Open**.

6 In the ADMIN page, click **Submit**.

The lower half of the ADMIN page lists the product files you are licensed to upload.

7 In the Products column, find **eXchange**, and then click the **Browse** button for it.

8 In the **Choose file** dialog, click **eXchange.sar**, and then click **Open**.

9 Repeat the previous two steps for other eXchange-related **.sar** files you are licensed to upload, such as SME Web Services or OTD libraries.

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

10 In the ADMIN page, click **Upload Now**. See Figure 8.

Figure 8 upload now button

3.2.2. Refreshing Enterprise Designer with eXchange

Before you begin

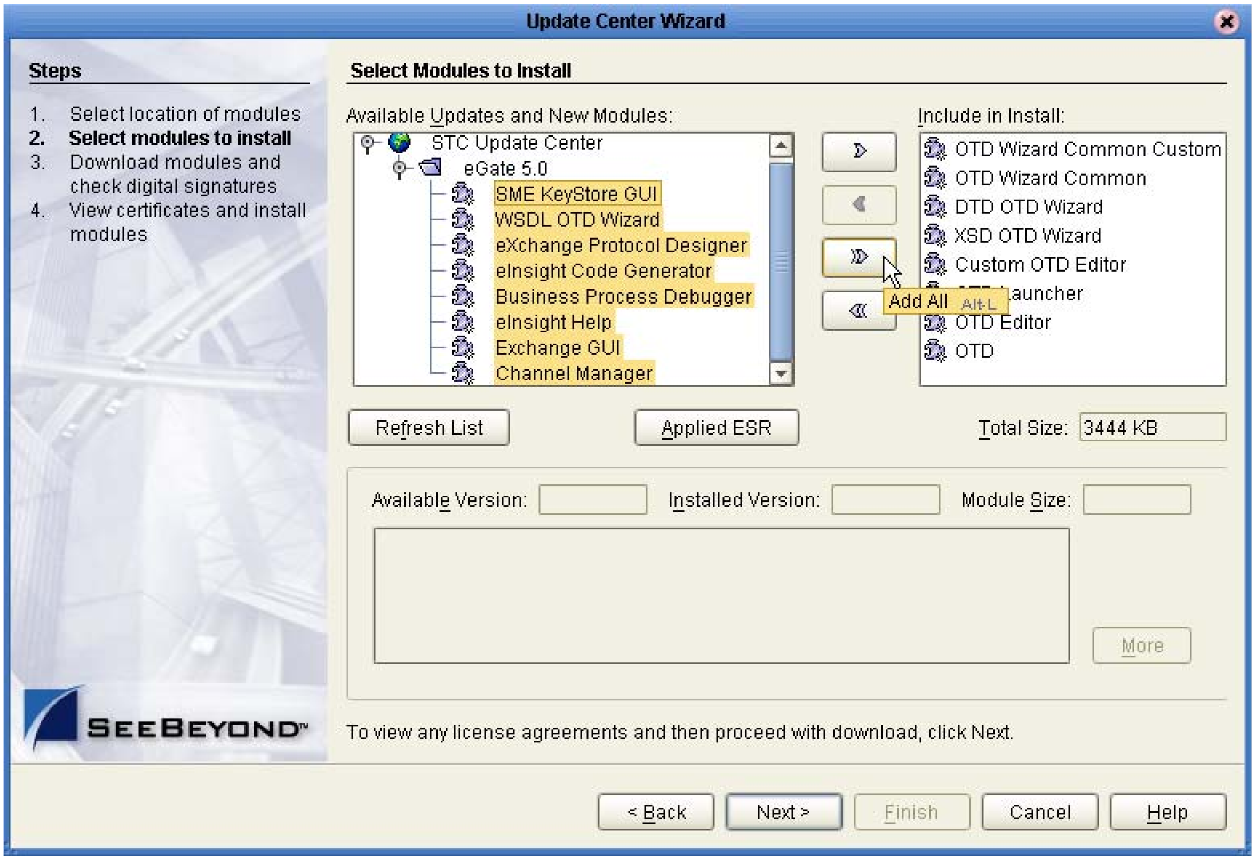
- You must have already downloaded and installed Enterprise Designer.
- A Repository server must be running on the machine where you uploaded the eXchange product files.

To refresh an existing installation of Enterprise Designer

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

   The Update Center shows a list of components ready for updating. See Figure 9.

Figure 9 Update Center Wizard: Select Modules to Install



Update Center Wizard

Steps

1. Select location of modules
2. **Select modules to install**
3. Download modules and check digital signatures
4. View certificates and install modules

Select Modules to Install

Available Updates and New Modules:

STC Update Center
eGate 5.0
SME KeyStore GUI
WSDL OTD Wizard
eXchange Protocol Designer
eInsight Code Generator
Business Process Debugger
eInsight Help
Exchange GUI
Channel Manager

Include in Install:

OTD Wizard Common Custom
OTD Wizard Common
DTD OTD Wizard
XSD OTD Wizard
Custom OTD Editor
OTD Launcher
OTD Editor
OTD

Total Size: 3444 KB

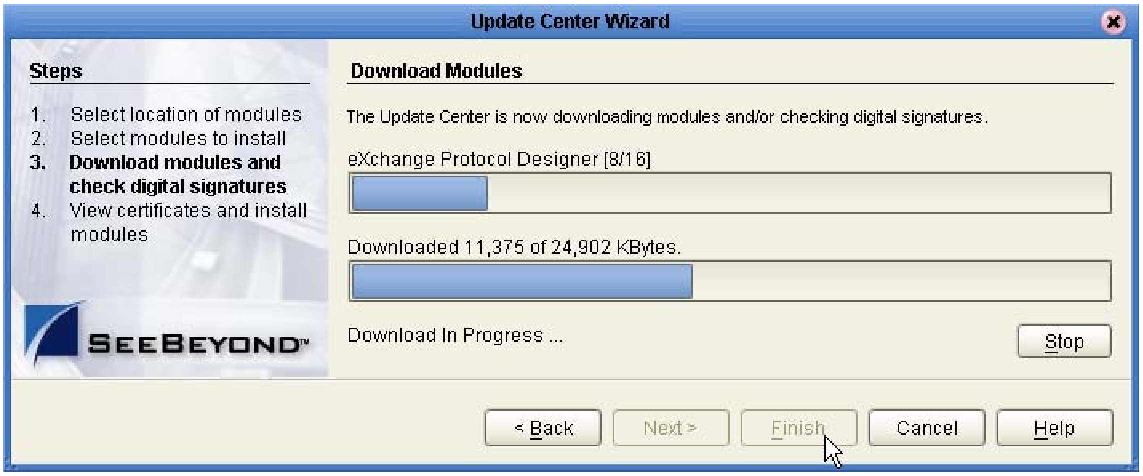
Available Version:
Installed Version:
Module Size:

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

3. Click **Add All** (the button with a doubled chevron pointing to the right).

   All modules move from the Available/New pane to the **Include in Install** pane.
4. Click **Next** and, in the next window, click **Accept** to accept the license agreement.
5. When the progress bars indicate the download has ended, click **Next**. See Figure 10.

Figure 10 Update Center Wizard: Progress Bars



Update Center Wizard

Steps

1. Select location of modules
2. Select modules to install
3. **Download modules and check digital signatures**
4. View certificates and install modules

Download Modules

The Update Center is now downloading modules and/or checking digital signatures.

eXchange Protocol Designer [8/16]

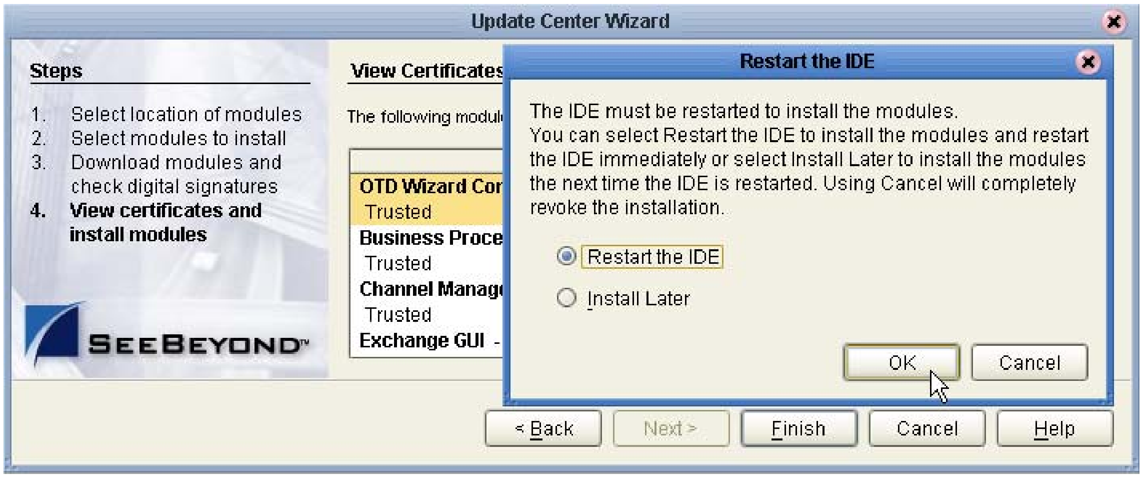
Downloaded 11,375 of 24,902 KBytes.

Download In Progress ...

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

7 When prompted to restart Enterprise Designer, click **OK**. See Figure 11.

Figure 11 Update Center Wizard: Restart Enterprise Designer



Update Center Wizard

Steps

1. Select location of modules
2. Select modules to install
3. Download modules and check digital signatures
4. **View certificates and install modules**

View Certificates

The following modul

OTD Wizard Cor Trusted
Business Proce Trusted
Channel Manag Trusted
Exchange GUI -

Restart the IDE

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

● Restart the IDE

○ Install Later

When Enterprise Designer restarts, the installation of eXchange Integrator is complete, and you can use all eXchange tools provided on the Enterprise Designer framework. However, to take advantage of the Web-based GUIs for monitoring B2B protocol pipelines, you must also enable *persistence* for the eInsight engine.

3.3 Persistence — Overview

eXchange provides database scripts to create the database schemas for both the eInsight engine and eXchange itself.

- The eXchange database schema allows you to collect and persist data about your trading partner profiles, and also allows you to track message delivery history.

  For eXchange, the areas to be configured are:

  - **Creating and Configuring the eXchange Database** on page 27
  - **Extracting and Customizing Database Scripts for eXchange** on page 28
  - **Running Database Scripts to Set Up the eXchange Database** on page 28

- The eInsight database schema allows you to collect and persist data from your B2B protocol pipelines; because the data is persisted, you can also use Enterprise Manager to monitor business processes and B2B protocol pipelines.

  For eInsight, the areas to be configured are:

  - **Configuring the eInsight Run-Time Engine** on page 30
  - **Extracting and Running Database Scripts for eInsight Engine** on page 31

3.3.1. Creating and Configuring the eXchange Database

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

1. Step 1 (“Operations”): Choose **Create a database**.
2. Step 2 (“Database Templates”): Choose **New Database**.
3. Step 3 (“Database Identification”): Enter (for example) **eXchange**
4. Step 4 (“Database Features”): Deselect all checkboxes and reply **Yes** to all prompts.
5. Step 5 (“Database Connection Options”): Choose **Dedicated** [...].
6. Step 6 (“Initialization Parameters”): Keep all values unchanged.
7. Step 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. Step 8 (“Creation Options”): Choose **Create Database**, and then click Finish.

After this, to set up the eXchange database, you will extract the files supplied in the Database Scripts project folder, edit scripts so they have the correct parameters for your Oracle database setup, and run the scripts to set up and initialize the database.

Extracting and Customizing Database Scripts for eXchange

To extract and customize the database scripts

1. In Enterprise Explorer, in the project tree, expand the following folders: **SeeBeyond > eXchange > DBScripts**
2. Right-click each of the following files and, on the popup context menu, click **Export**; then use the **Save** dialog box to save each file to a local directory.
   - **createdb.sql**
   - **createtablespaces.sql**
   - **createuser.sql**
   - **eXchange50Runtime.sql**
   - **in_user_seq.sql**
   - **x12_datamodel.sql**
3. If your Oracle location is not **c:\oracle\oradata**, or if your database instance name (SID) is other then **eXchange**, then open the **createtablespaces.sql** file and make the appropriate change or changes in the first line.

Running Database Scripts to Set Up the eXchange Database

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

To run the database scripts that install the schema

1. Open a command prompt, change directories to the location where you saved the **.sql** scripts, and enter the following SQL*Plus command:

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

   where:

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

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

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

   ```
   sqlplus system/manager1@eX50 @createtablespaces.sql
   sqlplus system/oraclePW@eXchange @createtablespaces.sql
   ```

   When this finishes, you will have created new tablespaces.
2. In the command prompt, enter the following SQL*Plus command:

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

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

   Here is an example of a valid command:

   ```
   sqlplus system/myPassWd@eX500DB @createuser.sql
   ```
3. In response to the system prompt, enter the username. For example: **ex_admin**

4 After the system finishes dropping tables, it asks for username and password so it can create the eXchange Administrator for the database instance). For example: **ex_admin ex_admin**

5 After the system finishes creating the eXchange Administrator, it asks a third time for the username, this time so it can grant resources and connect. Enter: **ex_admin** (or whatever you use asthe username).

6 After running the previous two SQL script, there is one more. In the command prompt, enter the following SQL*Plus command:

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

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

The system populates the tables, and you are now ready to use the database instance as your eXchange database. You can create Oracle OTDs based on this database, and use Enterprise Manager GUIs for trading partner configuration and message tracking.

3.3.2. Configuring the eInsight Run-Time Engine

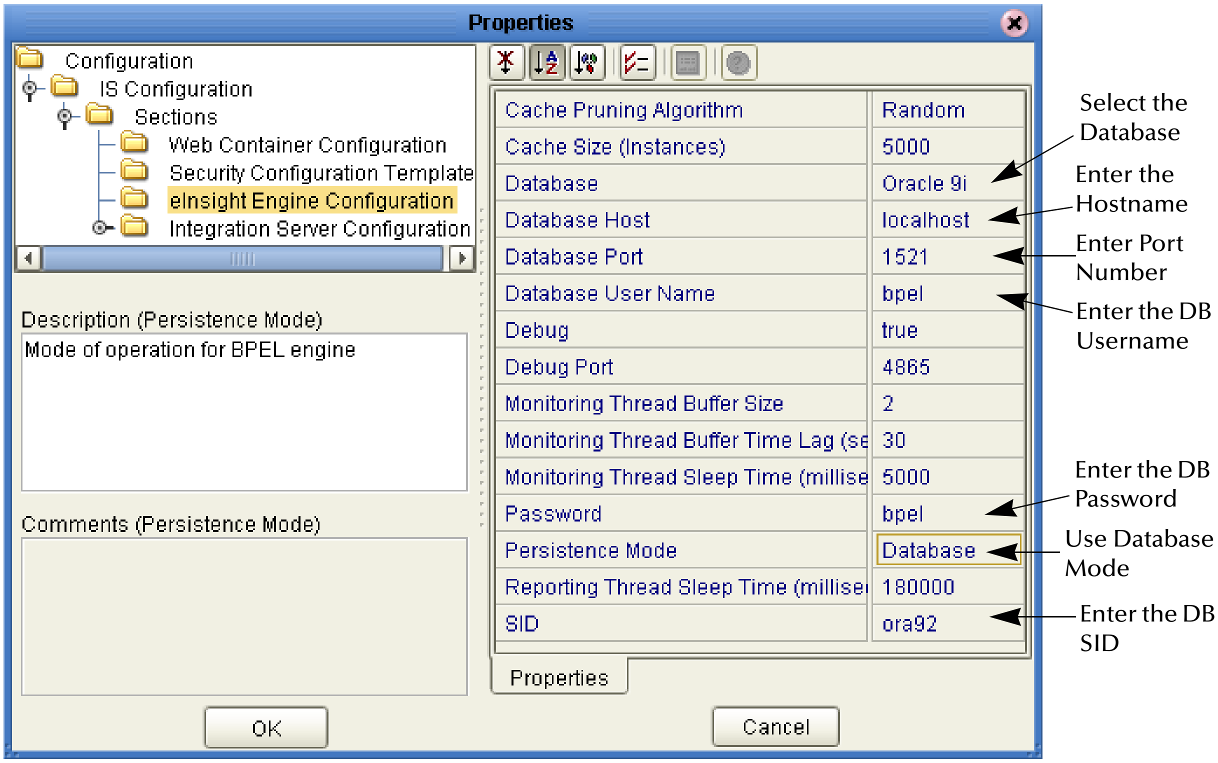
To configure the eInsight engine in an integration server

1 In Environment Explorer, open the environment and logical host, right-click the **Integration Server** (IS), and, on the popup context menu, click **Properties**.

The Properties dialog opens.

2 Open the Configuration > IS Configuration > **Sections** tree and select **eInsight Engine Configuration**. See Figure 12 on page 30.

Figure 12 eInsight Engine Configuration


Properties
Configuration
IS Configuration
Sections
Web Container Configuration
Security Configuration Template
eInsight Engine Configuration
Integration Server Configuration
Properties
OK
Cancel

Description (Persistence Mode)
Mode of operation for BPEL engine

Comments (Persistence Mode)

Cache Pruning Algorithm	Random
Cache Size (Instances)	5000
Database	Oracle 9i
Database Host	localhost
Database Port	1521
Database User Name	bpel
Debug	true
Debug Port	4865
Monitoring Thread Buffer Size	2
Monitoring Thread Buffer Time Lag (se	30
Monitoring Thread Sleep Time (millise	5000
Password	bpel
Persistence Mode	Database
Reporting Thread Sleep Time (millise	180000
SID	ora92

Select the Database

Enter the Hostname

Enter Port Number

Enter the DB Username

Enter the DB Password

Use Database Mode

Enter the DB SID

3 Configure the following to connect to a database:

- **Database** - Select Sybase 12.5, Oracle 8.1.7, SQL Server 2000 or Oracle 9i.
- **Database Host** - Enter the name of the machine where your database resides.
- **Database Port** - Enter the port number to connect to your database.
- **Database User Name** - Enter the User Name for your database.
- **Password** - Enter the password for your database user.
- **Persistence Mode** - Select Database to use persistence.
- **SID** - Enter the database name or SID.

Note: *The other eInsight Engine settings can be left as default, or you can change the options to your preference.*

3.3.3. Extracting and Running Database Scripts for eInsight Engine

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

To extract the database schema scripts

1. In Enterprise Explorer, in the project tree, expand the following folders: **SeeBeyond > eInsight** > **DatabaseScripts**
2. Right-click the **.zip** file associated with the appropriate database (oracle.zip, sqlserver.zip, or sybase.zip) and, on the popup context menu, click **Export**.
3. Extract the **.zip** file contents to a local folder, which will contain:
    - **install_db.bat**—Creates the tablespace, users, tables, stored procedures, and any initial value.
    - **uninstall_db.bat**—Reverses what the install_db.bat script creates; that is, it drops tables and users, and deletes stored procedures).
    - *database-specific* **.sql** *scripts*—Called by the install_db.bat and uninstall_db.bat commands. For example:, create_tables.sql, drop_tables.sql.
    - **Readme.txt**—Additional instructions specific to your database application.

To install the database schema

1. On Windows, open a command prompt and change to the directory where **install_db.bat** is located.

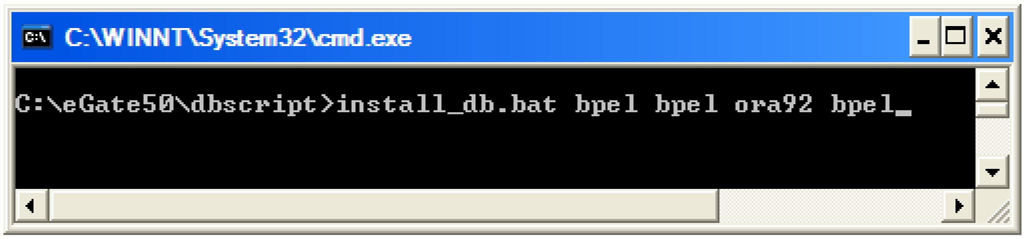
Important: *The database user that executes these scripts must have permission to create tables.*

2. Enter the following at the command prompt, as shown in Figure 13:

```
install_db <user> <password> <tns> <new_user>
```

- *<user>* is the database username
- *<password>* is the password for this database user
- *<tns>* is the database or TNS name
- *<new_user>* is the name of the account that eInsight uses

Figure 13 Install_db.bat



```
C:\eGate50\dbscript>install_db.bat bpel bpel ora92 bpel_
```

Note: *The default user and password created from these scripts is:* ***bpel****. You can modify the user, password, and disk space allocated for tables and user premissions, but the table and column definitions should not be modified.*

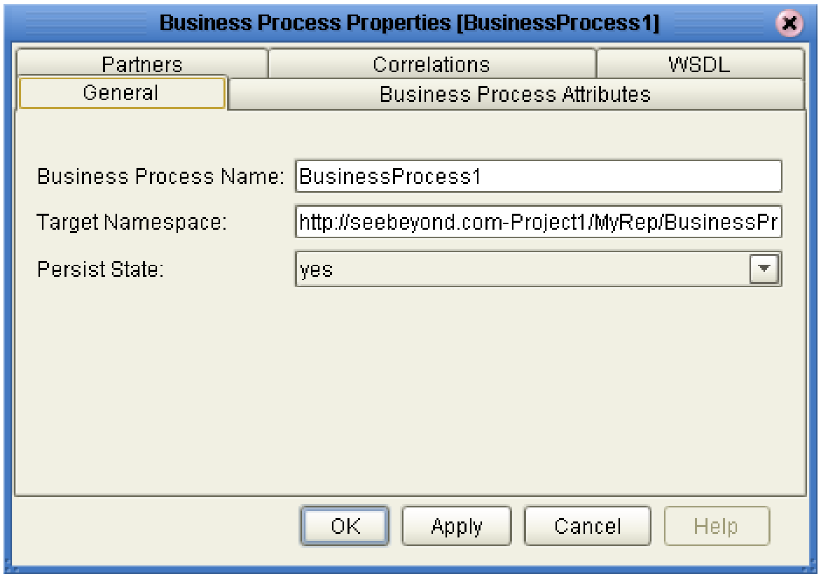
To create specific tables for B2B protocol pipelines

1. In Enterprise Designer, in the project tree, right-click the protocol pipeline and, on the popup context menu, click **Open Property Sheet.**

   You set persistence state for individual protocol pipelines. The default setting is **no**.

2. Change the **Persist State** value to **yes**, as shown in Figure 14, and then click **OK**.

Figure 14 Business Property Sheet



Business Process Properties [BusinessProcess1]
Partners
Correlations
WSDL
General
Business Process Attributes
OK
Apply
Cancel
Help

Business Process Name: BusinessProcess1

Target Namespace: http://seebeyond.com-Project1/MyRep/BusinessPr

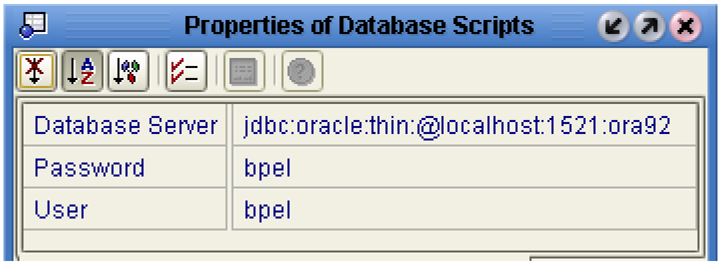
Persist State: yes

3. On the main Enterprise Designer toolbar, click **Save All**.

   This creates a **Database Install Script** option under your protocol pipeline.

4. Right-click **Database Scripts** and fill in the correct information for your database from the dialog that appears (as shown in Figure 15 on page 32).

Figure 15 Properties of Database Scripts



Database Server	jdbc:oracle:thin:@localhost:1521:ora92
Password	bpel
User	bpel

To run the database script for a B2B protocol pipeline

1. In the project tree, open your protocol pipeline and its **Database Scripts** folder.

2. In the **Database Scripts** folder, right-click the appropriate database and click **Run.**

The scripts complete the database creation process. Once persistence is configured, you can use the Enterprise Manager to monitor your protocol pipeline instances.

Chapter 4

Using eXchange in Enterprise Designer

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

In this chapter

- Steps to set up ***attribute definitions*** for transport and enveloping protocols
- Steps to set up ***B2B business protocols***, which reference enveloping protocols
- Steps to set up ***B2B hosts*** (which reference B2B business protocols) and their ***delivery channels*** (which reference enveloping and transport protocols)
- Steps to preactivate a ***B2B host environment*** with MessageTracker, to set the stage for later activation of projects using Channel Manager.
- Steps to configure eXchange deployment profiles and environment components , such as the B2B host's certificates and truststores.

4.1 Setting Up Protocol Attribute Definitions

The section explains how to create and configure attribute definitions, and how to generate the corresponding OTDs, for the following project-level components:

- Transport attribute definitions
- Enveloping attribute definitions

4.1.1. Creating and Configuring Transport Attribute Definitions

In general, a *protocol* is a code of behavior; a framework for interpretation and communication that is agreed upon by all parties. It prescribes 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. Some of the common standard transport protocols presupplied with eXchange include: FILE, FTP, HTTP, HTTPS, and SMTP. 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.

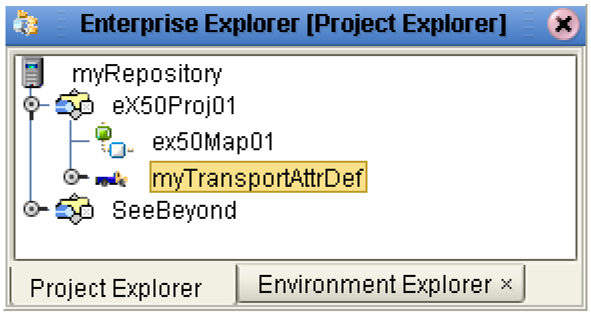
In addition to the standard transport protocols noted above, eXchange users can employ ***transport attribute definitions*** to create modifications or extensions of the standard transport protocols. These variants are called *custom* transport protocols.

To create an transport attribute definition

1. In Enterprise Designer with the Project Explorer tab active, in the project tree, right-click the project or subproject where the transport attribute definition will reside.
2. On the popup context menu, point at **New**, and then click **B2B Transport Attributes Definition**.

   The project tree displays the new component, whose icon suggests a blue transport vehicle. Its default name is **B2BTransportAttributesDefinition<*n*>**, but you can rename it however you like.

Figure 16 Transport Attribute Definition



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

To configure a transport attribute definition

1. In the project tree, right-click the transport attribute definition you want to modify.

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

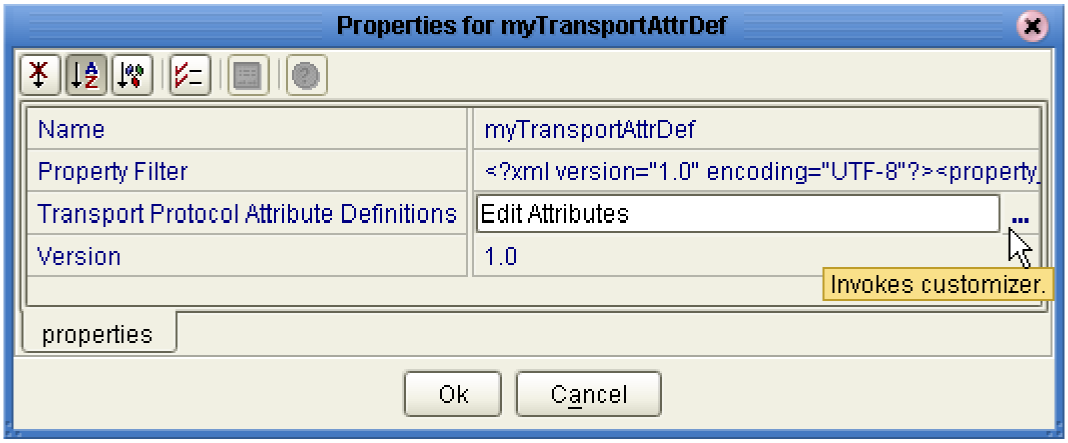
2. On the popup context menu, click **Properties**.

   The properties dialog appears. See Table 2 and Figure 17.

Table 2 Properties for Transport Attribute Definition

Property Name	Initial or Default Value	Description
Name	(component name)	Free text; this is the component name displayed in the project tree.
Version	**1.0**	Free text; reflects whatever version naming conventions you use. This is ***independent*** of the Repository-assigned versioning, which tracks check-outs/check-ins.
Extended Attribute Definitions	**null** (initially)	An XML string that holds the values displayed in the **Attribute Definitions** dialog. To open the dialog, click the ellipsis **[...]** on the far right.
Property Filter	(XML string)	An XML string that holds the values displayed in the **Property Filter** dialog. To open the dialog, click the ellipsis **[...]** on the far right.

Figure 17 Properties for Transport Attributes Definition



Properties for myTransportAttrDef

Name	myTransportAttrDef
Property Filter	<?xml version="1.0" encoding="UTF-8"?><property
Transport Protocol Attribute Definitions	Edit Attributes ...
Version	1.0

3 To the far right of the value for Extended Attribute Definitions, click the ellipsis **[...]**.

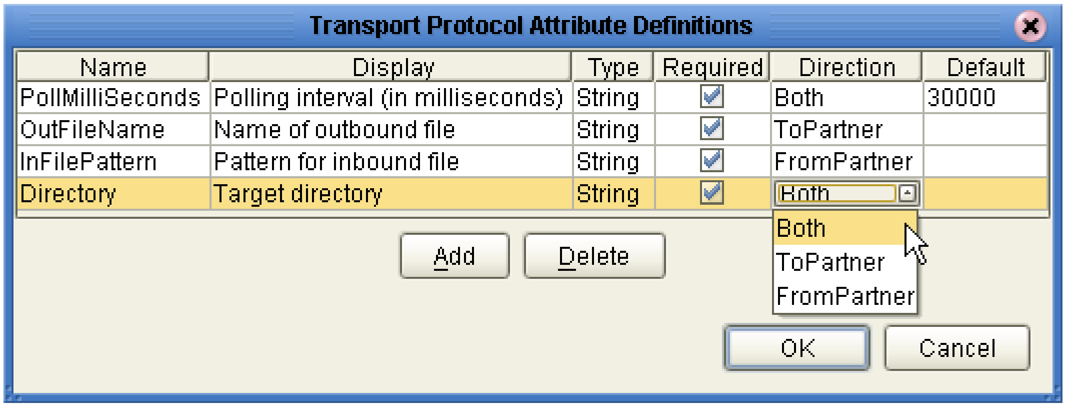
The **Extended Attribute Definitions** dialog appears. You use this dialog to create and set attributes. It is these attributes that determine the OTD nodes and govern the appearance and behavior of the parameters displayed in eXchange Partner Management when configuring external delivery channels for a trading partner (in the ToPartner Transport and FromPartner Transport subtabs).

4 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 eXchange Partner Management.
- For **Required**, select or clear the box according to whether you want the parameter to be a required or optional entry. (In the ePM GUI, parameters that have been designated as required are flagged with a red asterisk.)
- For **Direction**, choose ToPartner, FromPartner, or Both according to whether you want the parameter to appear with the ToPartner parameters, FromPartner parameters, or both.
- For **Default**, you can optionally enter a default value that will appear in eXchange Partner Manager (ePM). This is the value that will be used if it is not overridden during parameter configuration in the ePM GUI.

Attributes for a sample file-based transport attribute definition are shown in Figure 18.

Figure 18 Attribute Definitions for a Custom Protocol



Name	Display	Type	Required	Direction	Default
PollMilliSeconds	Polling interval (in milliseconds)	String	☑	Both	30000
OutFileName	Name of outbound file	String	☑	ToPartner	
InFilePattern	Pattern for inbound file	String	☑	FromPartner	
Directory	Target directory	String	☑	Both	

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

6 For **Property Filter**: If you want to change the packaging filters for this custom protocol (by default, both encryption truststore and signing certificates are filtered for both FromPartner bindings and ToPartner bindings), click the ellipsis at the far right and then, in the **Property Filter** dialog, clear the checkboxes for the filters you want disabled. When you have finished, click **OK**.

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

After you have completed these steps, the transport attribute definition will appear as a choice in the dropdown list of transport protocols when you configure the delivery channels of your B2B host, and it can now be used to generate an OTD. This OTD is used in the Business Rules Designer (lower pane of the eXchange Protocol Designer) so you can assign business logic from one activity to the next in your protocol pipeline.

To generate or refresh the OTD for a transport attribute definition

Before you begin: If the OTD has already been generated, make sure it is checked out.

1 In Enterprise Designer with the Project Explorer tab active, in the project tree, right-click the transport attribute definition whose OTD you want to generate or refresh.

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

2 On the popup context menu, click **Generate OTD**.

The project tree displays a newly generated OTD. If the name of your transport attribute definition was *abcd*, the name of the new OTD will be ***abcd*OTD_*abcd***.

4.1.2. Creating and Configuring Enveloping Attribute Definitions

An *enveloping protocol* (sometimes called a *packaging protocol* or *messaging protocol*) provides a way of specifying how data is bundled and unbundled—for example, it is at this level that encryption, acknowledgment, and nonrepduiation are addressed. SeeBeyond provides two standard enveloping protocols (AS2 and ebXML).

In addition to these standard enveloping/packaging protocols, eXchange users can employ ***enveloping attribute definitions*** to create modifications or extensions of the standard enveloping protocols. These variants are called *custom* enveloping protocols.

To create an enveloping attribute definition

1. In Enterprise Designer with the Project Explorer tab active, in the project tree, right-click the project or subproject where the enveloping attribute definition will reside.
2. On the popup context menu, point at **New**, and then click **B2B Enveloping Attributes Definition**.
3. In **Create B2B Enveloping Attributes Definition**, enter a name and click **OK**.

   The project tree displays both the new component, whose icon suggests a brown paper package, or envelope. It also displays a new subproject; if the name of your enveloping attribute definition was *xyz*, the subproject will be named ***xyz*_Project** and will contain two B2B protocol pipelines: ***xyz*_FromPartner** and ***xyz*_ToPartner**. Do not alter these names.

This new component is useful only by virtue of its configuration—you will need to add and define attributes that govern the nodes in the OTD that will be generated from it.

To configure an enveloping attribute definition

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

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

2. On the popup context menu, click **Properties**. See Table 3.

Table 3 Properties for Enveloping Attribute Definition

Property Name	Initial or Default Value	Description
Name	(component name)	Free text; this is the component name displayed in the project tree.
Version	**1.0**	Free text; reflects whatever version naming conventions you use. This is ***independent*** of the Repository-assigned versioning, which tracks check-outs/check-ins.
InputXSD	**null**	(not yet supported)
OutputXSD	**null**	(not yet supported)
Protocol Attribute Definitions	**null** (initially)	An XML string that holds the values displayed in the **Protocol Attribute Definitions** dialog. To open the dialog, click the ellipsis **[...]** on the far right.
Enveloping Pipeline	**null**	(not yet supported)
Deenveloping Pipeline	**null**	(not yet supported)
Property Filter	(XML string)	An XML string that holds the values displayed in the **Property Filter** dialog. To open the dialog, click the ellipsis **[...]** on the far right.

3 To the far right of the value for **Protocol Attribute Definitions**, click the ellipsis **[...]**.

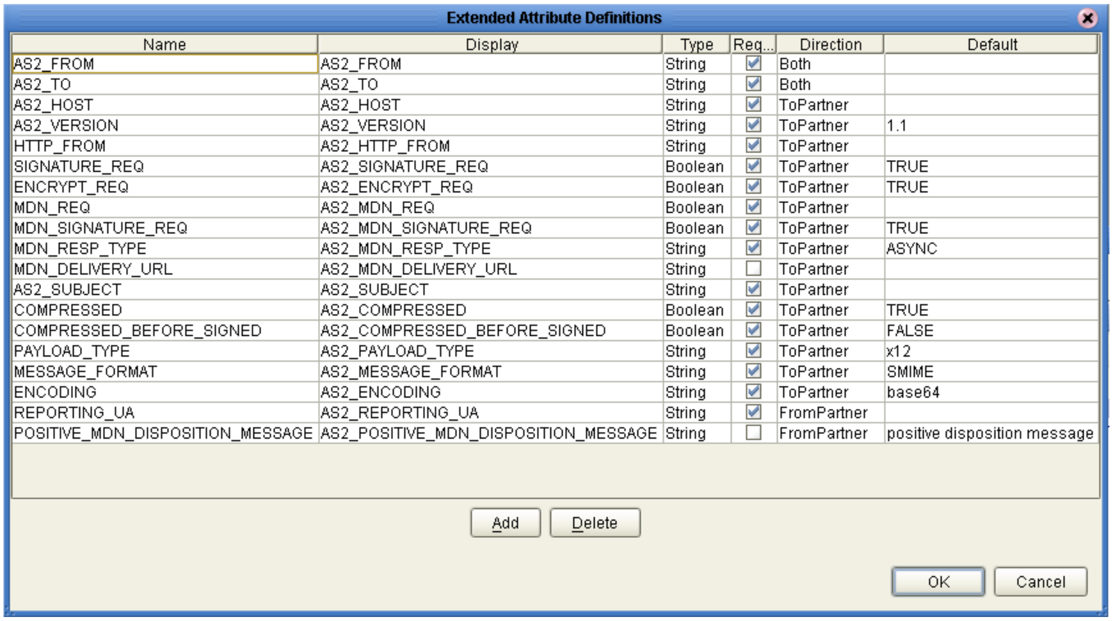
The **Protocol Attribute Definitions** dialog appears. You use this dialog to create and set the attributes; these attributes determine the OTD nodes and govern the appearance and behavior of the parameters displayed in eXchange Partner Management when configuring external delivery channels for a trading partner (in the ToPartner Transport and FromPartner Transport subtabs).

4 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 eXchange Partner Management.
- 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**, enter the default value you want this parameter to assume if it is not changed in trading partner management.

Figure 19 on page 38 shows one example of extended attributes for a custom protocol; as a different example, the extended attributes of a standard enveloping protocol (in this case, AS2 version 1.1) are shown in Figure 19.

Figure 19 Extended Attribute Definitions for AS2 Version 1.1



Extended Attribute Definitions

Name	Display	Type	Req...	Direction	Default
AS2_FROM	AS2_FROM	String	☑	Both	
AS2_TO	AS2_TO	String	☑	Both	
AS2_HOST	AS2_HOST	String	☑	ToPartner	
AS2_VERSION	AS2_VERSION	String	☑	ToPartner	1.1
HTTP_FROM	AS2_HTTP_FROM	String	☑	ToPartner	
SIGNATURE_REQ	AS2_SIGNATURE_REQ	Boolean	☑	ToPartner	TRUE
ENCRYPT_REQ	AS2_ENCRYPT_REQ	Boolean	☑	ToPartner	TRUE
MDN_REQ	AS2_MDN_REQ	Boolean	☑	ToPartner	
MDN_SIGNATURE_REQ	AS2_MDN_SIGNATURE_REQ	Boolean	☑	ToPartner	TRUE
MDN_RESP_TYPE	AS2_MDN_RESP_TYPE	String	☑	ToPartner	ASYNC
MDN_DELIVERY_URL	AS2_MDN_DELIVERY_URL	String	□	ToPartner	
AS2_SUBJECT	AS2_SUBJECT	String	☑	ToPartner	
COMPRESSED	AS2_COMPRESSED	Boolean	☑	ToPartner	TRUE
COMPRESSED_BEFORE_SIGNED	AS2_COMPRESSED_BEFORE_SIGNED	Boolean	☑	ToPartner	FALSE
PAYLOAD_TYPE	AS2_PAYLOAD_TYPE	String	☑	ToPartner	x12
MESSAGE_FORMAT	AS2_MESSAGE_FORMAT	String	☑	ToPartner	SMIME
ENCODING	AS2_ENCODING	String	☑	ToPartner	base64
REPORTING_UA	AS2_REPORTING_UA	String	☑	FromPartner	
POSITIVE_MDN_DISPOSITION_MESSAGE	AS2_POSITIVE_MDN_DISPOSITION_MESSAGE	String	□	FromPartner	positive disposition message

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

6 For **Property Filter**: If you want to change the packaging filters for your enveloping attribute definition (by default, all truststores and certificates are filtered for both FromPartner bindings and ToPartner bindings), click the ellipsis at the far right and then, in the **Property Filter** dialog, clear the checkboxes for the filters you want disabled. When you have finished, click **OK**.

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

The enveloping attribute definition can now be used to generate an OTD.

To generate or refresh the OTD for an enveloping attribute definition

Before you begin: If the OTD has already been generated, be sure it is checked out.

1 In the project tree, right-click the enveloping attribute definition whose OTD you want to generate or refresh.

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

2 On the popup context menu, click **Generate OTD**.

The project tree displays a newly generated OTD. If the name of your enveloping attribute definition was *abcd*, the name of the new OTD will be ***abcd*OTD_*abcd***.

After you have completed these steps, the enveloping attribute definition will appear as a choice in the dropdown list of enveloping protocols when you configure the delivery channels of your B2B host.

4.2 Setting Up B2B Business Protocols

A B2B business protocol is, in brief, a project-level (logical) component that sets forth rules of exchange between an enterprise and its trading partners.

The section explains how to create, populate, configure, and compile a B2B business protocol, using the **Dialog Designer** editor. These steps must be followed to allow the B2B protocol to be used by a B2B host.

To create a B2B business protocol

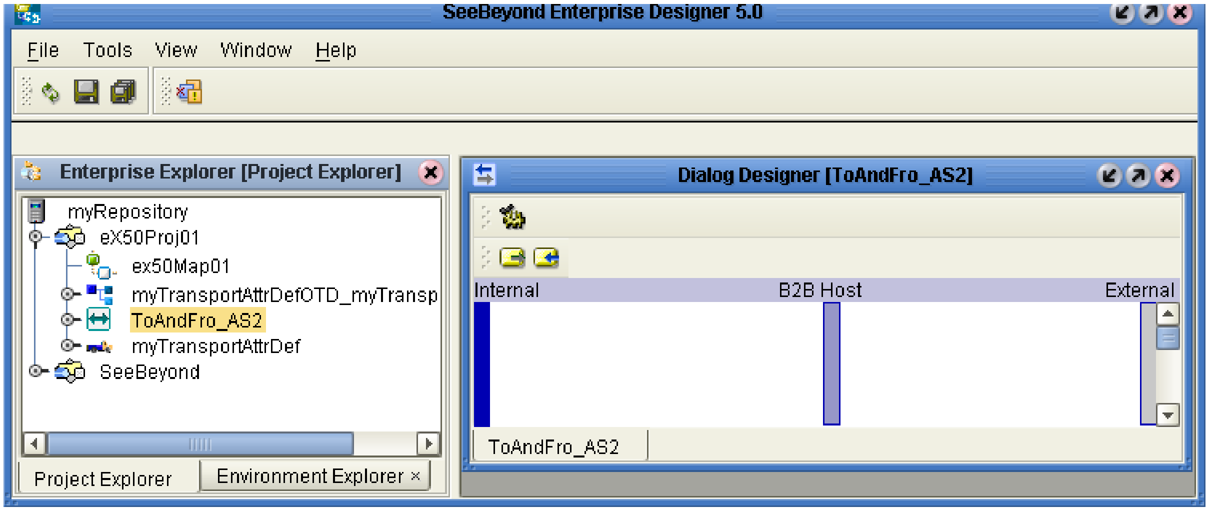
1 In Enterprise Designer with the Project Explorer tab active, in the project tree, right-click the project or subproject where the B2B business protocol will reside.

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

The project tree displays the new component, whose icon suggests bidirectional communication. By default, it is named **B2B Business Protocol<*n*>**, but you can rename it as you like.

The Dialog Designer opens to display a blank canvas. See Figure 22.

Figure 20 B2B Business Protocol and Dialog Designer

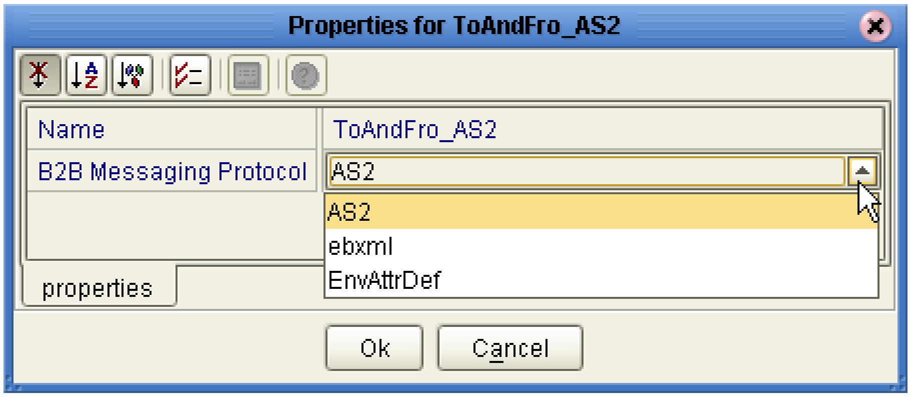


This new component still needs to be configured, populated, and compiled—you will need to add activities that describe the interchange of messages to and from the trading partner, configure it to reference an enveloping protocol, and then compile it.

To configure, populate, and compile a B2B business protocol

1. In the project tree, right-click the B2B business protocol and, on the popup context menu, click **Properties**.
2. In the Properties dialog, select an enveloping protocol from the choices displayed in the dropdown list. This list will include all standard enveloping protocols supplied by SeeBeyond (such as AS2 and ebXML), and will also show you any attribute definitions that have been created for custom enveloping protocols. See Figure 21.

Figure 21 Properties for B2B Business Protocol, Showing Enveloping Protocol Choices



Name: ToAndFro_AS2
B2B Messaging Protocol: AS2

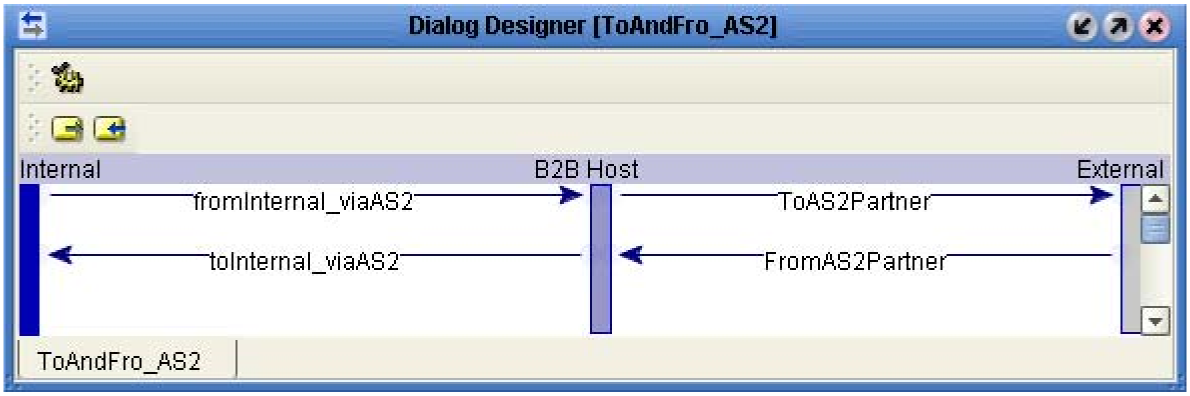
3. When you have chosen an enveloping protocol, click **OK**.
4. If Dialog Designer is not already open, open it by double-clicking the B2B business protocol you want to populate.
5. Do at least one of the following:
   - Click **from_internal** (the right-pointing arrow icon) to add an activity that originates in the enterprise and passes through the B2B host to the trading

partner. Then, double-click, the label on each of the two right-pointing arrows, editing the names to whatever you want.

- Click **message_in** (the left-pointing arrow icon) to add an activity that originates in the trading partner and passes through the B2B host to the enterprise. Then, double-click the label on each of the two left-pointing arrows, editing the names to whatever you want.

6 Optionally, add other activities as needed. See Figure 22.

Figure 22 B2B Business Protocol With Outbound and Inbound Activities



Depending on the complexity of interaction between the enterprise and the trading partner, a B2B business protocol might have only one activity, or five fromInternal-toPartner activities followed by one fromPartner-toInternal reply, or an interleaved series of to-and-fro exchanges. The characteristic quality of a B2B business protocol is that each activity passes through the B2B host; there is no separate back-and-forth interactive communication between just the B2B host and either side.

7 Click **compile** (button in the upper left; the icon suggests a gear with a checkmark).

Note: *You cannot compile a B2B business protocol that does not reference a valid enveloping protocol; see steps 1 through 3 in this procedure.*

Once the B2B business protocol is successfully compiled, the project tree treats it as a container of the activities you modeled in the Dialog Designer. Although these activities can be referenced, they are not web service operations, and are therefore ineligible for dragging onto canvases. These are also the activities that appear at the lowest level of the tree in eXchange Partner Manager.

4.3 Setting Up B2B Hosts

A B2B Host is, in brief, both a logical component that oversees delivery channels and also an environment component that is exposed to web GUIs such as eXchange Partner Management (ePM) and message tracking. It provides bindings for Channel Managers.

The section explains how to create, populate, configure, and predeploy a B2B Host. These steps must be followed before Channel Manager can be activated, and before eXchange Partner Management (ePM) can be used for trading partner configuration.

The editor used for setting up B2B hosts is the **B2B Host Designer**.

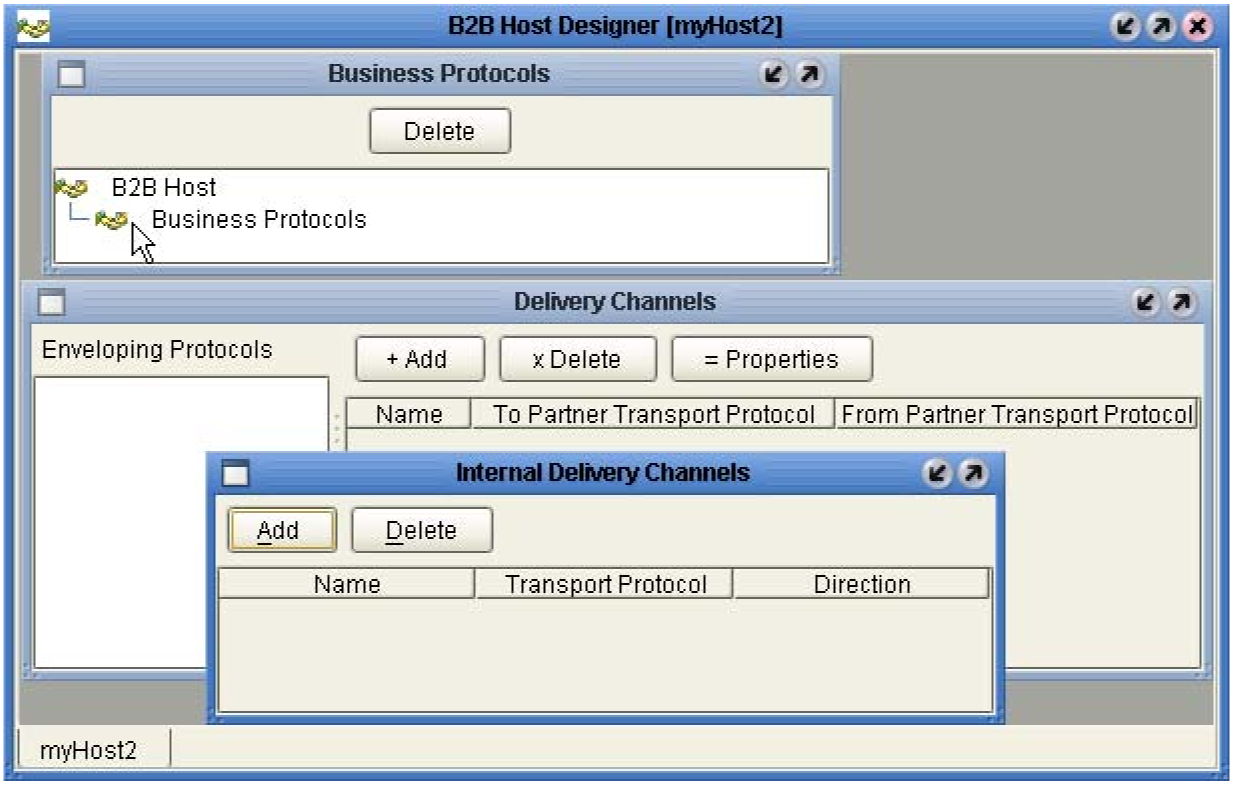
To create a B2B Host

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. On the popup context menu, point at **New**, and then click **B2B Host**.
3. In the **Create B2B Host** dialog, enter a name for the host, and then click **OK**.

   The project tree displays the new component, whose icon suggests two partners shaking hands.

   The B2B Host Designer opens to display three blank windows: Business Protocols, Delivery Channels, and Internal Delivery Channels. See Figure 23.

Figure 23 B2B Host Designer



- The **Business Protocols** window maintains a tree of Business Protocols that are exposed to the B2B host. These protocols are not added directly to the **Business Protocols** branch of the tree, but are instead organized by enveloping protocol.
- The **Delivery Channels** window lists these enveloping protocols in its left pane; the right pane lists delivery channels defined for the highlighted enveloping protocol, and allows you to add, delete, and modify the properties of those delivery channels.
- The **Internal Delivery Channels** window lists names, transport protocols, and direction (Sender=ToInternal; Receiver=FromInternal), and allows you to add and delete IDCs.

To add B2B business protocols to a B2B host, organized by enveloping protocol

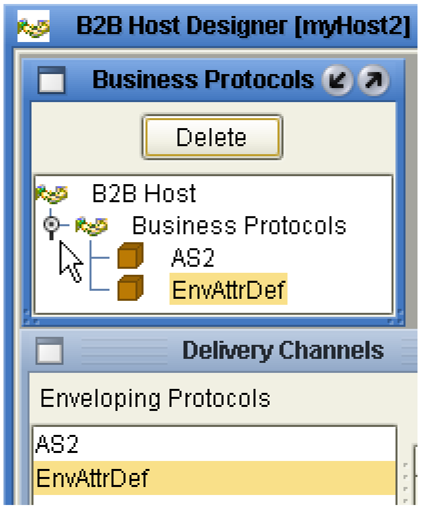
Before you begin: The B2B Host Designer must be open and the project tree available.

1 Into the **Business Protocols** window, under the Business Protocols branch, drag in an enveloping protocol (either custom or standard) from the project tree.

Note: *Standard enveloping protocols are located under the* ***SeeBeyond*** *project folder; each resides at the bottom of its corresponding eXchange >* ***MessagingProtocols*** *folder.*

The enveloping protocol appears both in the business protocols tree and in the Enveloping Protocols pane of the Delivery Channels window. See Figure 24.

Figure 24 Business Protocols Tree Organized by Enveloping Protocol

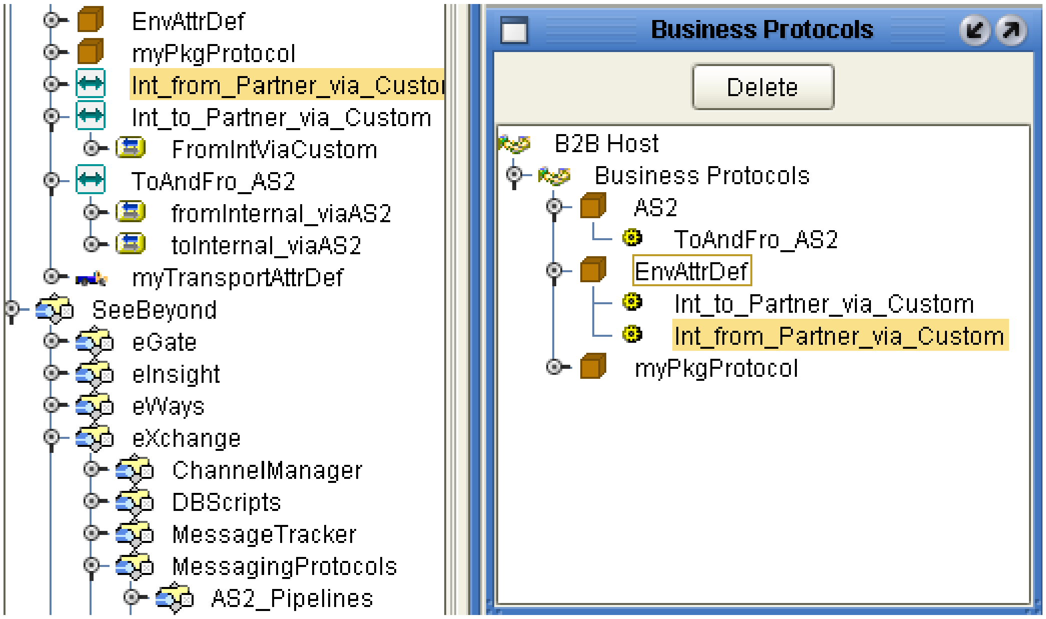


2 Under this enveloping protocol, drag in a B2B business protocol whose properties reference it. (The GUI does not allow you to drag in a B2B business protocol that references a different enveloping protocol, or no protocol.)

The B2B business protocol appears in the business protocols tree, subordinate to the enveloping protocol it references.

3 Repeat the previous step as needed, or repeat all the previous steps. See Figure 25.

Figure 25 Business Protocols Tree Populated with Standard and Custom Protocols



EnvAttrDef
myPkgProtocol
Int_from_Partner_via_Custo
Int_to_Partner_via_Custom
FromIntViaCustom
ToAndFro_AS2
fromInternal_viaAS2
toInternal_viaAS2
myTransportAttrDef
SeeBeyond
eGate
eInsight
eWays
eXchange
ChannelManager
DBScripts
MessageTracker
MessagingProtocols
AS2_Pipelines

Business Protocols

Delete

B2B Host
Business Protocols
AS2
ToAndFro_AS2
EnvAttrDef
Int_to_Partner_via_Custom
Int_from_Partner_via_Custom
myPkgProtocol

4 If you need to delete a leaf or a branch from the tree, highlight it and click **Delete**.

To add external delivery channels to a B2B host, organized by enveloping protocol

Before you begin: The Delivery Channels window must already display at least one item in its Enveloping Protocols pane; if it does not, refer to the previous procedure.

1 In the **Delivery Channels** window, in the Enveloping Protocols pane, click the enveloping protocol for which you want to add an external delivery channel.

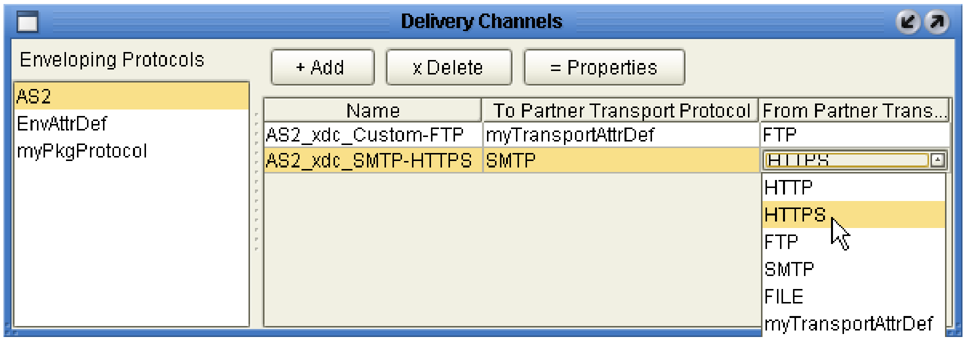
2 Click **Add**. When the new external delivery channel appears on the right, rename it.

3 In the **ToPartner Transport Protocol** list, choose from the list of standard and custom transport protocols displayed. This specifies transport protocol attributes for messages outbound from the B2B host to the trading partner on this channel.

4 In the **FromPartner Transport Protocol** list, choose from the list of standard and custom transport protocols displayed. This specifies transport protocol attributes for messages inbound on this channel from trading partner to B2B host.

5 Repeat the last three steps as needed, or repeat all previous steps. See Figure 26.

Figure 26 External Delivery Channels Populated and Configured



Delivery Channels

Enveloping Protocols
AS2
EnvAttrDef
myPkgProtocol

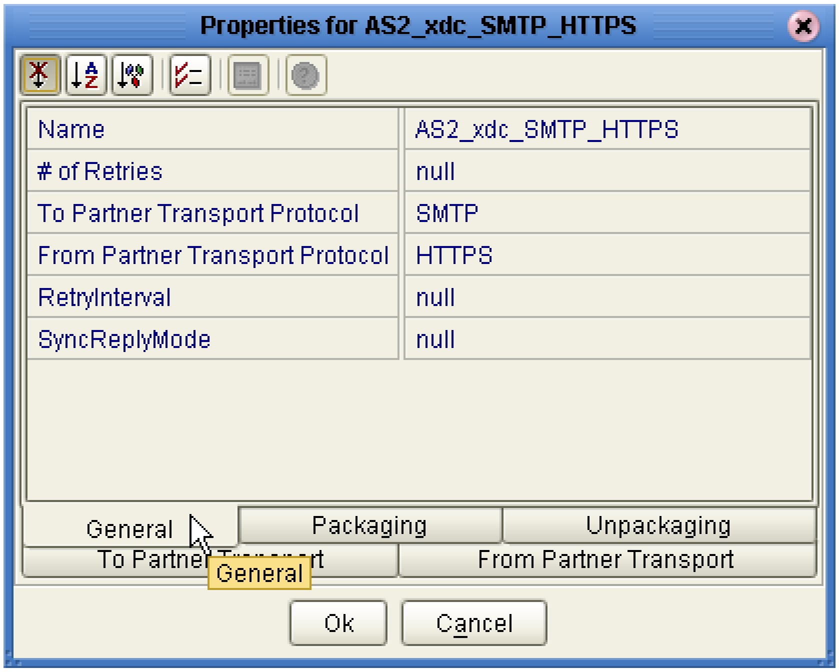
+ Add | x Delete | = Properties

Name	To Partner Transport Protocol	From Partner Trans...
AS2_xdc_Custom-FTP	myTransportAttrDef	FTP
AS2_xdc_SMTP-HTTPS	SMTP	HTTPS

HTTP
HTTPS
FTP
SMTP
FILE
myTransportAttrDef

6 For any external delivery channel for which you want to override default properties, click the channel to highlight it, click the **Properties** button to open its property sheet, and navigate to the property or properties you want to modify. See Figure 27 and Table 4 through **Figure 30 on page 47**.

Figure 27 Default Property Settings for External Delivery Channels — General



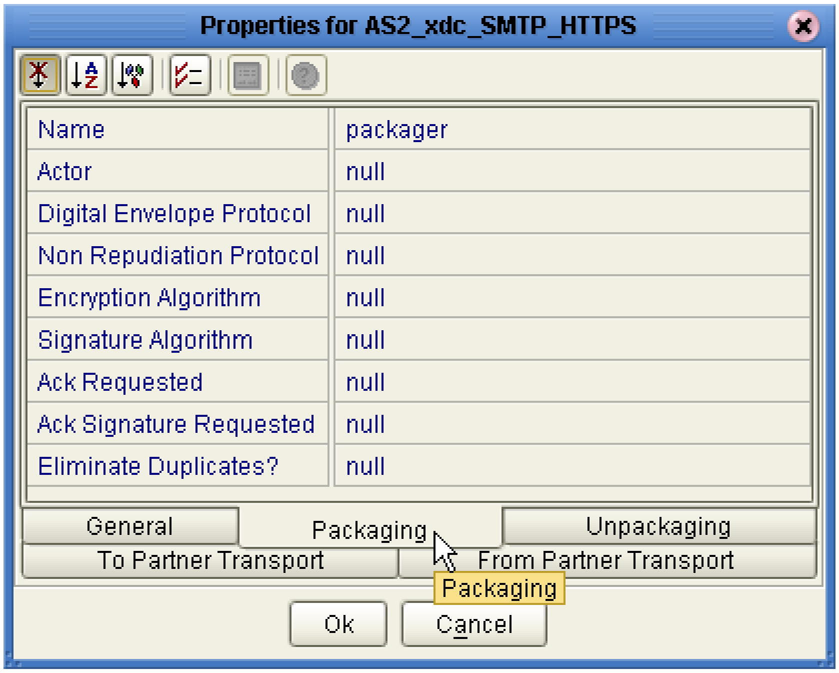
Properties for AS2_xdc_SMTP_HTTPS

Name	AS2_xdc_SMTP_HTTPS
# of Retries	null
To Partner Transport Protocol	SMTP
From Partner Transport Protocol	HTTPS
RetryInterval	null
SyncReplyMode	null

Table 4 Properties for External Delivery Channels — General

Property Name	Initial or Default Value	Description
Name	(component name)	Free text; this is the component name displayed in the project tree.
# of retries	**null**	Number of times to retry a send of the message.
To Partner Transport Protocol	(as specified in setup)	Transport protocol to be used when sending messages outbound to the trading partner.
From Partner Transport Protocol	(as specified in setup)	Transport protocol to be used when receiving messages inbound from the trading partner.
Protocol Attribute Definitions	**null** (initially)	An XML string that holds the values displayed in the **Protocol Attribute Definitions** dialog.
Retry interval	**null** (initially)	Interval to use for retrying a message send.
SyncReplyMode	**null** (initially)	The mode (synchronous or asynchronous) to be used for messaging.

Figure 28 Default Property Settings for External Delivery Channels — Packaging

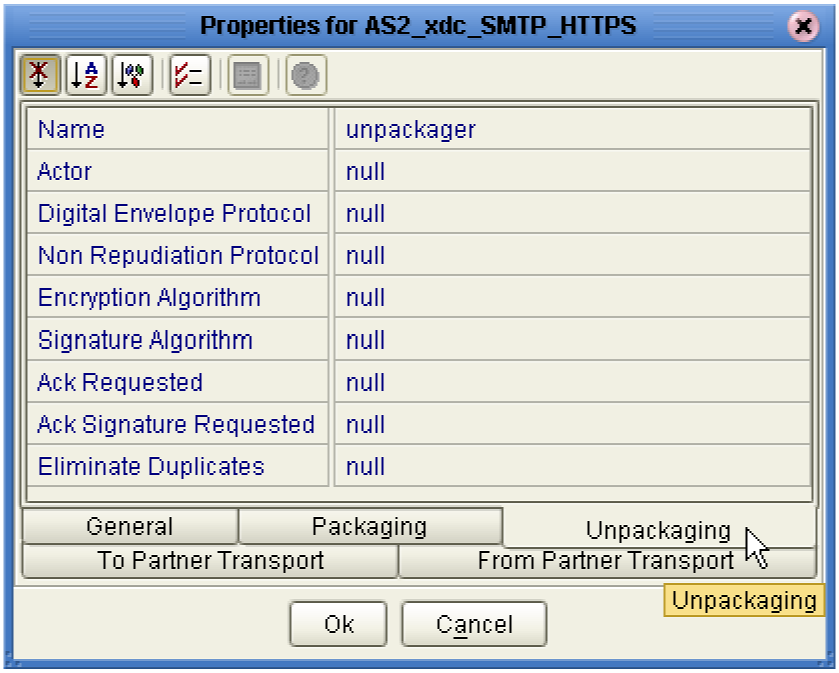


Name	packager
Actor	null
Digital Envelope Protocol	null
Non Repudiation Protocol	null
Encryption Algorithm	null
Signature Algorithm	null
Ack Requested	null
Ack Signature Requested	null
Eliminate Duplicates?	null

Table 5 Properties for External Delivery Channels — Packaging/Unpackaging

Property Name	Initial or Default Value	Description
Name	(component name)	Free text; this is the component name displayed in the project tree.
Actor	**null** (initially)	Namespace identifying the actor, such as urn:oasis:names:tc:ebxml-msg:actor:nextMSH
Digital Envelope Protocol	**null** (initially)	Digital envelope protocol and version to be used, of: **SMIME/2.0**, **SMIME/3.0**, or **DSIG/1.0**.
Nonrepudiation Protocol	**null** (initially)	Nonrepudiation protocol and version to be used, of: **SMIME/2.0**, **SMIME/3.0**, or **DSIG/1.0**.
Encryption Algorithm	**null** (initially)	Encryption algorithm to be used, of: **RC2** or **DES3**.
Signature Algorithm	**null** (initially)	Encryption algorithm to be used. If set, must be **xmldsig#dsa-sha1**.
Ack Requested	**null** (initially)	Specifies circumstances in which a message acknowledgment is requested (such as TA1), of: **Always**, **Never**, or **Per Message**.
Ack Signature Requested	**null** (initially)	Specifies circumstances in which a signature is required on the message acknowledgment, of: **Always**, **Never**, or **Per Message**.
Eliminate Duplicates?	**null** (initially)	Specifies circumstances in which duplicate messages are to be eliminated, of: **Always**, **Never**, or **Per Message**.

Figure 29 Default Property Settings for External Delivery Channels — Unpackaging

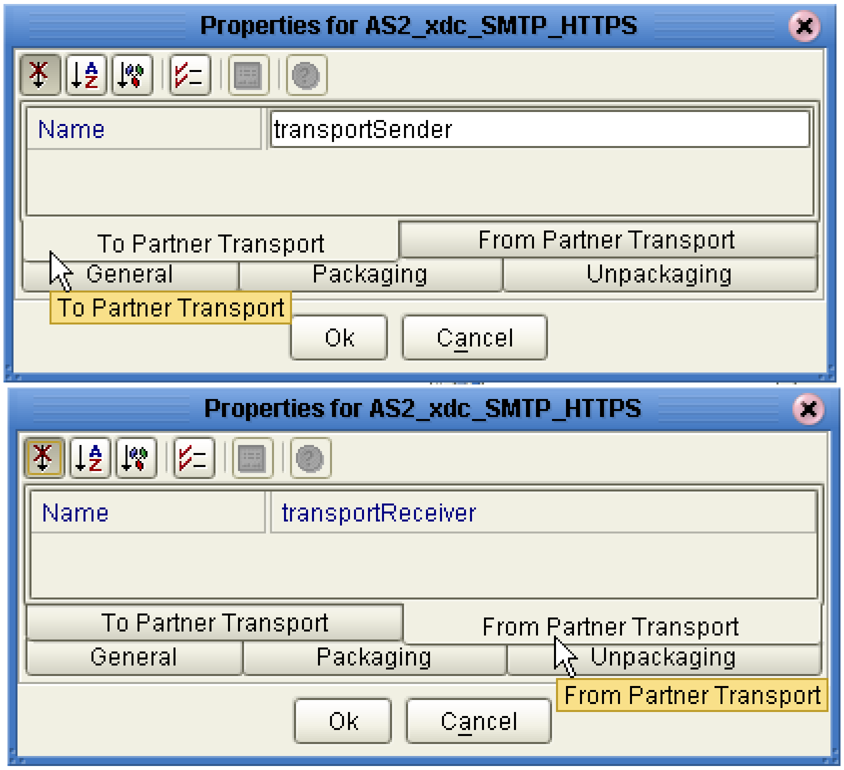


Properties for AS2_xdc_SMTP_HTTPS

Name	unpackager
Actor	null
Digital Envelope Protocol	null
Non Repudiation Protocol	null
Encryption Algorithm	null
Signature Algorithm	null
Ack Requested	null
Ack Signature Requested	null
Eliminate Duplicates	null

For unpackaging properties, names, values, and descriptions are the same as for packaging properties; see **Table 5 on page 46**.

Figure 30 Default Property Settings for External Delivery Channels—Partner Transport



Properties for AS2_xdc_SMTP_HTTPS

Name	transportSender

Properties for AS2_xdc_SMTP_HTTPS

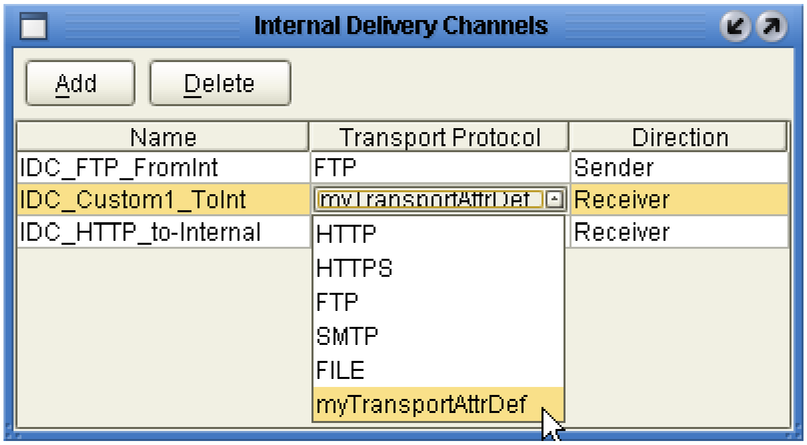
Name	transportReceiver

To add internal delivery channels to a B2B host

1. In the B2B Host Designer, in the **Internal Delivery Channels** window, click **Add**.
2. Rename the new internal delivery channel to something meaningful.
3. In the **Partner Transport Protocol** list, choose from the list of standard and custom transport protocols displayed.
4. In the **Direction** list, choose a designation of the role played by the Internal system:
   - **Sender** designates this internal delivery channel as coming from Internal.
   - **Receiver** designates this internal delivery channel as going to Internal.
5. As needed, repeat previous steps to add and configure internal delivery channels.

See Figure 31.

Figure 31 Internal Delivery Channels Populated and Configured



Name	Transport Protocol	Direction
IDC_FTP_FromInt	FTP	Sender
IDC_Custom1_ToInt	myTransportAttrDef	Receiver
IDC_HTTP_to-Internal		Receiver

4.4 Activating 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. As a server, it hosts Channel Managers and is exposed to the eXchange Partner Management GUI.

Unlike most other environment components, it is not offered as a template in the environment to be statically configured. Instead, since it has a dynamic dependency on the Oracle database that persists eXchange trading partner information (and message tracking information, if used), it must be predeployed to the environment. This section provides the steps required for predeployment: creating an environment, creating a map containing a B2B host, and activating a deployment profile that binds the map's components to the environment.

4.4.1. Creating an Environment

These steps create the minimal environment required to activate a B2B host.

To create and populate the environment prior to B2B host deployment

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

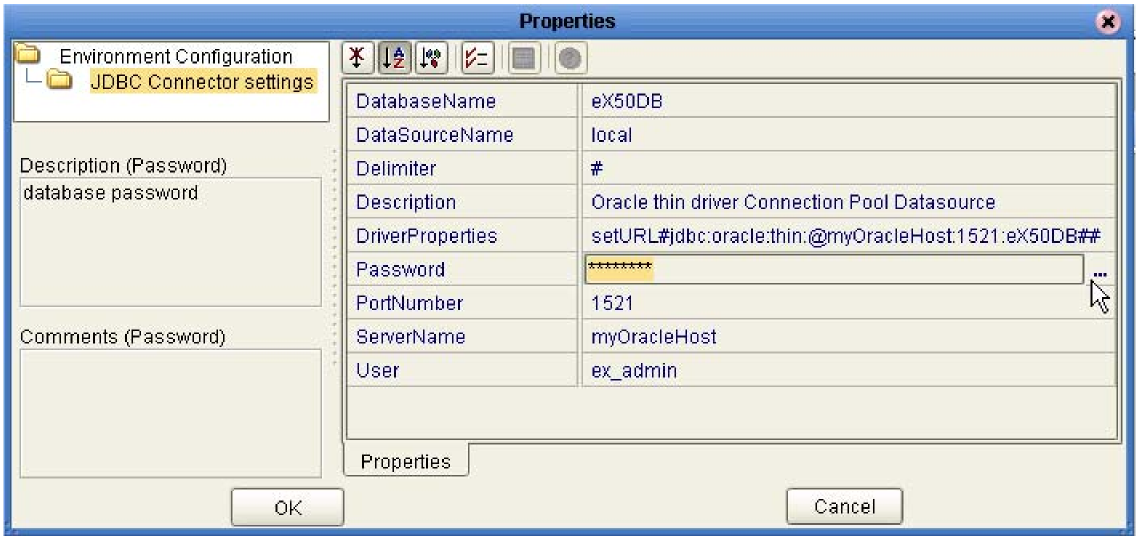
   The explorer tree displays a new environment, and the Environment Editor opens. Optionally, you can rename the environment to something meaningful.

2. In the environment explorer tree, right-click the new environment and, on the popup context menu, click **New Oracle External System**.

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

4. Configure the Oracle external with the values for your eXchange database instance. For details on configuring the Oracle external, see the Chapter 3 of the *Oracle eWay User's Guide*. For sample settings typical of an eXchange database, see Figure 32.

Figure 32 Environment Configuration for Oracle External System



Description (Password)
database password

Comments (Password)

DatabaseName	eX50DB
DataSourceName	local
Delimiter	#
Description	Oracle thin driver Connection Pool Datasource
DriverProperties	setURL#jdbc:oracle:thin:@myOracleHost:1521:eX50DB##
Password	********
PortNumber	1521
ServerName	myOracleHost
User	ex_admin

5. In the environment explorer tree, right-click the new environment again and, on the popup context menu, click **New 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.

6. In the environment explorer tree, right-click the new logical host and, on the popup context menu, click **New 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. For example, see **"Configuring the eInsight Run-Time Engine" on page 30** for instructions on configuring eInsight engine settings for database connections and persistence.

Once you finish these last steps, the environment now has all you need to deploy a B2B host: An integration server and an Oracle external system. However, you still need to create a map that links the B2B host to an Oracle eWay.

4.4.2. Creating a Map with a B2B Host and a MessageTracker

These steps create a map that allows you to activate a B2B host with a Message Tracker.

To create and populate the map prior to B2B host deployment

Before you begin: Your project must already contain a B2B host with at least one validly configured external delivery channel; see **Setting Up B2B Hosts** on page 42.

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

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

2 In the toolbar along the top of the canvas, click the **External Applications** tool and, from the dropdown list, select the checkbox for **Oracle External Application**.

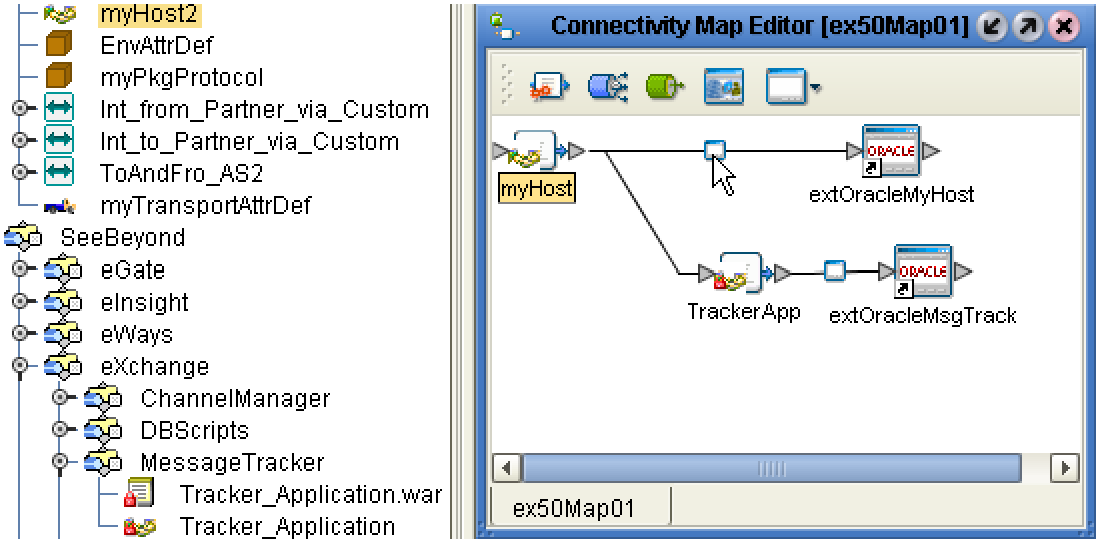
3 Drag two Oracle externals onto the canvas, towards the right side.

4 From the project tree, drag your B2B host onto the upper left side of the canvas.

5 In the project tree, open the **SeeBeyond > eXchange >MessageTracker** folder and drag its **Tracker_Application** protocol pipeline onto the lower center of the canvas.

6 Rename components to something meaningful, and connect as shown in Figure 33.

Figure 33 Map Showing B2B Host and Message Tracker to be Deployed



7 Configure both eWays, designating them outbound to external Oracle applications.

Now that the map is populated and configured, the B2B host is ready to be activated.

4.4.3. Activating the B2B Host

These steps create a deployment profile—a binding of particular logical components to an environment—and then activates it. Project activation always makes components available to external servers (such as the SeeBeyond integration server and Oracle), but activation of a B2B host makes it available as an external server. Activation also exposes the B2B host and MessageTracker application so they can communicate with the web-based eXchange GUIs, eXchange Partner Management and Message Tracking.

To create, configure, and activate a deployment profile

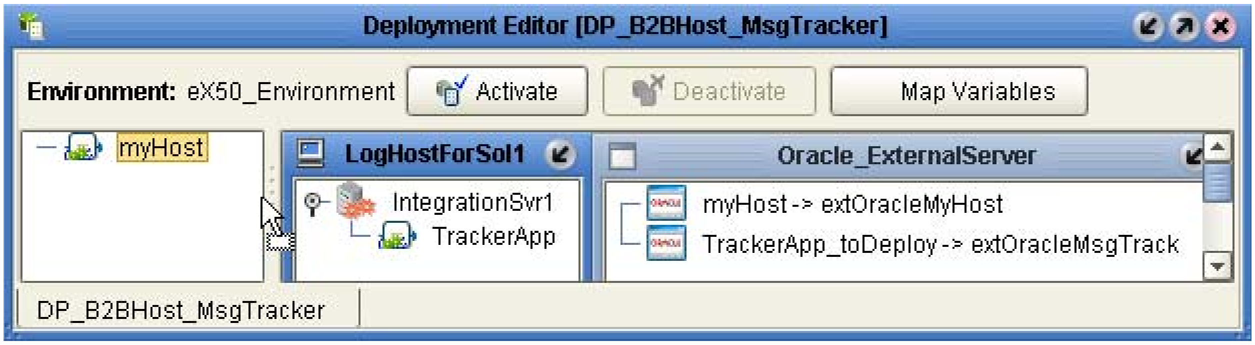
Before you begin: Your project must already contain a validly configured B2B host used in a map that connects it, via an outbound eWay, with a configured Oracle external.

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

The project tree displays a new deployment, and the Deployment Editor opens, showing a canvas whose left pane shows all components on the map and whose right pane shows all external servers in the environment.

2 One by one, drag both Oracle eWays to the Oracle server. Then, one by one, drag the B2B host and the tracker application to the integration server. See Figure 34.

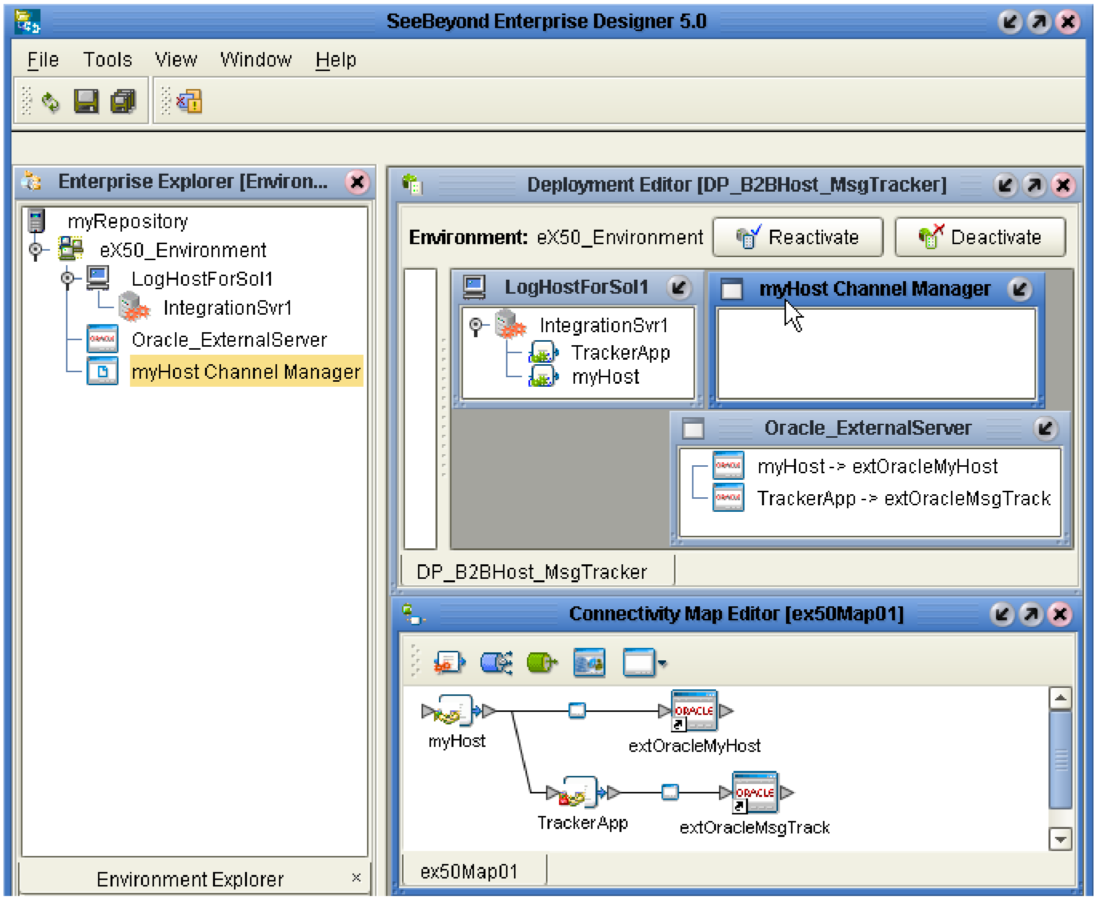
Figure 34 Deployment Profile Showing B2B Host Being Assigned to Integration Server


Deployment Editor [DP_B2BHost_MsgTracker]
Environment: eX50_Environment
Activate
Deactivate
Map Variables
myHost
LogHostForSol1
IntegrationSvr1
TrackerApp
Oracle_ExternalServer
myHost -> extOracleMyHost
TrackerApp_toDeploy -> extOracleMsgTrack
DP_B2BHost_MsgTracker

3 After all components have been assigned to external servers, click **Activate**.

Upon successful activation, the Environment Editor and Deployment Editor are updated to show a new B2B host server. See Figure 35.

Figure 35 Activated B2B Host in Environment and Deployment Profile



Once deployed, the B2B host server is capable of hosting Channel Managers; it is also available to Enterprise Manager facilities that perform pipeline monitoring, trading partner management, and message tracking.

To associate certificate and keystore information with a B2B host's delivery channels

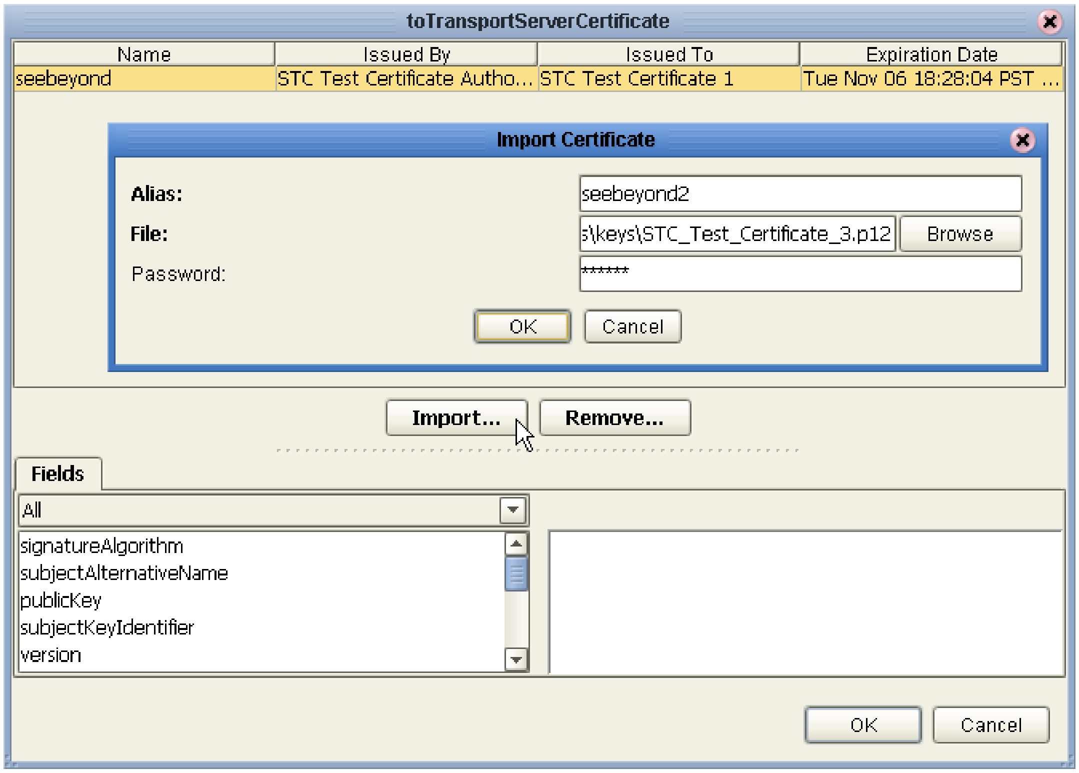
Before you begin: You must have already installed SMEWebServices.jar (see step 9 in the **procedure on page 24**), and you must have access to the locations and names of the necessary certificates and keystores.

1. In Enterprise Designer with the **Environment Explorer** tab active, in the tree, right-click the B2B host Channel Manager and, on the popup menu, click **Properties**.

   The properties page opens, showing all external delivery channels for this host.

2. One by one, as needed, click the ellipsis **[...]** for the type of certificate or truststore you want to define for a particular external delivery channel and then either select from the dropdown list or else use Import to import a new certificate or truststore. For certificate import, see Figure 36; for truststore import, see Figure 37.

Figure 36 Importing a Certificate for a Delivery Channel in a B2B Host Environment



toTransportServerCertificate

Name	Issued By	Issued To	Expiration Date
seebeyond	STC Test Certificate Autho...	STC Test Certificate 1	Tue Nov 06 18:28:04 PST ...

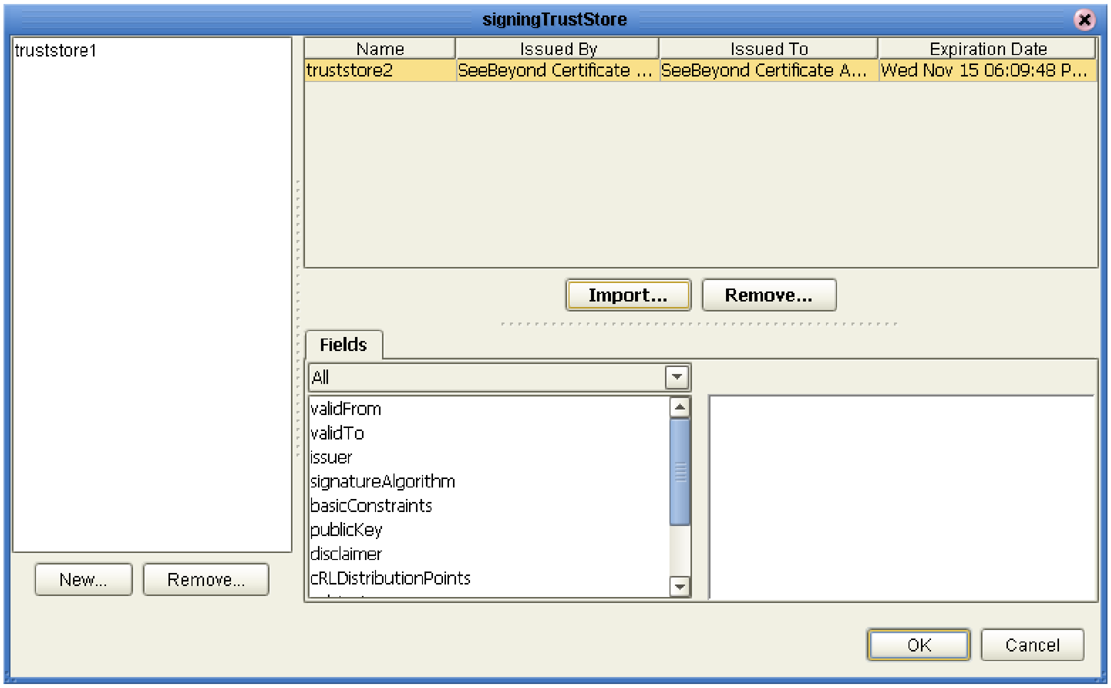
Import Certificate

Alias: seebeyond2

File: s\keys\STC_Test_Certificate_3.p12

Password: ******

Figure 37 Importing a TrustStore for a Delivery Channel in a B2B Host Environment



signingTrustStore

truststore1

Name	Issued By	Issued To	Expiration Date
truststore2	SeeBeyond Certificate ...	SeeBeyond Certificate A...	Wed Nov 15 06:09:48 P...

Chapter 5

Designing B2B Protocol Pipelines

You can use eXchange to configure the components depicted by each activity in your B2B protocol pipelines. This chapter provides the background information you need to create and understand eXchange B2B protocol pipelines.

5.1 Overview

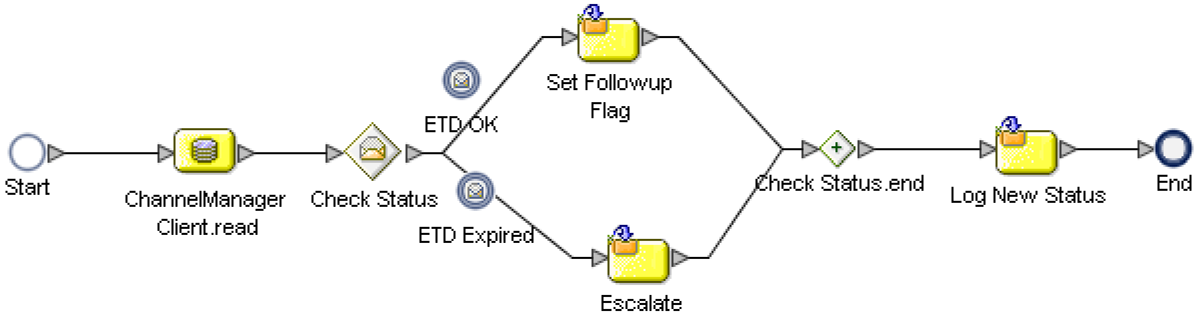
Topics in this chapter are:

5.2 Building a B2B Protocol Pipeline

A business process is a collection of actions that take place in your company, revolving around a specific business practice. These processes can involve a variety of participants and may include internal and external computer systems or employees. In eXchange, you create a graphical representation of the business process called a *B2B protocol pipeline*.

A business process modeled in eXchange may look something like Figure 38.

Figure 38 Sample B2B Protocol Pipeline



Add a B2B Protocol Pipeline to your Project

Adding a B2B protocol pipeline to your project provides an empty modeling canvas where you add and manipulate items on the canvas, called *activities*. Before you can model your business process, you must add a new protocol pipeline to your project.

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

1. In Enterprise Designer, in Project Explorer, right-click the project and, on the popup context ment, point at **New** and click **B2B Protocol Pipeline**.
2. Enter a new name for your B2B protocol pipeline.

5.2.1. Modeling a Business Process in eXchange

To model a business process in eXchange, drag and drop modeling elements on the Exchange Protocol Designer, and then link these components to reflect the logical flow of the business process. eXchange provides the tools you need to quickly develop B2B protocol pipeline models, including graphical editing tools to help you adjust, size, and align model components.

eXchange Protocol Designer

Once you create a new protocol pipeline, you will build your model in the eXchange Protocol Designer (as shown in Figure 39). The eXchange Protocol Designer is the area in the Enterprise Designer where you view, create, and edit your protocol pipelines.

Create a B2B Protocol Pipeline

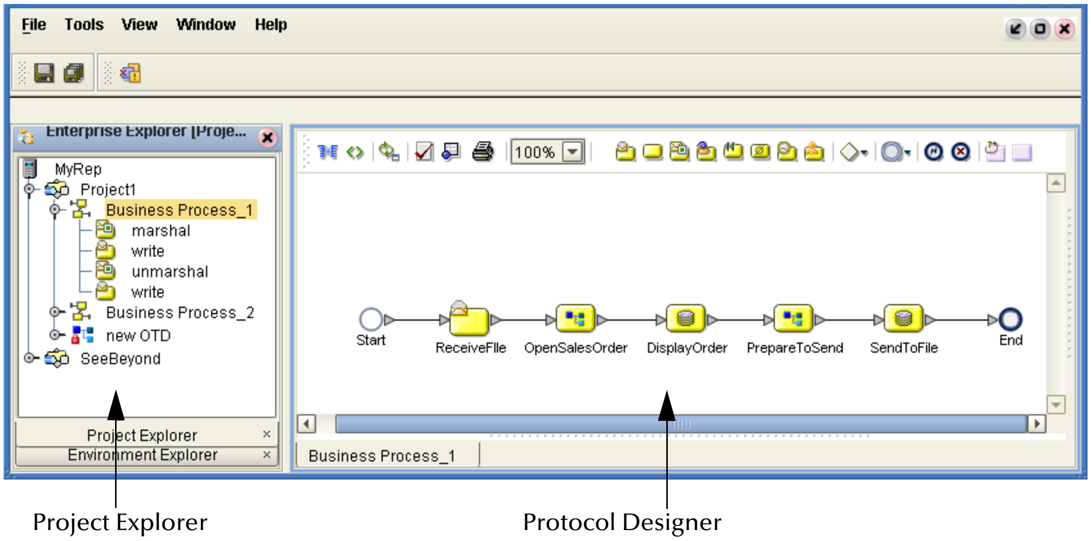
Begin designing your protocol pipeline by dragging and dropping modeling elements from the toolbar onto the Exchange Protocol Designer canvas.

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

1. Drag the appropriate modeling elements to your blank protocol pipeline to the Exchange Protocol Designer canvas. See Figure 39.

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

Figure 39 B2B Protocol Pipeline



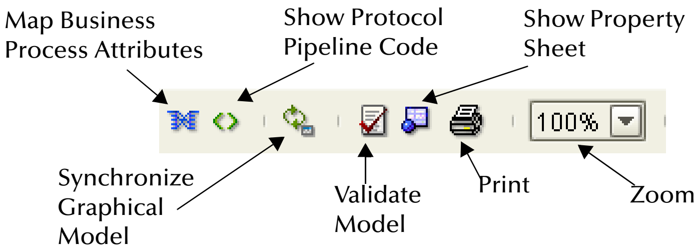
2 Draw links between the modeling elements to show the process flow (Figure 39)

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

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

5.3 Using the eXchange Protocol Designer GUI

Figure 40 Toolbar Options


Map Business Process Attributes
Show Protocol Pipeline Code
Show Property Sheet
100%
Synchronize Graphical Model
Validate Model
Print
Zoom

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

5.4 Modeling Elements

The eXchange Protocol Designer is where the user creates the protocol pipeline flow. It provides a palette of modeling elements for designing your B2B protocol pipeline. Like other logical components in a project, protocol pipelines 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 B2B protocol pipeline you create consists of basic elements as described in the following sections:

- **Activity Elements** on page 57
- **Branching Activities** on page 59
- **Intermediate Events** on page 60
- **Scope** on page 61
- **While** on page 61

Activity Elements

You can include several different kinds of activities and subprocesses in a B2B protocol pipeline. For examples of each of the different kinds of activities, see Table 6.

To add an activity

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

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

The activity appears on the modeling canvas.

Link modeling elements

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

To link modeling elements, do the following:

1. Move your cursor over the connector portion of your modeling element.
2. Hold the cursor over the outside edge of the modeling element until it changes from the arrow pointer to a hand (see Figure 41).

Figure 41 Selected Activity

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

Table 6 Activity Elements

Button	Command	Function
○▷ Start	Start Node	The Start Node is a modeling element indicating the start of the process. This element appears in the eXchange Protocol Designer Editor by default, when you create a new B2B protocol pipeline. A Start Node can only link to an activity that has a **receive** or **read** capability, signaled by a subicon in the upper left resembling an opened envelope (see **Receive Activity** just below).
▷——▷ ▷—✕—▷	Link Link with Business Rule	Links indicate the flow of the protocol pipeline 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 protocol pipeline attributes. eXchange ensures the model is being properly linked because it does not allow invalid links to connect. Links can also accept mapped values. A link with mapped values is highlighted in blue.
▷● End	End Node	The End modeling element indicates the completed state of a protocol pipeline. This element appears in the B2B Protocol Designer by default, when you create a new B2B protocol pipeline.
✉	Receive Activity	The Receive activity indicates the invocation of a protocol pipeline or a wait state pending the arrival of an inbound message. The Receive activity represents the actual method by which a B2B protocol pipeline is initiated. For example: ▪ An eWay polls a file system or database and retrieves data that is passed to the engine, along with the indication that a protocol pipeline instance has started. ▪ A user types a URL into their browser and a servlet initiates a protocol pipeline by sending a message to eGate or eInsight.
▭	Activity	An activity is a step in the protocol pipeline in which the engine will invoke a web service operation or an eGate component. Depending upon the configuration of the component, a response may or may not be required. One example would be a synchronous extraction process from a database to return the credit status of a trading partner.

Table 6 Activity Elements

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

Branching Activities

Branching activities are objects you add to your B2B protocol pipelines to specify the logical flow of information. eXchange provides three different kinds of branching activities—Decisions, Event Based Decisions, and Flow.

Add a Branching activity

To add a Branching activity to the modeling canvas:

1 On the toolbar, click the **Branching Activities** pulldown icon, and then release the mouse button.

2 Point at the type of Branching activity you want to add, click, and then drag the activity from the toolbar to the eXchange Protocol Designer canvas.

The selected Branching activity appears on the modeling canvas.

Table 7 Branching Activities

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

Intermediate Events

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

Add an Intermediate event

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

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

   The selected Intermediate event appears on the modeling canvas.

Table 8 Intermediate Events

⏪	Compensation Handler	Used when something in a protocol pipeline fails and requires a rollback or upstream activities (like money has to be returned to the customer). On an automatic basis in the protocol pipeline, upstream steps in the protocol pipeline 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 protocol pipeline. The components within that scope will throw the exceptions when things go wrong and the exception handler will automatically initiate the appropriate process to handle the problem.

Table 8 Intermediate Events

	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 pipeline. Each of these elements can be a different message.
	Timer Event	A timeout condition is set upon Activities, sets of Activities, or a protocol pipeline as a whole, to ensure that processes complete within given amount of time. Timeout conditions also allow you to design the pipeline 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 protocol pipeline.

While

	While	This allows you to create a looping process within a protocol pipeline (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 protocol pipeline continues.

5.4.1. Validating a B2B Protocol Pipeline

After creating a B2B protocol pipeline, 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 protocol pipeline for errors

- On the toolbar, click **Validate Protocol Pipeline 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 B2B Protocol Pipeline"** for information on repairing errors. Repairing the error may entail such items as adding logic to Decisions or adding attributes to activities.*

5.4.2. Saving an Unfinished B2B Protocol Pipeline

Even if a B2B protocol pipeline is not complete and/or contains errors, you can save it as a work in progess and return to it later.

To save a B2B protocol pipeline

- Do one of the following:
  - On the **File** menu, click **Save**
  - On the main toolbar, click **Save**.
  - On the keyboard, press **Ctrl+S**.

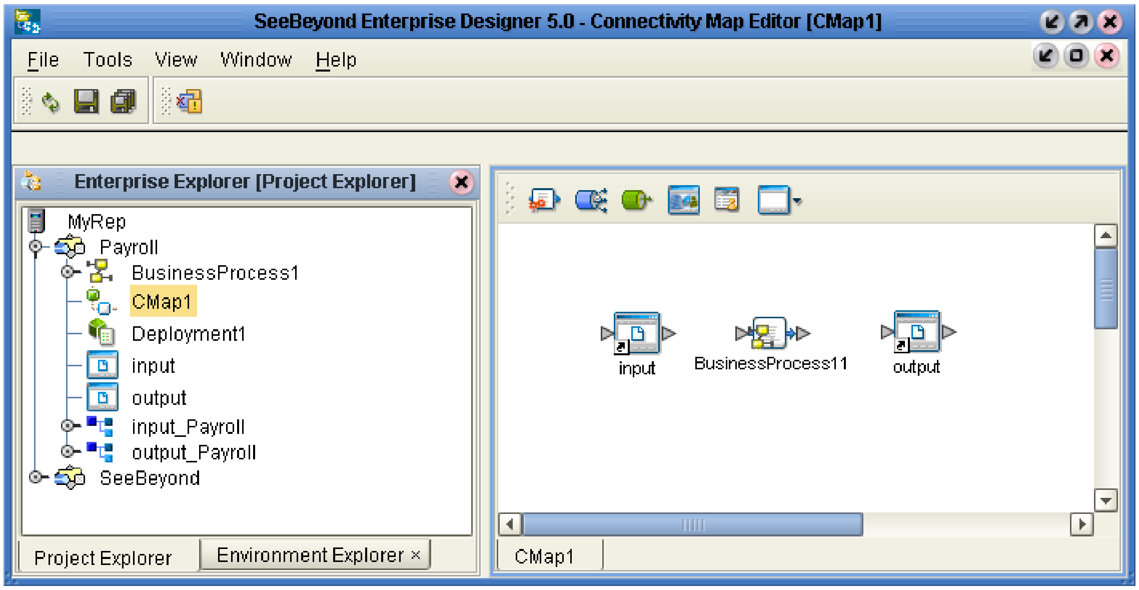
5.5 Using B2B Protocol Pipelines in a Connectivity Map

The connectivity map represents connection information in the ICAN Suite. The flow is represented at a higher level than in the B2B protocol pipeline. eXchange also uses the information in the connectivity map to establish and maintain connections to systems for the correct step in a protocol pipeline.

To include a protocol pipeline as a service on a connectivity map

1. In the Connectivity Map Editor, drag a protocol pipeline onto the canvas.
2. Add and connect other components and external systems as needed. See Figure 42.

Figure 42 Connectivity Map with B2B Protocol Pipeline

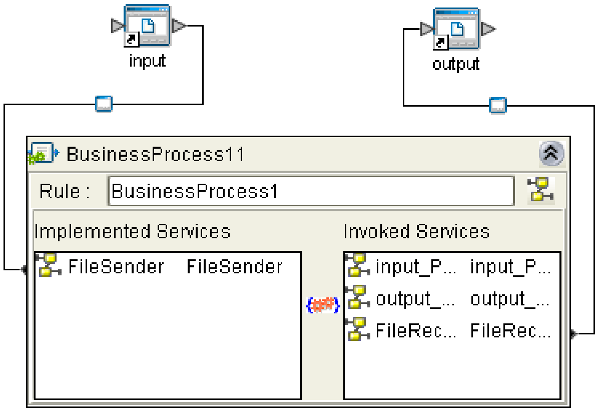


To connect the B2B Protocol Pipeline activities to the externals

1. In the map, double-click the protocol pipeline 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 43.

Figure 43 Connectivity Map: Protocol Pipeline Binding



5.6 Deploying a Project With a B2B Protocol Pipeline

5.6.1. Deployment Profiles Containing Protocol Pipelines

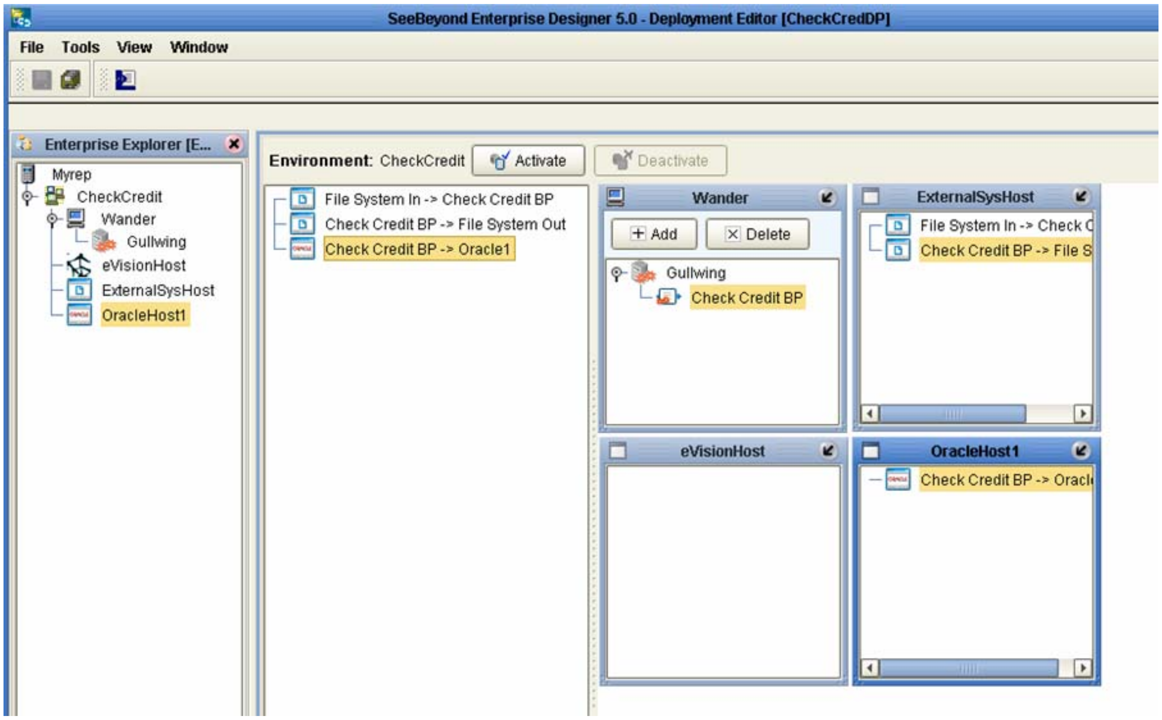
An environment becomes useful only after it has been populated with:

- One or more logical hosts, each containing one or more integration servers. Configuration steps may be required for integration servers, depending on the nature of the project and run-time environment.
- All necessary external hosts, properly configured.

Note: *If a B2B host is present in a pipeline or map, it requires special predeployment steps. See* **Activating a B2B Host** *on page 48.*

When these conditions are met, a deployment profile that references the environment is used to assign each logical component, eWay, and Channel Manager to a corresponding integration server and external host system. See Figure 44.

Figure 44 Deployment Profile with Some Components Assigned


SeeBeyond Enterprise Designer 5.0 - Deployment Editor [CheckCredDP]
File
Tools
View
Window
Enterprise Explorer [E...
Myrep
CheckCredit
Wander
Gullwing
eVisionHost
ExternalSysHost
OracleHost1
Environment: CheckCredit
Activate
Deactivate
File System In -> Check Credit BP
Check Credit BP -> File System Out
Check Credit BP -> Oracle1
Wander
Add
Delete
Gullwing
Check Credit BP
ExternalSysHost
File System In -> Check C
Check Credit BP -> File S
eVisionHost
OracleHost1
Check Credit BP -> Oracl

- Each B2B protocol pipeline and each B2B host component should be dragged under the correct integration server of the appropriate logical host.
- Each eWay should be dragged into the correct external host system.
- Each Channel Manager, if present, should be dragged into a predeployed B2B host.

After all logical components have been assigned, click **Activate** to generate the code that will be run by the integration server within the logical host engine.

Chapter 6

Exception Handling

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

- **“Scope” on page 66**
- **“Compensation” on page 67**

6.1 Overview

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

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

Exception handling in protocol pipelines 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.

Protocol pipelines 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.

6.1.1. Exception Handling Configuration

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

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

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

1. Build the exception handling logic as a protocol pipeline.
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 protocol pipeline that appropriately handles each exception.
6. Model manual exceptions in a protocol pipeline.
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 protocol pipeline 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 (i.e. time-outs or bad messages).

Failed components/systems create visual alerts via the protocol pipeline 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.

6.2 Scope

Scope allows you to define a range

- For handling of exceptions
- For creating compensation logic

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

Scope or Process level exceptions

Either the **Named** or **Catch All exceptions** can be used at the protocol pipeline level.

Named exception

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

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

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

Catch All Exceptions

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

6.3 Compensation

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

Using Scope and Exceptions to Trigger Compensation

- Compensation Activity—In an exception handler, initiates the compensation process. It models the compensation as a protocol pipeline, 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.

6.4 Validate the B2B Protocol Pipeline

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

Chapter 7

Using eXchange in Enterprise Manager

This chapter provides step-by-step procedures for using the Web-based facilities in Enterprise Manager to configure and monitor eXchange components and processes.

In this chapter

Each of these three facilities requires that you have already created a valid project and activated it.

7.1 eXchange Partner Management (ePM)

Before you begin: You must have already configured a B2B host with one or more delivery channels (see **procedure on page 42**, **"Setting Up B2B Hosts"**) and you must have already predeployed it (see **procedure on page 48**, **"Activating a B2B Host"**). If you want to use encryption in the trading partner, you must have already configured the proper certificates and truststores in the environment properties of the B2B host.

Overview

In this procedure, you will add a trading partner, bind it to at least one external delivery channel, add a profile beneath it, configure the profile, assign a business protocol to it, and configure all of the actions within the business protocol.

Accessing eXchange Partner Manager (ePM)

To access eXchange Partner Manager and locate a B2B host in the explorer tree

Before you begin: Your Repository server must already be running.

1. Open a browser window and point it at the hostname and port where your Repository server is running. For example:

   ```
   http://localhost:16271
   ```

2. Log in to Enterprise Manager.

3. When you have logged in, point your browser at a new URL that is the same as the previous one but with the string **/epm** appended. For example:

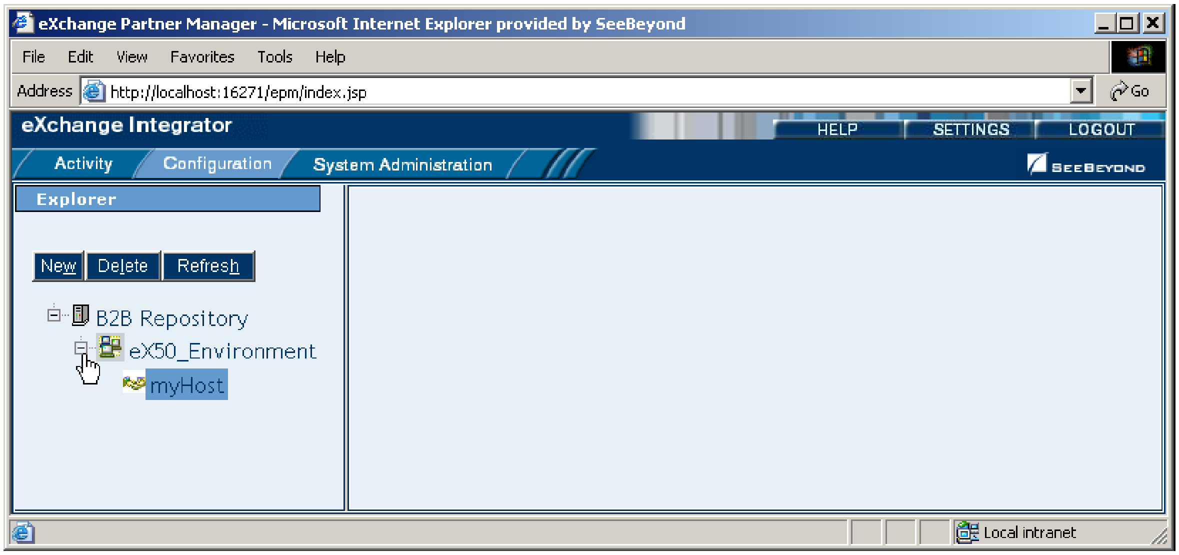
```
http://localhost:16271/epm
```

Note: *This URL is case-sensitive. If your browser warns you, "Requested resource not available," double-check that you entered the suffice* ***/epm*** *in all-lowercase.*

After a pause, the window displays the eXchange Partner Manager GUI, a two-pane window with an explorer tree on the left and a canvas on the right.

4 In the tree, open the root (named **B2B Repository**) and then open the environment where you deployed your B2B host. So far, the canvas remains blank. See Figure 45.

Figure 45 eXchange Partner Management (ePM) — No Configuration



Creating Trading Partners

To create and name a trading partner

1 With the B2B host highlighted, click **New** (top left button in Explorer pane).

2 Enter a name for the trading partner, and then click **Save**.

In the explorer tree, the new trading partner appears under the B2B host.

3 Click the trading partner and then click **New** (upper left).

The **Trading Partner Configuration** canvas displays two tabs: **Properties** and Components.

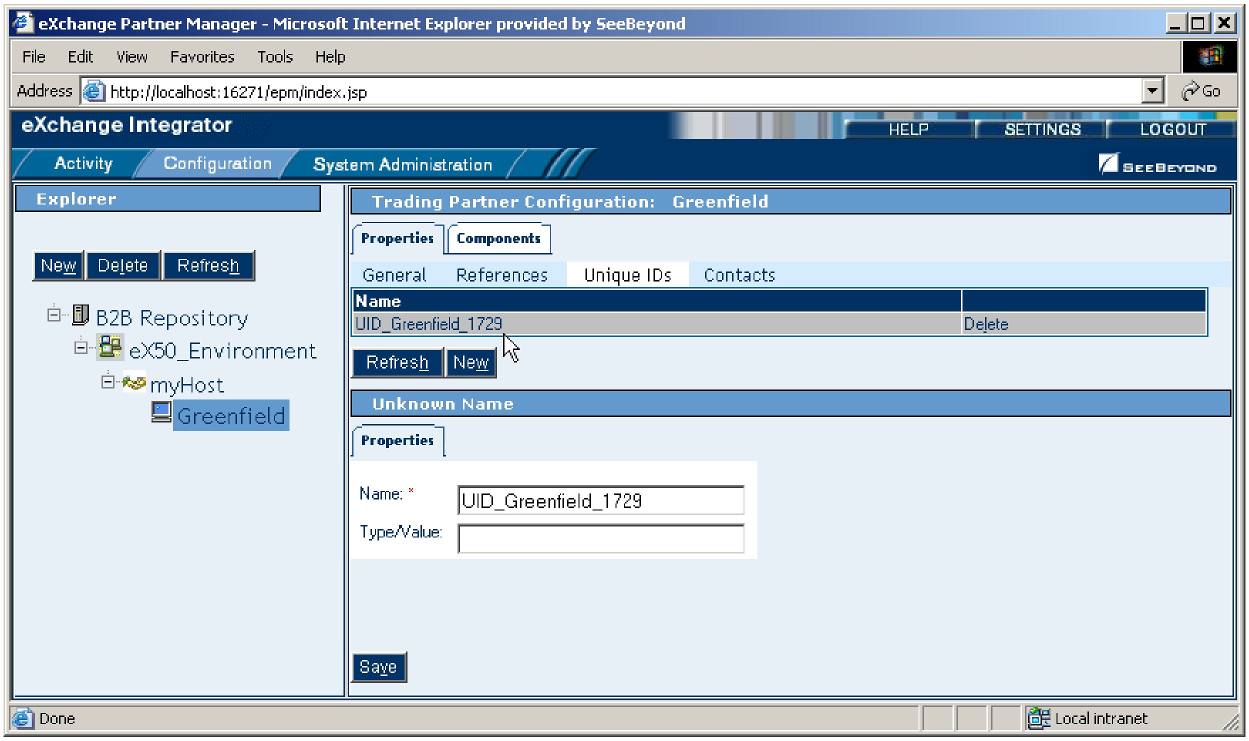
4 In the Properties tab with the **General** subtab active, click **Save**.

5 Click **Unique IDs** (the third subtab), and then click **New** (button in lower center).

6 Enter a unique ID for this trading partner, and then click Save.

The new unique ID appears under the Name column. See Figure 46.

Figure 46 New Trading Partner with Unique ID



eXchange Partner Manager - Microsoft Internet Explorer provided by SeeBeyond
File Edit View Favorites Tools Help
Address http://localhost:16271/epm/index.jsp
Go
eXchange Integrator
HELP SETTINGS LOGOUT
Activity Configuration System Administration
SeeBeyond
Explorer
New Delete Refresh
B2B Repository
eX50_Environment
myHost
Greenfield
Trading Partner Configuration: Greenfield
Properties Components
General References Unique IDs Contacts
Refresh New
Unknown Name
Properties
Save
Done
Local intranet

Name	
UID_Greenfield_1729	Delete

Name: * UID_Greenfield_1729

Type/Value:

Creating Bindings to External Delivery Channels

To access an external delivery channel and create a binding to it

1. With a trading partner highlighted, in the configuration canvas, click **Components**.

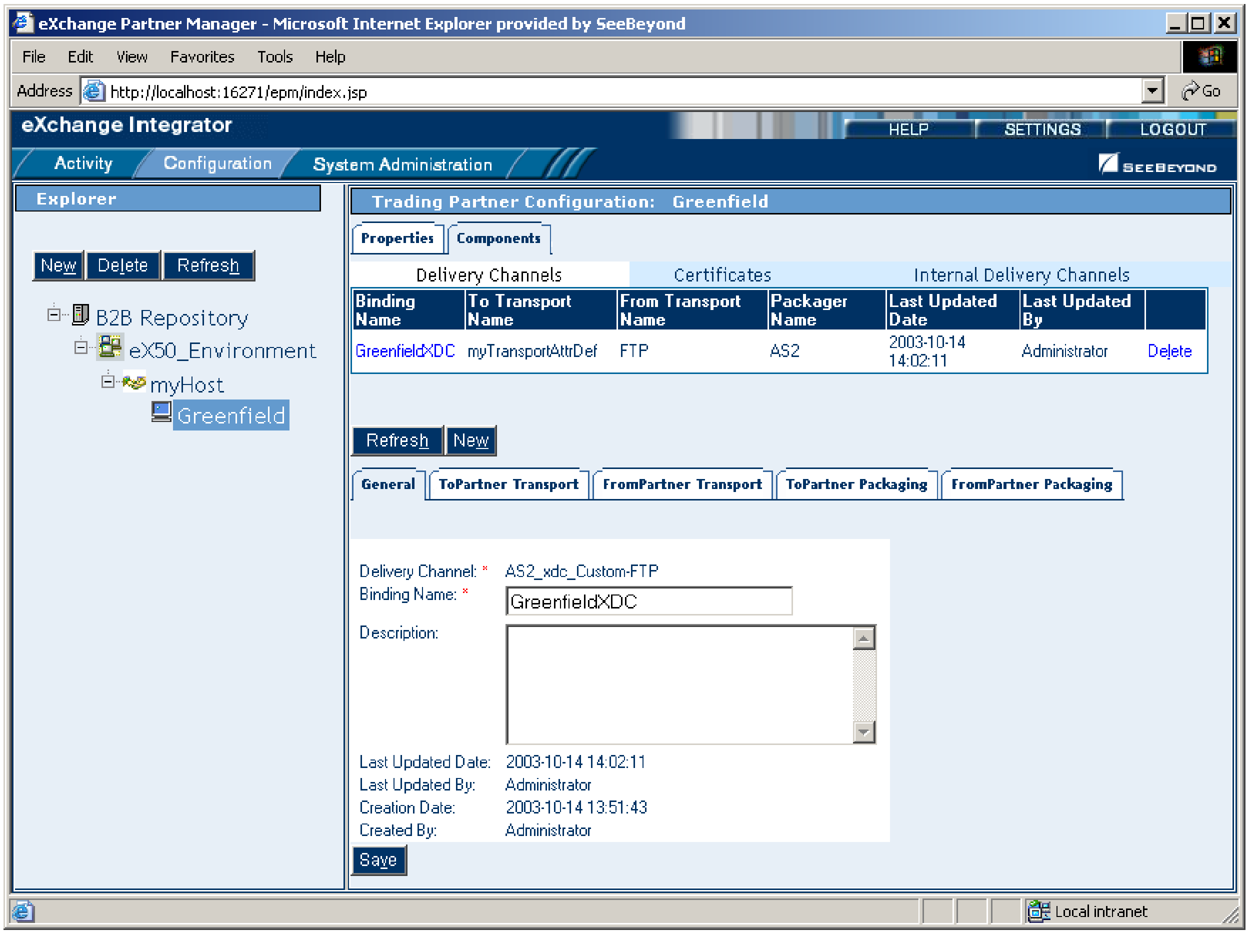
   When the Components tab becomes active, it displays three subtabs.

2. With **Delivery Channels** active (the first subtab), click **New** (lower center button).

3. Select a delivery channel from the list, enter a binding name, and click **Continue**.

   Five new subtabs are displayed.

Figure 47 Trading Partner Newly Associated with an External Delivery Channel



Binding Name	To Transport Name	From Transport Name	Packager Name	Last Updated Date	Last Updated By	
GreenfieldXDC	myTransportAttrDef	FTP	AS2	2003-10-14 14:02:11	Administrator	Delete

Delivery Channel: * AS2_xdc_Custom-FTP
Binding Name: * GreenfieldXDC
Description:
Last Updated Date: 2003-10-14 14:02:11
Last Updated By: Administrator
Creation Date: 2003-10-14 13:51:43
Created By: Administrator

You will need to provide specific parameter values to be used by the transport and enveloping (=packaging) protocols.

4 Click the **ToPartner** Transport tab and enter appropriate transport protocol values.

The number and type of parameters depends on the particular transport protocol you are using. For example, a File transport protocol might require only three values, an FTP transport protocol might require six and permit twelve, and a protocol that uses user-defined attributes can require hundreds of values, or none.

5 When you have entered all values, click **Save**.

6 Repeat the previous two steps for the other three tabs (FromPartner Transport, ToPartner Packaging, and FromPartner Packaging). Be sure to click **Save** each time.

7 Optionally, if you are using Secure Messaging Exchange (SME), you can import an Encryption Key in the ToPartner Packaging tab and import a Signature certificate in the FromPartner Packaging tab.

To import a signature certificate or encryption key

1 Do one of the following:

- In a Components Delivery Channels > **[ ]Partner Packaging** tab, click **Import**.

- In the Components >**Certificates** tab, click **New**.

2 Enter a value for **Certificate Name**., click **Browse**, locate and select the correct certificate, and then click **Open** to choose the file and return.

3 Click **Save**.

The imported certificate is saved. It is now displayed both under the **Certificates** tab and in the drop-down list.

Creating Bindings to Internal Delivery Channels

To access an internal delivery channel and create a binding to it

1 With a trading partner highlighted, in the configuration canvas, click **Components**.

When the Components tab becomes active, it displays three subtabs.

2 With **Internal Delivery Channels** active (rightmost tab), click **New** (lower center).

3 Select an IDC from the list, accept or enter a binding name, and click **Continue**.

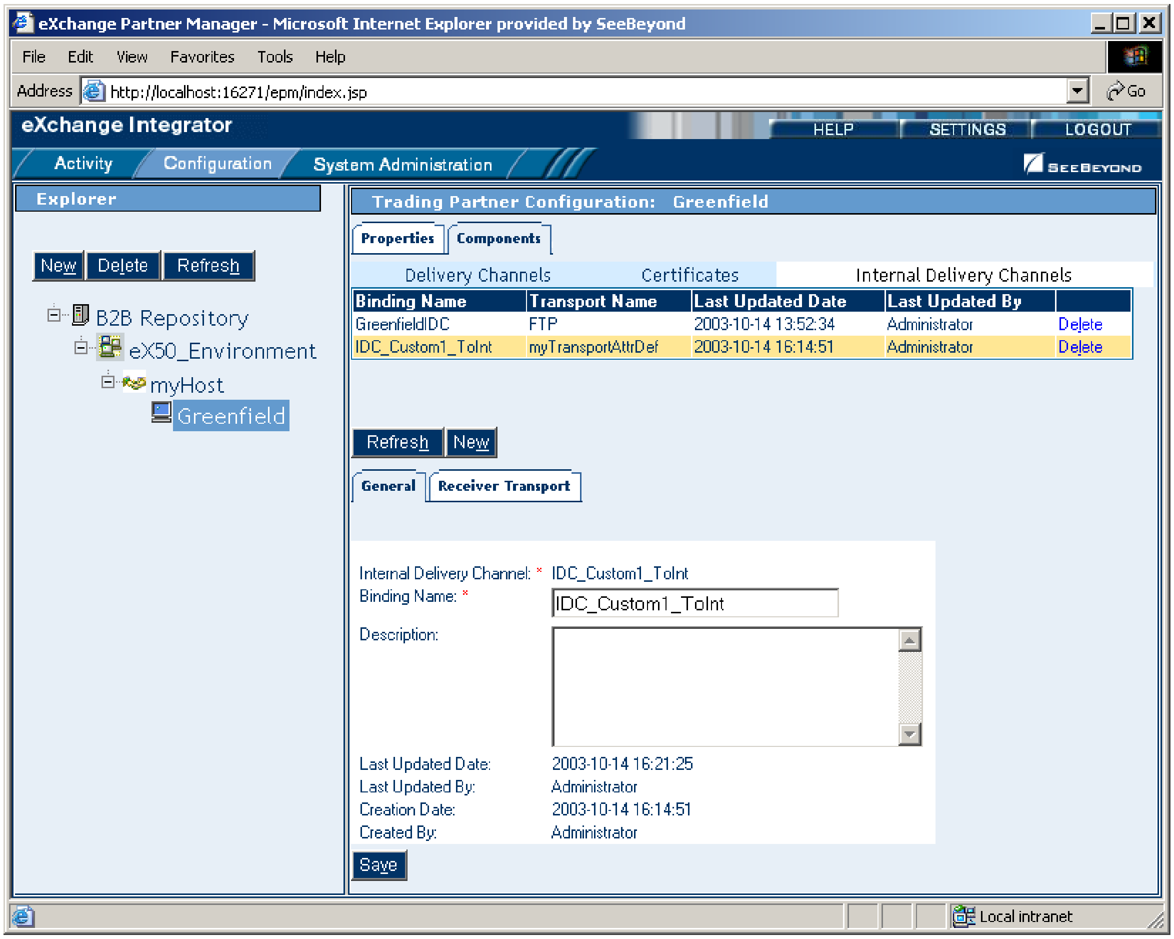
Two new subtabs are displayed—in addition to **General**, one or the other of:

- **Sender Transport** (for IDCs that the B2B host designated as From Internal)
- **Receiver Transport** (for IDCs that the B2B host designated as To Internal)

4 Click the Sender or Receiver tab and supply values for that transport protocol.

5 Click **Save**. See Figure 48.

Figure 48 Trading Partner Newly Associated with an External Delivery Channel



Trading Partner Configuration: Greenfield

Binding Name	Transport Name	Last Updated Date	Last Updated By	
GreenfieldIDC	FTP	2003-10-14 13:52:34	Administrator	Delete
IDC_Custom1_ToInt	myTransportAttrDef	2003-10-14 16:14:51	Administrator	Delete

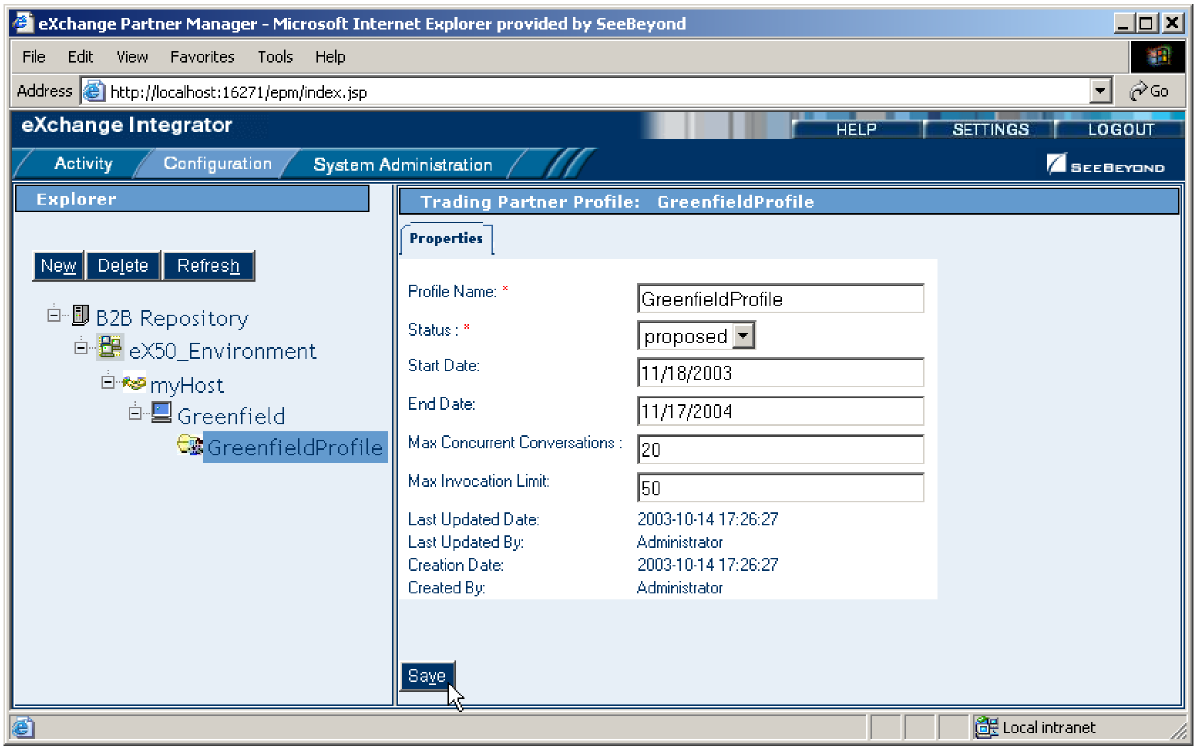
Internal Delivery Channel: * IDC_Custom1_ToInt
Binding Name: * IDC_Custom1_ToInt
Description:
Last Updated Date: 2003-10-14 16:21:25
Last Updated By: Administrator
Creation Date: 2003-10-14 16:14:51
Created By: Administrator

Creating Profiles and Activating Trading Partners

To create a trading partner profile and activate the trading partner

1. With a trading partner highlighted, click **New** (upper left).
2. Enter a profile name, select a status (if not **proposed**), and optionally enter a start date, end date, and maximum values concurrent conversations and invocations.
3. Click **Save**. See Figure 49.

Figure 49 Newly Created Trading Partner Profile



Trading Partner Profile: GreenfieldProfile

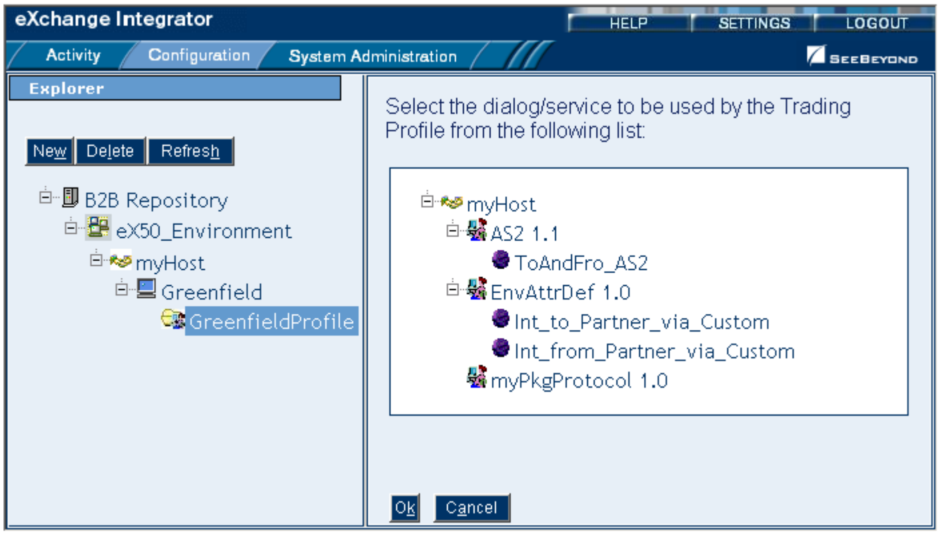
Profile Name: * GreenfieldProfile
Status : * proposed
Start Date: 11/18/2003
End Date: 11/17/2004
Max Concurrent Conversations : 20
Max Invocation Limit: 50
Last Updated Date: 2003-10-14 17:26:27
Last Updated By: Administrator
Creation Date: 2003-10-14 17:26:27
Created By: Administrator

The explorer tree displays a new trading partner ***profile***.

4 In the explorer tree, click the newly created profile, and then click **New** (upper left).

You are creating a new business protocol binding based on one of the business protocols that are the leaves of the tree now displayed on the canvas. See Figure 50.

Figure 50 Business Protocols Organized Under Enveloping Protocols



Select the dialog/service to be used by the Trading Profile from the following list:

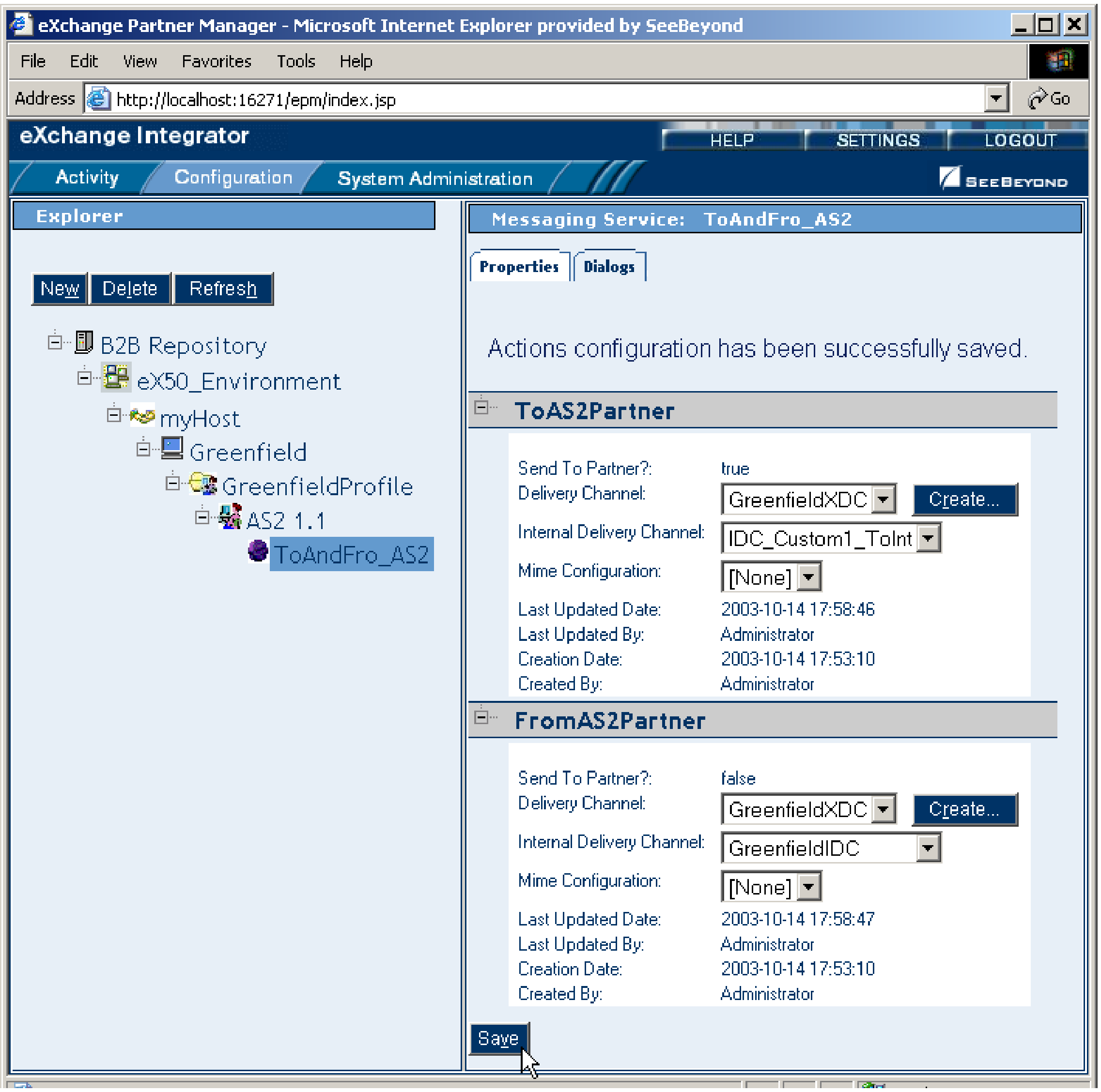
5 Click one of the business protocols and then click **OK**.

The business protocol appears as the bottommost leaf of the explorer tree, under the enveloping protocol (standard or custom) with which it is associated.

6 Click **Save**, and then click the **Dialogs**. tab to display the actions that you defined for that business protocol.

7 For the actions, assign appropriate parameter values and click **Save**. See Figure 51.

Figure 51 Business Protocol in Explorer Tree with Actions Shown on Canvas


eXchange Partner Manager - Microsoft Internet Explorer provided by SeeBeyond
File Edit View Favorites Tools Help
Address http://localhost:16271/epm/index.jsp
Go
eXchange Integrator
HELP SETTINGS LOGOUT
Activity Configuration System Administration
SeeBeyond
Explorer
New Delete Refresh
B2B Repository
eX50_Environment
myHost
Greenfield
GreenfieldProfile
AS2 1.1
ToAndFro_AS2
Properties Dialogs
Create...
Create...
Save

Messaging Service: ToAndFro_AS2

Actions configuration has been successfully saved.

ToAS2Partner

Send To Partner?:	true
Delivery Channel:	GreenfieldXDC
Internal Delivery Channel:	IDC_Custom1_ToInt
Mime Configuration:	[None]
Last Updated Date:	2003-10-14 17:58:46
Last Updated By:	Administrator
Creation Date:	2003-10-14 17:53:10
Created By:	Administrator

FromAS2Partner

Send To Partner?:	false
Delivery Channel:	GreenfieldXDC
Internal Delivery Channel:	GreenfieldIDC
Mime Configuration:	[None]
Last Updated Date:	2003-10-14 17:58:47
Last Updated By:	Administrator
Creation Date:	2003-10-14 17:53:10
Created By:	Administrator

8 In the explorer tree, click the trading partner to display the Trading Partner Configuration canvas.

9 Click **Activate**. In response to the prompt, click **Activate**.

A success message confirms that the trading partner configuration is now in the database, and accessible to Channel Manager and other partner lookup facilities.

7.2 Monitoring B2B Protocol Pipelines

You use standard eGate tools within Enterprise Manager to monitor your B2B protocol pipelines.

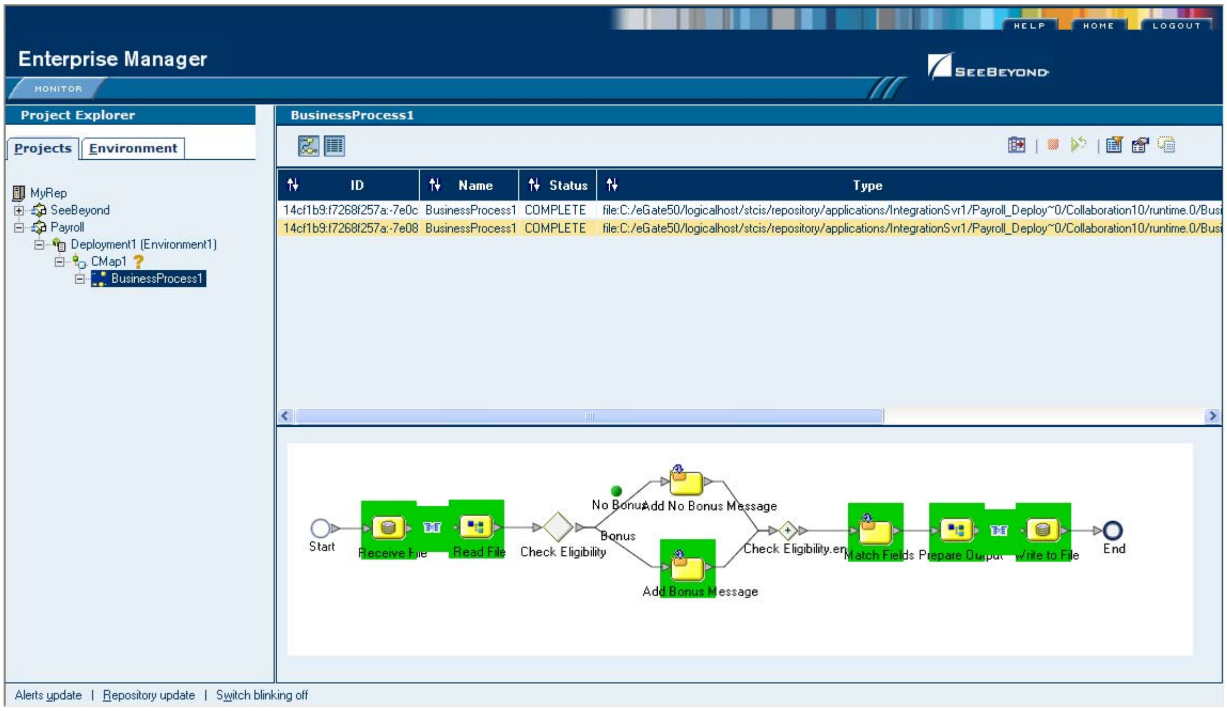
Before you begin

- You must already have activated a project whose connectivity map contains one or more services for B2B protocol pipelines.
- You must already have bootstrapped a logical host to run this project.

To monitor a B2B protocol pipeline

1. Start **Enterprise Manager**. and click the **Home** tab.
2. Select the **Monitor** icon to bring up the tree structure which allows you to navigate through projects or environments.
3. Select the **Projects** tab.
4. Navigate to the correct **Project/Deployment Profile/Connectivity Map**, and select the protocol pipeline name.

Figure 52 Monitor View



ID	Name	Status	Type
14cf1b9:f7268f257a:-7e0c	BusinessProcess1	COMPLETE	file:C:/eGate50/logicalhost/stcis/repository/applications/IntegrationSvr1/Payroll_Deploy~0/Collaboration10/runtime.0/Busi
14cf1b9:f7268f257a:-7e08	BusinessProcess1	COMPLETE	file:C:/eGate50/logicalhost/stcis/repository/applications/IntegrationSvr1/Payroll_Deploy~0/Collaboration10/runtime.0/Busi

Monitor options

In the right pane of Enterprise Manager, you can see the model and/or the instance list (the toggle at the top of the pane controls the view).

The options are:

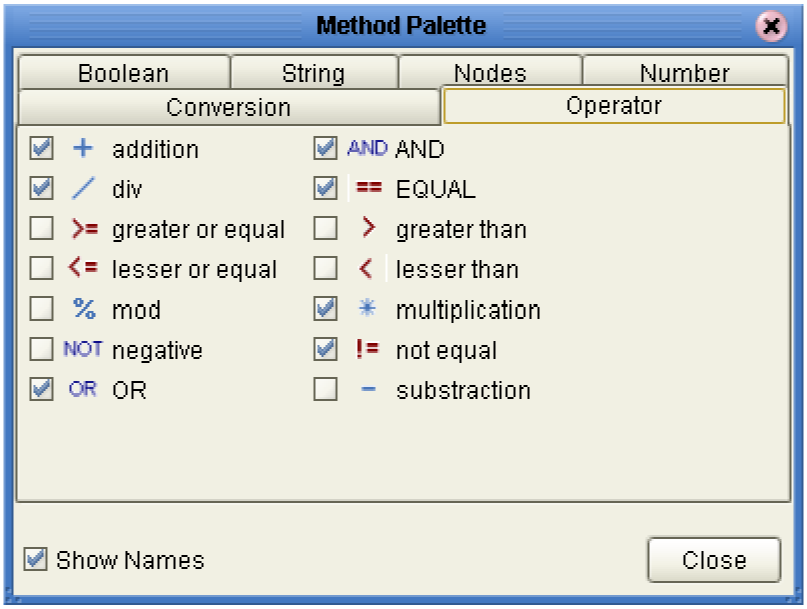
- show the model
- show the instance list
- show the instance list and model

Appendix A

Method Palette

A.1 Operators

Figure 53 Method Palette: Operator tab



Method Palette

Boolean | String | Nodes | Number
Conversion | Operator

☑ + addition
☑ / div
□ >= greater or equal
□ <= lesser or equal
□ % mod
□ NOT negative
☑ OR OR

☑ AND AND
☑ == EQUAL
□ > greater than
□ < lesser than
☑ * multiplication
☑ != not equal
□ - substraction

☑ Show Names

Close

Table 9 Operator Methods

Method Box	Name	Description/Usage
 AS2_FROM AS2_TO AS2_HOST	addition	Adds the value of *number1* to the value of *number2*, returns the sum.

Table 9 Operator Methods (Continued)

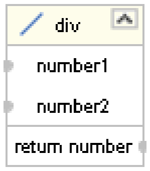
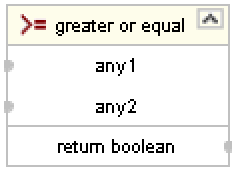
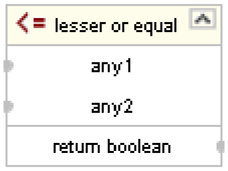
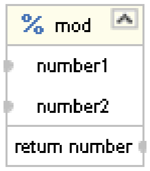
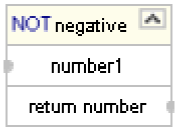
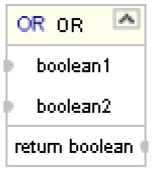
Method Box	Name	Description/Usage
	division	Divides the value of *number1* by the value of *number2*, returns the quotient.
	greater_or_equal	Returns Boolean true if *number1* is greater than or equal to *number2*; otherwise, returns Boolean false.
	lesser_or_equal	Returns Boolean true if *number1* is less than or equal to *number2*; otherwise, returns Boolean false.
	mod	
	negative	Converts the input number to negative. Result is a negative number having the same absolute value as the input number.
	or	Returns Boolean false if both *boolean1* and *boolean2* are false; otherwise, returns Boolean true.

Table 9 Operator Methods (Continued)

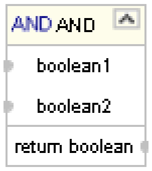
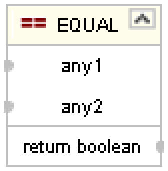
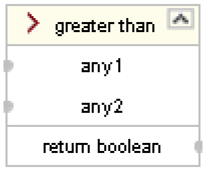
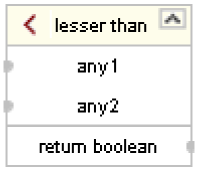
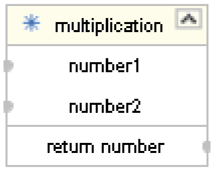
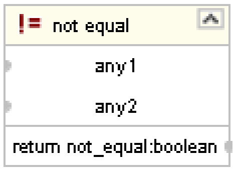
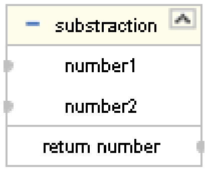
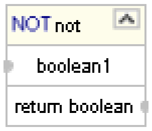
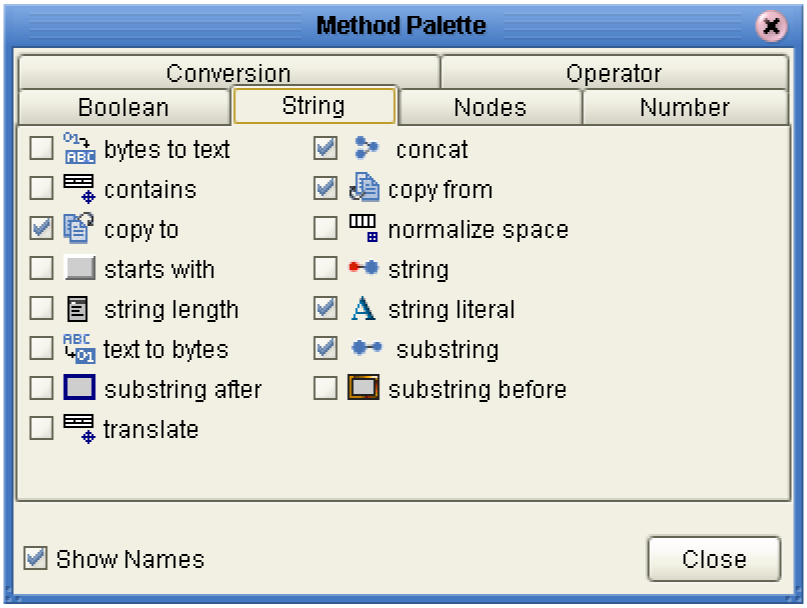
Method Box	Name	Description/Usage
	and	Returns Boolean true if both *boolean1* and *boolean2* are true; otherwise, returns Boolean false.
	equal	Returns Boolean true if *number1* is equal to *number2*; otherwise, returns Boolean false.
	greater_than	Returns Boolean true if *number1* is greater than *number2*; otherwise, returns Boolean false.
	less_than	Returns Boolean true if *number1* is less than *number2*; otherwise, returns Boolean false.
	multiplication	Multiplies the value of *number1* by the value of *number2*, returns the product.
	not_equal	Returns Boolean true if *number1* is not equal to *number2*; otherwise, returns Boolean false.

Table 9 Operator Methods (Continued)

Method Box	Name	Description/Usage
	subtraction	Subtracts the numerical value of *number2* from the numerical value of *number1*, returns the difference.
	not	Returns the inverse of *boolean1*.

2.2 String

Figure 54 Method Palette: String tab



Method Palette

Conversion | Operator
Boolean | String | Nodes | Number

☐ bytes to text
☐ contains
☑ copy to
☐ starts with
☐ string length
☐ text to bytes
☐ substring after
☐ translate

☑ concat
☑ copy from
☐ normalize space
☐ string
☑ string literal
☑ substring
☐ substring before

☑ Show Names

Table 10 String Methods

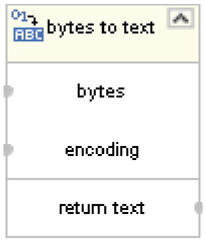
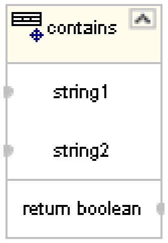
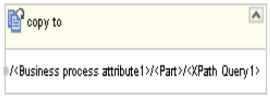
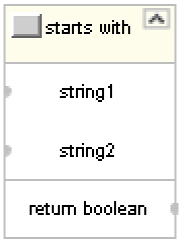
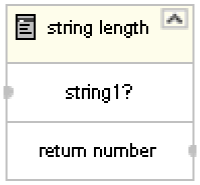
Symbol	Name	Description
	bytes to text	Decodes bytes into text using the specified encoding. If no encoding is specified, the platform's default encoding is used.
	contains	Returns true if the second string is contained within the first string, otherwise it returns false
	copy to	Allows you to type in the xpath expression for the destination of a copy operation. This is useful for entering xpath predicates. Note: This is for advanced users who are familiar with xpath and BPEL syntax.
	starts with	Returns true if the first string starts with the second string, otherwise it returns false
	string length	Returns the number of characters in a string

Table 10 String Methods (Continued)

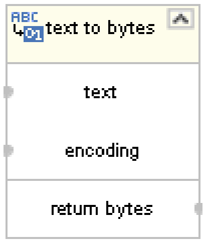
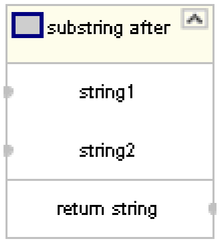
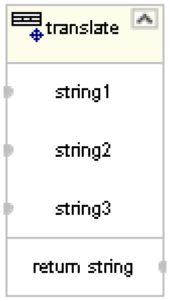
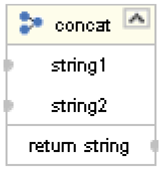
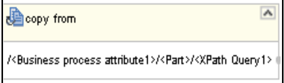
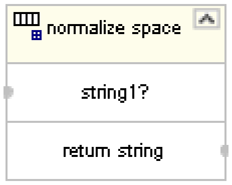
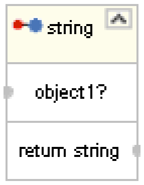
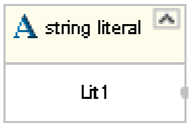
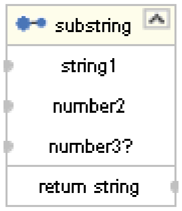
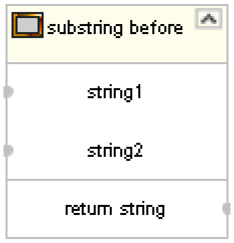
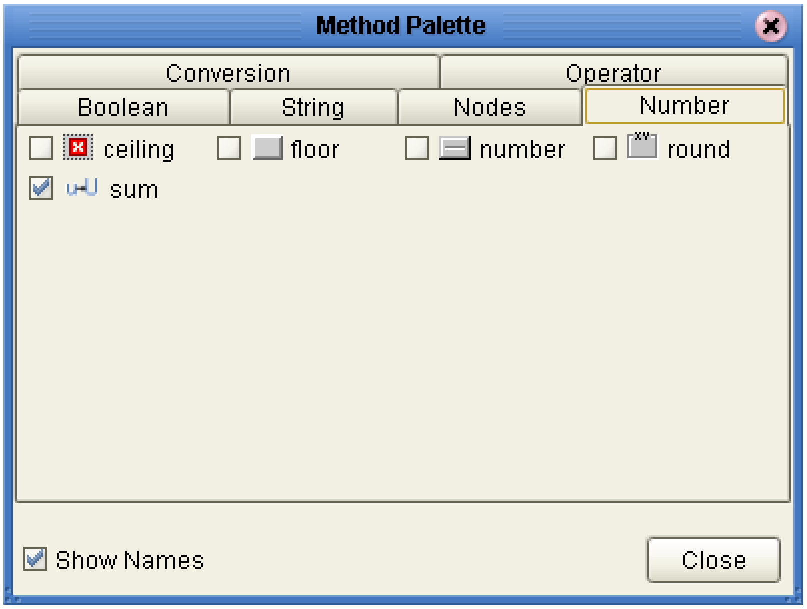
Symbol	Name	Description
	text to bytes	Encodes the input text into a sequence of bytes using the specified encoding. If no encoding is specified, the platform's default encoding is used
	substring after	Returns the part of the string in the string argument that occurs after the substring in the substring argument
	translate	Performs a character by character replacement. It looks in the value argument for characters contained in string1, and replaces each character for the one in the same position in the string2
	concat	Returns the concatenation of all its arguments
	copy from	Allows you to type in xpath expression for the source of a copy operation. This is useful for entering xpath predicates. Note: This is for advanced users who are familiar with xpath and BPEL syntax

Table 10 String Methods (Continued)

Symbol	Name	Description
	normalize space	Removes leading and trailing spaces from a string
	string	Converts the value argument to a string
	string literal	A sequence of characters of fixed length and content
 substring string1 number2 number3? return string	substring	Returns a part of the string in the string argument
	substring before	Returns the part of the string in the string argument that occurs before the substring in the substr argument

2.3 Number

Figure 55 Method Palette: Number tab



□ ceiling □ floor □ number □ round

☑ sum

☑ Show Names

Table 11 Number Methods

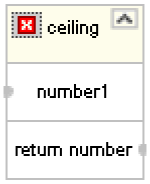
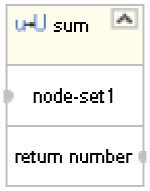
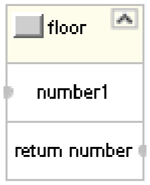
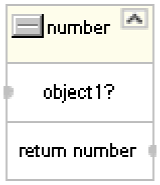
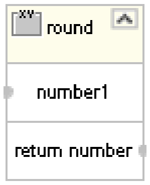
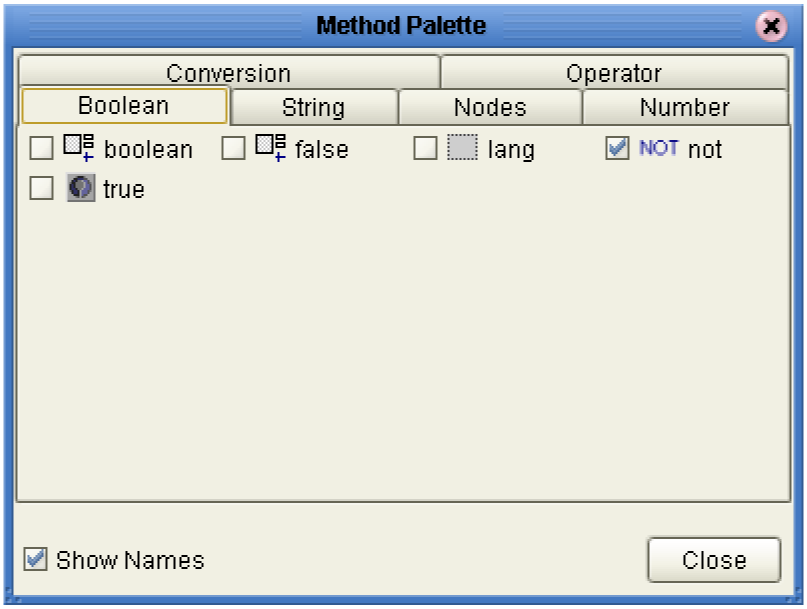
Symbol	Name	Function
	ceiling	Returns the smallest integer that is not less than the number argument
	sum	Returns the total value of a set of numeric values in a node-set
	floor	Returns the largest integer that is not greater than the number argument

Table 11 Number Methods (Continued)

Symbol	Name	Function
	number	Converts the value argument to a number
	round	Rounds the number argument to the nearest integer

2.4 Boolean

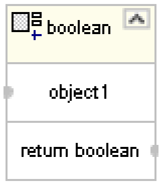
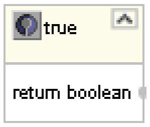
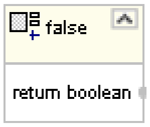
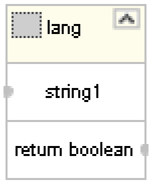
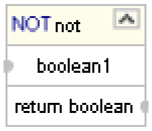
Figure 56 Method Palette: Boolean tab



□ boolean □ false □ lang ☑ not
□ true

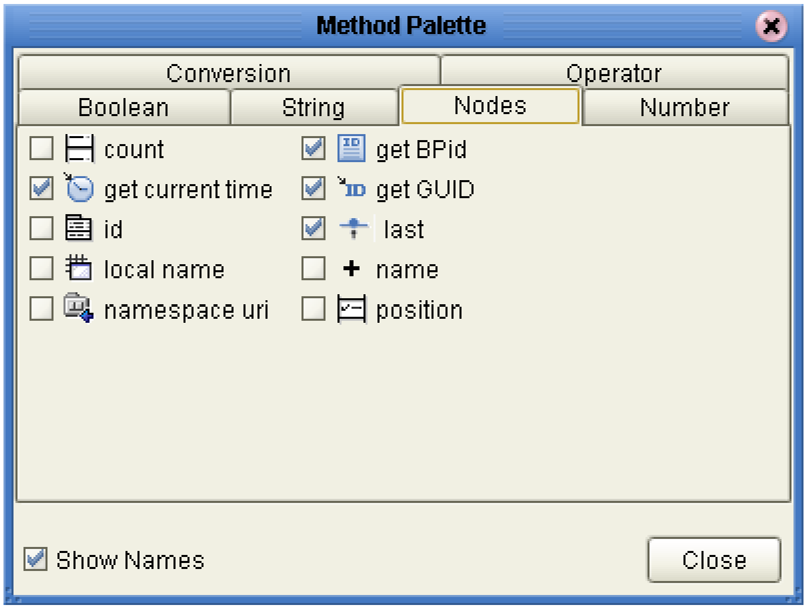
☑ Show Names

Table 12 Boolean Methods

Symbol	Name	Function
	boolean	Converts the value argument to Boolean and returns true or false
	true	Returns true
	false	Returns false
	lang	Returns true if the language argument matches the language of the the xsl:lang element, otherwise it returns false
	not	Returns true if the condition argument is false, and false if the condition argument is true

2.5 Nodes

Figure 57 Method Palette: Nodes tab



☐ count
☑ get current time
☐ id
☐ local name
☐ namespace uri
☑ get BPid
☑ get GUID
☑ last
☐ name
☐ position

☑ Show Names

Table 13 Nodes Methods

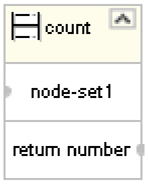
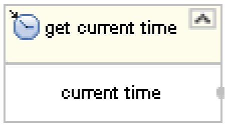
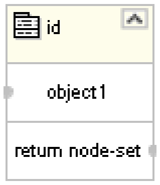
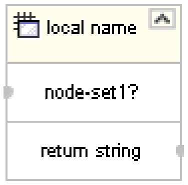
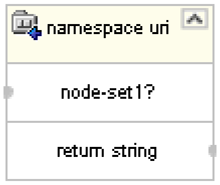
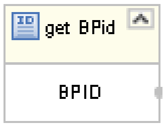
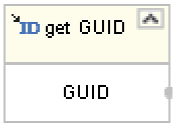
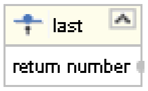
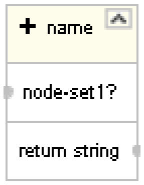
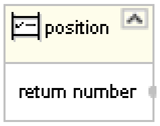
Symbol	Name	Function
	count	Returns the number of nodes in a node-set
	getCurrentTime	Gets the current time in ISO 8601 format (e.g. 2003-08-15T02:03:49.92Z).
	id	Selects elements by their unique ID

Table 13 Nodes Methods (Continued)

Symbol	Name	Function
	local name	Returns the local part of a node. A node usually consists of a prefix, a colon, followed by the local name
	namespace uri	Returns the namespace URI of a specified node
	getBPId	Gets the B2B protocol pipeline instance ID.
	getGUId	Gets a randomly generated globally unique ID.
	last	Returns the position number of the last node in the processed node list
	name	Returns the name of a node
	position	Returns the position in the node list of the node that is currently being processed