Contents

Preface ................................................................................................................................................................ix
   Audience ................................................................................................................................................................ix
   Documentation Accessibility ............................................................................................................................ ix
   Related Documents ...................................................................................................................................... x
   Conventions .................................................................................................................................................. x

Part I   Introduction to Oracle B2B

1 Introduction to Oracle B2B
   1.1 Oracle B2B and Business-to-Business E-Commerce .............................................................. 1-1
   1.2 Protocols Supported in Oracle B2B .......................................................................................... 1-2
   1.3 Oracle B2B Metadata .................................................................................................................. 1-3
   1.4 Security Features of Oracle B2B .............................................................................................. 1-4
      1.4.1 Payload Obfuscation ........................................................................................................... 1-5
      1.4.2 Restricting Access to Document Types ............................................................................ 1-6
   1.5 How Does Oracle B2B Fit into a SOA Implementation? .......................................................... 1-8
   1.6 Sending a Purchase Order: An Example of a SOA Implementation ........................................ 1-8
   1.7 Oracle B2B Samples and Cookbooks ...................................................................................... 1-10
   1.8 Administering Oracle B2B in the Oracle Fusion Middleware Environment .................... 1-10

2 Getting Started with Oracle B2B
   2.1 What You Need to Get Started with Oracle B2B................................................................. 2-1
   2.2 Logging in to Oracle B2B ......................................................................................................... 2-2
      2.2.1 Finding Port Information .................................................................................................... 2-3
      2.2.2 Accessing Oracle B2B Through Single Sign-On (SSO) .................................................. 2-4
      2.2.3 Enabling the weblogic User for Logging in to Oracle B2B ............................................. 2-5
   2.3 Using the Oracle B2B Interface .................................................................................................. 2-5
      2.3.1 Administration ...................................................................................................................... 2-5
      2.3.2 Partners ................................................................................................................................... 2-6
      2.3.3 Reports ................................................................................................................................. 2-7
      2.3.4 Metrics ................................................................................................................................... 2-8
   2.4 Creating a B2B Transaction: An Overview of the Process Flow ........................................... 2-9
   2.5 Using Oracle B2B in the Oracle JDeveloper Environment ..................................................... 2-14
      2.5.1 How To Use B2B Binding Components in a SOA Composite Application ................. 2-15
      2.5.2 About Using the JMS Integration Type in the B2B Configuration Wizard ................. 2-23
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.3</td>
<td>The B2BX12OrderGateway Composite in the Fusion Order Demo</td>
<td>2-24</td>
</tr>
<tr>
<td>2.6</td>
<td>What You May Need To Know About Using Oracle B2B</td>
<td>2-25</td>
</tr>
<tr>
<td>2.6.1</td>
<td>Enabling Debug Mode at Run Time</td>
<td>2-25</td>
</tr>
<tr>
<td>2.6.2</td>
<td>Logging Out: SSO Logout Configuration for Oracle Access Manager</td>
<td>2-25</td>
</tr>
<tr>
<td></td>
<td><strong>Part II</strong> Oracle B2B Process Flow</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Creating Guideline Files</td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction to Oracle B2B Document Editor</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>Installing Oracle B2B Document Editor</td>
<td>3-3</td>
</tr>
<tr>
<td>3.3</td>
<td>Creating Guideline Files: EDIFACT D98 Example</td>
<td>3-3</td>
</tr>
<tr>
<td>4</td>
<td>Creating Document Definitions</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction to Document Protocols</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1.1</td>
<td>The Document Hierarchy</td>
<td>4-2</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Document Protocols with Acknowledgments</td>
<td>4-3</td>
</tr>
<tr>
<td>4.2</td>
<td>Creating Document Definitions</td>
<td>4-3</td>
</tr>
<tr>
<td>4.3</td>
<td>Deleting a Document Definition</td>
<td>4-7</td>
</tr>
<tr>
<td>5</td>
<td>Configuring Trading Partners</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction to Trading Partners</td>
<td>5-1</td>
</tr>
<tr>
<td>5.2</td>
<td>Creating Trading Partner Profiles</td>
<td>5-2</td>
</tr>
<tr>
<td>5.3</td>
<td>Adding Trading Partner Users</td>
<td>5-7</td>
</tr>
<tr>
<td>5.4</td>
<td>Adding Document Definitions</td>
<td>5-11</td>
</tr>
<tr>
<td>5.5</td>
<td>Configuring Channels</td>
<td>5-12</td>
</tr>
<tr>
<td>5.5.1</td>
<td>About MLLP</td>
<td>5-29</td>
</tr>
<tr>
<td>5.5.1.1</td>
<td>Overriding the Connection Mode</td>
<td>5-30</td>
</tr>
<tr>
<td>5.5.1.2</td>
<td>Generic Support for TCP</td>
<td>5-30</td>
</tr>
<tr>
<td>5.5.1.3</td>
<td>Dynamic Endpoints</td>
<td>5-31</td>
</tr>
<tr>
<td>5.5.1.4</td>
<td>Using a Transport Callout to Extract Custom Headers</td>
<td>5-32</td>
</tr>
<tr>
<td>5.5.1.5</td>
<td>Message Sequencing in MLLP Exchanges</td>
<td>5-32</td>
</tr>
<tr>
<td>5.5.1.5.1</td>
<td>Outbound Message Sequencing</td>
<td>5-32</td>
</tr>
<tr>
<td>5.5.1.5.2</td>
<td>Inbound Message Sequencing</td>
<td>5-33</td>
</tr>
<tr>
<td>5.5.1.5.3</td>
<td>Sequencing Without Dispatching</td>
<td>5-33</td>
</tr>
<tr>
<td>5.6</td>
<td>Using the Auto Create Agreement Feature</td>
<td>5-34</td>
</tr>
<tr>
<td>5.7</td>
<td>Using Identifiers for Trading Partner Lookup</td>
<td>5-34</td>
</tr>
<tr>
<td>6</td>
<td>Creating and Deploying Trading Partner Agreements</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction to Agreements</td>
<td>6-1</td>
</tr>
<tr>
<td>6.2</td>
<td>Creating an Agreement</td>
<td>6-2</td>
</tr>
<tr>
<td>6.3</td>
<td>Deploying an Agreement</td>
<td>6-5</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Redeploying an Agreement</td>
<td>6-6</td>
</tr>
<tr>
<td>6.4</td>
<td>Deleting and Exporting Agreements</td>
<td>6-7</td>
</tr>
<tr>
<td></td>
<td><strong>Part III</strong> Oracle B2B Administration</td>
<td></td>
</tr>
</tbody>
</table>
7 Using Document Protocols

7.1 Using the Custom Document Protocol

7.1.1 How to Configure the XPath Expression for a Custom XML Document

7.1.1.1 Option 1: Specify the XPath and the Matching Value

7.1.1.2 Option 2: Check for the Existence of a Node

7.1.1.3 Option 3: Check the Value of an Attribute

7.2 Using the EDIFACT Document Protocol

7.3 Using the EDI X12 Document Protocol

7.4 Using the HL7 Document Protocol

7.5 Using the Positional Flat File Document Protocol

7.6 Using the RosettaNet Document Protocol

7.6.1 PIPs

7.6.2 RosettaNet Validation

7.7 Using the UCCNet Document Protocol

7.7.1 Creating a 1Sync Document

7.8 Changing Document Details

7.8.1 Changing Document Definitions After Deploying an Agreement

7.8.2 Changing Document Definitions After Importing Metadata

7.9 Using Document Routing IDs

8 Managing Deployments

8.1 Introduction to Agreement Deployment States

8.2 Managing Deployed Agreements

8.2.1 Searching for Deployed Agreements

8.2.2 Changing the Deployment State

8.2.3 Exporting an Active Agreement

9 Creating Types

9.1 Creating Custom Identifier Types

9.2 Creating Custom Contact Information Types

9.3 Creating Custom Trading Partner Parameter Types

10 Importing and Exporting Data

10.1 Importing and Exporting the Design-Time Repository

10.2 What Is Copied When You Import or Export from the Import/Export Tab

10.3 About the Exported File

10.3.1 Exported ZIP Files Containing Agreement Names in Multibyte Character Languages

11 Batching EDI Messages

11.1 Setting Up a Batch

11.2 Managing Batched Messages

12 Managing Callouts

12.1 Introduction to Callouts
12.1.1 Transport Callouts ................................................................. 12-3
12.1.2 Creating a Callout Library JAR File ................................................. 12-4
12.2 Creating a Callout ........................................................................ 12-4
12.3 Including a Callout in an Agreement ............................................... 12-7
12.4 Implementing a Callout ................................................................. 12-8

13 Purging Data
13.1 Purging Design Metadata and Instance Data ...................................... 13-1

14 Configuring Listening Channels
14.1 Adding a Listening Channel and Protocol ........................................... 14-1
14.2 Using Transport Protocols ............................................................... 14-3
14.3 Adding Listening Channel Details ...................................................... 14-4
14.3.1 Transport Protocol Parameters ...................................................... 14-4
14.3.2 Channel Attributes ........................................................................ 14-7
14.3.3 Exchange Protocol Parameters ...................................................... 14-8
14.3.4 Security Parameters ................................................................. 14-8
14.4 Configuring a Listening Channel ....................................................... 14-9

15 Configuring B2B System Parameters
15.1 Setting Configuration Parameters .................................................... 15-1

Part IV Reports and Metrics

16 Creating Reports
16.1 Introduction to Reports ................................................................. 16-1
16.1.1 The Monitor User Role ............................................................. 16-1
16.1.2 Purging Messages ................................................................. 16-2
16.1.3 Resubmitting Messages from Oracle B2B .................................... 16-2
16.2 Creating Business Message Reports ............................................... 16-2
16.3 Creating Wire Message Reports .................................................... 16-5
16.4 Creating Application Message Reports .......................................... 16-7
16.5 Creating Error Reports ............................................................... 16-10
16.6 Creating Conversation Reports .................................................... 16-12

17 Using B2B Metrics
17.1 Introduction to B2B Metrics .......................................................... 17-1
17.2 B2B System Metrics ................................................................. 17-2
17.3 B2B Partner Metrics ................................................................. 17-4

Part V Scripts and Utilities

18 B2B Command-Line Tools
18.1 Prerequisites for Running the Command-line Tools .......................... 18-1
18.2 Purging Data .............................................................................. 18-2
C  Back-End Applications Interface
C.1  Mapping B2B IP_MESSAGE_TYPE to SCA Normalized Message Properties ..........  C-1
C.2  Normalized Message Properties..............................................................................  C-1

D  Exception Handling
D.1  Inbound Messages ........................................................................................................  D-1
D.1.1  Request or Response Messages.................................................................................  D-1
D.1.1.1  Inbound ebMS, AS1, and AS2 Messages.................................................................  D-2
D.1.2  Acknowledgment Messages.......................................................................................  D-2
D.1.3  Exception Messages..................................................................................................  D-2
D.2  Outbound Messages .....................................................................................................  D-2
D.3  Using a JMS Queue for Error Message Delivery .........................................................  D-3
D.4  Using a Custom Exception Queue for Error Message Delivery.................................  D-3
D.5  Inbound Exception Handling Scenarios.......................................................................  D-4
D.6  Exception Payload Definition .....................................................................................  D-6

Index
Preface

This guide describes how to use Oracle B2B.

Audience

Oracle Fusion Middleware User’s Guide for Oracle B2B is intended for businesses that need to extend business processes to trading partners, and want to design, deploy, monitor, and manage business process integrations.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at http://www.fcc.gov/cgb/consumerfacts/trs.html, and a list of phone numbers is available at http://www.fcc.gov/cgb/dro/trsphonebk.html.
Related Documents

For information about Oracle SOA Suite products, see the following:

- Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite
- Oracle Fusion Middleware Administrator’s Guide for Oracle SOA Suite
- Oracle Fusion Middleware Installation Guide for Oracle SOA Suite
- Oracle Fusion Middleware User’s Guide for Technology Adapters

For information about the Java API documentation (Javadoc), see the following:

- Oracle Fusion Middleware B2B Callout Java API Reference

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><code>monospace</code></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Part I

Introduction to Oracle B2B

This part contains the following chapters:

- Chapter 1, "Introduction to Oracle B2B"
- Chapter 2, "Getting Started with Oracle B2B"
Oracle B2B is an e-commerce gateway that enables the secure and reliable exchange of business documents between an enterprise and its trading partners. Oracle B2B supports business-to-business document standards, security, transports, messaging services, and trading partner management. With Oracle B2B used as a binding component within an Oracle SOA Suite composite application, end-to-end business processes can be implemented. Oracle B2B also supports Health Level 7, which enables health care systems to communicate with each other.

For more information about Oracle SOA Suite, see Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite.

This chapter contains the following topics:

- Section 1.1, "Oracle B2B and Business-to-Business E-Commerce"
- Section 1.2, "Protocols Supported in Oracle B2B"
- Section 1.3, "Oracle B2B Metadata"
- Section 1.4, "Security Features of Oracle B2B"
- Section 1.5, "How Does Oracle B2B Fit into a SOA Implementation?"
- Section 1.6, "Sending a Purchase Order: An Example of a SOA Implementation"
- Section 1.7, "Oracle B2B Samples and Cookbooks"
- Section 1.8, "Administering Oracle B2B in the Oracle Fusion Middleware Environment"

### 1.1 Oracle B2B and Business-to-Business E-Commerce

E-commerce is the buying and selling of products or services over the Internet, including business-to-business (B2B). In B2B e-commerce, an enterprise extends its business processes over the Internet to reach trading partners. B2B e-commerce represents classic business processes, mature business documents, and industry-tempered messaging services. It requires a unified business process platform, end-to-end instance tracking, visibility and auditing, integrated process intelligence, process and service governance, and centralized security.

You can think of an e-commerce transaction between businesses as analogous to a mail or express carrier (shipping) transaction. In both kinds of transactions, the sender must consider the details required for packaging and sending an item, and the receiver’s requirements. Table 1–1 provides an example that compares the two kinds of transactions.
This guide describes how to use Oracle B2B to define the document, the packaging, and the delivery, in addition to configuring trading partners, creating and deploying agreements, and monitoring a deployment.

1.2 Protocols Supported in Oracle B2B

Oracle B2B supports numerous industry-standard e-commerce protocols, as defined for a range of industries, including health care, retail, IT, telecom, electronics, manufacturing, the food industry, and more. Table 1–2 lists the protocols supported in Oracle B2B.

**Table 1–1 Comparing Traditional and E-Commerce Transactions**

<table>
<thead>
<tr>
<th>Traditional Shipping Transaction</th>
<th>E-Commerce Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the item to be shipped, that is, the transaction item?</td>
<td>Cell phone</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>How is the item packaged?</td>
<td>Box, bubble wrap</td>
</tr>
<tr>
<td>How is the item sent and received?</td>
<td>Truck, ship, airplane</td>
</tr>
<tr>
<td>Who is the carrier?</td>
<td>DHL, FedEx, UPS, USPS</td>
</tr>
<tr>
<td>What carrier services are required?</td>
<td>Required?</td>
</tr>
<tr>
<td></td>
<td>■ Signed receipt</td>
</tr>
<tr>
<td></td>
<td>■ Overnight/next day</td>
</tr>
<tr>
<td></td>
<td>■ Delivery attempts</td>
</tr>
</tbody>
</table>

This guide describes how to use Oracle B2B to define the document, the packaging, and the delivery, in addition to configuring trading partners, creating and deploying agreements, and monitoring a deployment.

1.2 Protocols Supported in Oracle B2B

Oracle B2B supports numerous industry-standard e-commerce protocols, as defined for a range of industries, including health care, retail, IT, telecom, electronics, manufacturing, the food industry, and more. Table 1–2 lists the protocols supported in Oracle B2B.
## 1.3 Oracle B2B Metadata

Oracle B2B instance data is stored and managed within the SOAINFRA schema of your database. Oracle B2B metadata for design-time and configuration is stored and

### Table 1–2 Protocols Supported in Oracle B2B

<table>
<thead>
<tr>
<th>Protocol Type</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document protocol</td>
<td>Custom (user-defined)</td>
</tr>
<tr>
<td></td>
<td>EDI EDIFACT, all versions</td>
</tr>
<tr>
<td></td>
<td>EDI X12, all versions</td>
</tr>
<tr>
<td></td>
<td>HL7, all versions</td>
</tr>
<tr>
<td></td>
<td>RosettaNet PIP business documents</td>
</tr>
<tr>
<td></td>
<td>Positional flat file (includes SAP iDoc)</td>
</tr>
<tr>
<td></td>
<td>UCCnet</td>
</tr>
<tr>
<td>Packaging protocol</td>
<td>MIME 1.0</td>
</tr>
<tr>
<td></td>
<td>S/MIME 2.0, S/MIME 3.0</td>
</tr>
<tr>
<td></td>
<td>SOAP</td>
</tr>
<tr>
<td></td>
<td>XML digital signature (XMLDSig)</td>
</tr>
<tr>
<td></td>
<td>XML encryption (XMLEncrypt)</td>
</tr>
<tr>
<td>Transport protocol</td>
<td>AQ</td>
</tr>
<tr>
<td></td>
<td>Email (SMTP 1.0, IMAP 1.0, POP3)</td>
</tr>
<tr>
<td></td>
<td>File</td>
</tr>
<tr>
<td></td>
<td>FTP and SFTP (SSH FTP)</td>
</tr>
<tr>
<td></td>
<td>HTTP (HTTP 1.0, HTTP 1.1) and HTTPS (HTTPS 1.0, HTTPS 1.1)</td>
</tr>
<tr>
<td></td>
<td>JMS</td>
</tr>
<tr>
<td></td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Message exchange protocol</td>
<td>AS1-1.0, AS2-1.1</td>
</tr>
<tr>
<td></td>
<td>MLLP-1.0</td>
</tr>
<tr>
<td></td>
<td>ebMS-1.0, ebMS-2.0 (ebXML Messaging Service)</td>
</tr>
<tr>
<td></td>
<td>RosettaNet-01.10, RosettaNet-V02.00</td>
</tr>
<tr>
<td></td>
<td>Generic File-1.0</td>
</tr>
<tr>
<td></td>
<td>Generic AQ-1.0</td>
</tr>
<tr>
<td></td>
<td>Generic FTP-1.0</td>
</tr>
<tr>
<td></td>
<td>Generic SFTP-1.0</td>
</tr>
<tr>
<td></td>
<td>Generic JMS-1.0</td>
</tr>
<tr>
<td></td>
<td>Generic HTTP-1.0</td>
</tr>
<tr>
<td></td>
<td>Generic Email-1.0</td>
</tr>
</tbody>
</table>

**About Document Types:** Using the Custom and positional flat file document protocols, you can use many other document types, including W3CXML Schema (OAGIS, xCBL, UBL, ebXML, and more). Use Oracle B2B Document Editor to create the guideline documents.
managed through Metadata Services (MDS), available in Oracle Fusion Middleware. See Oracle Fusion Middleware Administrator’s Guide for more information about MDS.

1.4 Security Features of Oracle B2B

Oracle B2B leverages the security features of Oracle Platform Security Services, a comprehensive security platform framework. Oracle Platform Security Service supports:

- Authentication
- Identity assertion and management
- Authorization
- The specification and management of application-specific policies
- Credential and key store management through the Credential Store Framework
- Auditing
- Role administration, and role and credential mappings
- The User and Role API
- Single sign-on solutions
- Security configuration and management
- Cryptography

The default administrator user created during Oracle SOA Suite installation is assigned the administrator role, which has access to all Oracle B2B functionality. The default administrator user can create additional users and assign the following roles:

- Host administrator—This role has access to all Oracle B2B functionality. Only a host trading partner user can have the administrator role for all data.
- Host monitor—This role can access reports and view run-time data for all trading partners.
- Remote administrator—This role has limited access to the Partners page. Users with this role can view and edit only their own design data (channels, documents, and so on); can view only those agreements for which they are a partner; and can access only their own run-time report data.
- Remote monitor—This role can access reports and view run-time data related to its own exchange with the host trading partner.


The partner data you design, deploy, and manage with the Oracle B2B user interface is secured by its centralized storage in the Metadata Service (MDS) repository.

Other security features include:

- Transport protocol-based security for HTTP, FTP, and SMTP exchanges
- Digital envelopes and certificates
- Digital signatures for host and remote trading partners
- Integration with Credential Store Framework for storing all passwords and security credentials
- Secure HTTP (using Secure Socket Layer (SSL))
Encrypted Key Store password for a host trading partner

---

**Note:** Oracle B2B run time does not support the CLIENT-CERT authentication method. Therefore, B2B is not able to post to OAM-SSO protected URLs.

See the following for more information about security:

- *Oracle Fusion Middleware Security Guide*

### 1.4.1 Payload Obfuscation

Oracle B2B supports payload obfuscation before payloads are stored in the instance repository. The security infrastructure of Oracle Fusion Middleware is used to obfuscate, store, and retrieve the payloads, and ensure that payloads in wire messages, business messages, and application messages are visible to authorized users only. The encryption algorithm is not specifiable. Keys are stored in the Credential Store.

At run time, the payload is obfuscated before it is stored in the instance repository. When this payload is retrieved from the instance store during processing, it is automatically unobfuscated so that B2B engine processes it.

Similarly, in the outbound direction, if payload obfuscation is required, then the payload is obfuscated before it is stored in the instance repository. If exchange-level encryption is specified, then the payload is encrypted using the encryption scheme specified before it is put on the wire.


When you enable payload obfuscation, consider the following:

- Large payloads, as defined in the **Large Payload Size** parameter on the **Configuration** tab, are not obfuscated because they are stored in a directory (file system) rather than the instance repository. Storing a large payload in the file system is a security risk.

- The obfuscated payload can be accessed in the UI only by authorized users who have access to the document type. The payload is unobfuscated and displayed in the UI for these authorized users. Other users cannot access the document type at all. The users can be provisioned to access document types. See Section 1.4.2, “Restricting Access to Document Types,” for information about document-type provisioning.

Obfuscation is available for payloads that use multibyte characters, and is available for non-Oracle databases.

If you migrate instance stores that contain obfuscated payloads, then you must ensure that you export the Credential Store Framework (CSF) as well, because the CSF has the key to unobfuscate those payloads (the same key is used for obfuscation and unobfuscation). If this is a new store, then no migration is required because the key is created (if not already present) the first time the payload is obfuscated.

A payload that was obfuscated and persisted in B2B is passed unobfuscated to other SOA components within a composite application, when using the Default or JMS integration types. Users viewing this unobfuscated payload in other SOA components
are responsible for ensuring that the payload is obfuscated and persisted securely, and that users are authorized to view the payload.

### 1.4.2 Restricting Access to Document Types

Note: Restricting access to document types is enabled by the `b2b.r1ps1` property, which is set to true by default.

See Oracle Fusion Middleware Administrator’s Guide for Oracle SOA Suite for more information.

Oracle B2B supports payload security by restricting access based on document type. The following user permissions for document-type access are available:

- **Admin permission for all document types**
  With this permission, the user can add, access, edit, and delete all document types. This user also has access to administrative functions such as import, export, and purge.

- **Admin permission for specified document types**
  With this permission, the user can access, edit, and delete the specified document types for which he has permission. The user is not allowed to access, edit, or delete the restricted document types. The user cannot add new document types or have access to any administrative functions such as import, export, and purge.

- **Monitor permission for all document types**
  With this permission, the user can access and view (but not edit or delete) all document types.

- **Monitor permission for specified document types**
  With this permission, the user can access and view (but not edit or delete) the specified document types. The user cannot access and view the restricted document types.

The default administrator user can restrict document-type access to other roles as follows:

- The host administrator can be granted access to all document types, in which case this user can restrict document-type access to other host or remote administrators.

- The host administrator can be granted access only to specified document types, in which case this user cannot restrict document-type access to other host or remote administrators.

- The remote administrator can be granted access to specified document types only, or all document types pertaining to the remote trading partner. In either case, the remote trading partner administrator cannot create document types in the system, or provision users for that particular remote trading partner. Users can only be provisioned by a host trading partner administrator user.

- The host monitor can be granted view-only access to all document types or to specified document types, but cannot restrict document-type access to other users.

- The remote monitor can be granted view-only access to all document types pertaining to the remote trading partner or to specified document types pertaining...
to the remote trading partner, but cannot restrict document-type access to other users.

**Note:** Admin users with access to all Administration tab functions lose admin privileges when permission for any or all document types is assigned, and the Administration tab is no longer available.


When access to specific document types is restricted, consider the following:

- New document definitions for a restricted document type cannot be added.
- No document types can be imported, exported, or purged.
- No document types can be modified on the Partners > Documents tab, as shown in Figure 1–1.

**Figure 1–1  Accessing a Restricted Document Type from the Documents Tab**

- The restricted document types are listed, but details cannot be viewed or accessed, on the following tabs:
  - Administration > Document tab
  - Reports tabs
  - Metrics tabs
- Agreements that include document definitions for restricted document types cannot be modified or exported.
In a SOA composite with a B2B binding component, restrictions on document types are not in effect. All document types are available to any user in the B2B Configuration Wizard of Oracle JDeveloper.

1.5 How Does Oracle B2B Fit into a SOA Implementation?

As a business-to-business gateway, Oracle B2B is used to extend business processes to trading partners. When Oracle B2B is used in a SOA composite application, you can model an end-to-end business process integration.

Oracle SOA Suite provides a complete set of service infrastructure components for designing, deploying, and managing composite applications. The multiple technology components of a composite application share common capabilities, including a single deployment and management model and tooling, end-to-end security, and unified metadata management. See Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite for more information.

In a SOA implementation, Oracle B2B functions as a binding component, with network protocols and services that enable message sending and receiving:

- As a service (inbound), the SOA composite application receives messages from Oracle B2B
- As a reference (outbound), the SOA composite application passes a message to Oracle B2B, which in turn sends the message to partners.

In addition to messages, Oracle B2B can also send attachments and large payloads in a SOA implementation. See Appendix A, “Performance Tuning and Large Payloads,” for information about handling large payloads.

**Note:** With the integration of the B2B, Mediator, and BPEL components within Oracle SOA Suite, the XML Gateway Internal Delivery channels are not needed in Oracle B2B 11g to communicate with Oracle E-Business Suite. This can be achieved by using the Oracle Application Adapter available in Oracle SOA Suite.

1.6 Sending a Purchase Order: An Example of a SOA Implementation

The following example describes how the components of a SOA composite application are used to send a purchase order that originates from Oracle E-Business Suite, as shown in Figure 1–2.
The outbound purchase order (P.O.) is an XML document that participates in an end-to-end business process as follows:

1. An application, for example, Oracle E-Business Suite, initiates the P.O. process. The P.O. document uses the application-generated XML.

2. Oracle Mediator receives the P.O. from Oracle E-Business Suite. The P.O. is translated to canonical XML through XSLT Mapper, and is validated by using the schema obtained when the composite application was validated. Oracle Mediator routes the message to Oracle BPEL Process Manager.

3. Oracle BPEL Process Manager receives the P.O. from Oracle Mediator. Business processes such as human workflow, business rules, and error handling can apply before Oracle BPEL Process Manager sends the P.O. back to Oracle Mediator.

4. Oracle Mediator receives the P.O. from Oracle BPEL Process Manager. The P.O. is transformed from canonical XML to the target XML through XSLT Mapper and then routed to Oracle B2B.

5. Oracle B2B receives the P.O. from Mediator, translates the P.O. to EDI native format, for example, and manages the interaction with the trading partner.

6. Oracle Business Activity Monitoring (BAM) monitors the end-to-end process.

See the following for more information:

- Section 2.5, "Using Oracle B2B in the Oracle JDeveloper Environment," for how to include a B2B binding component in a SOA composite application
- *Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite* for information about Oracle SOA Suite and SOA composite applications
1.7 Oracle B2B Samples and Cookbooks

The B2B samples guide you through the steps to create guideline files, design B2B transactions, deploy and monitor trading partner agreements, and create and deploy SOA composite applications. The composite applications include a B2B binding component and use the document definitions that you create in Oracle B2B.

Samples are available for the following document types:

- 1Sync
- Custom XML
- EDI EDIFACT
- EDI X12
- HL7
- RosettaNet

These end-to-end samples are based on a scenario in which two trading partners, Acme and GlobalChips, participate in a transaction. Acme is the initiator (the buyer, in the case of a purchase order scenario) and GlobalChips is the responder (the seller in a purchase order scenario). In the HL7 sample, Acme (initiator) sends an ADT_A01 admit patient message and receives an ACK_A01 acknowledgment from GlobalChips.

The samples include instructions and sample document definition files for you to create all the documents, agreements, and SOA composites you need to run the samples. The completed SOA composite application is also provided for each sample.

To download the samples, go to http://www.oracle.com/technology/sample_code/products/b2b

The Oracle B2B cookbooks provide detailed information on the implementation of ebXML, EDI (EDIFACT and X12), and HL7 in Oracle B2B. The cookbooks include configuration steps, performance best practices, discussions of typical errors, and FAQs. The following cookbooks are available:

- An ebXML Cookbook for Oracle B2B
- An EDI Cookbook for Oracle B2B
- An HL7 Cookbook for Oracle B2B

To download the cookbooks, go to http://www.oracle.com/technology/products/soa/b2b/index.html

See Section 2.1, "What You Need to Get Started with Oracle B2B," for the components required to use the samples and cookbooks.

1.8 Administering Oracle B2B in the Oracle Fusion Middleware Environment

The following components provide monitoring, configuration, and performance tuning capabilities for Oracle B2B:

  
  See Section A.1, "Settings for Performance Tuning," for more information.

- Oracle WebLogic Server Administration Console—Administer settings for performance tuning.
See Section A.1, "Settings for Performance Tuning," for more information.

- Oracle Enterprise Manager Fusion Middleware Control—Set B2B Server properties to enable Enterprise Manager metrics and monitor the B2B Infrastructure.

See the following for more information:
- Section B.1, "Properties To Set in Fusion Middleware Control"

Within the Oracle B2B interface, use the following for monitoring and configuration:

- **Administration > Configuration** tab

- **Administration > Reports** tab
  See Chapter 16, "Creating Reports."

- **Administration > Metrics** link
  See Chapter 17, "Using B2B Metrics."

This chapter contains the following topics:

- Section 2.1, "What You Need to Get Started with Oracle B2B"
- Section 2.2, "Logging in to Oracle B2B"
- Section 2.3, "Using the Oracle B2B Interface"
- Section 2.4, "Creating a B2B Transaction: An Overview of the Process Flow"
- Section 2.5, "Using Oracle B2B in the Oracle JDeveloper Environment"
- Section 2.6, "What You May Need To Know About Using Oracle B2B"

2.1 What You Need to Get Started with Oracle B2B

In addition to installing Oracle SOA Suite, which includes Oracle B2B, you will need to install:

- Oracle B2B Document Editor
- Oracle JDeveloper

Use the standards-based templates of Oracle B2B Document Editor to create guideline files. Then, using Oracle B2B, you create and deploy the transaction as part of a B2B agreement. To include the B2B transaction in a SOA composite application, use Oracle JDeveloper, as shown in Figure 2–1.
2.2 Logging in to Oracle B2B

These instructions assume that you have installed Oracle SOA Suite, which includes Oracle B2B. See Oracle Fusion Middleware Installation Guide for Oracle SOA Suite for more information.

Use a supported Web browser:
- Microsoft Internet Explorer 7.x
- Mozilla Firefox 2.x
- Mozilla Firefox 3.x

To log in to Oracle B2B:
1. Open a supported Web browser and go to:
   
   http://hostname:port/b2bconsole

   where:
   - *hostname* is the name of the host on which Oracle SOA Suite is installed
port is the port number used by the Managed Server to listen for regular HTTP (non-SSL) connections. (In a cluster environment, the port can be the router port.) See Section 2.2.1, "Finding Port Information," for more information.

/b2bconsole (or /b2b) accesses the B2B interface (/b2b is redirected to /b2bconsole).

See Section 2.2.2, "Accessing Oracle B2B Through Single Sign-On (SSO)," for information on protecting the b2bui page by adding /b2bconsole and /b2b settings to the mod_wl_ohs.conf file of Oracle HTTP Server.

**Note:** To access Oracle B2B when SAML is enabled or in Windows Native Authentication Environments, use either of the following protected servlet URLs for automatic authentication:

- http://hostname:port/b2b/ssologin
- http://hostname:port/b2bconsole/ssologin

2. On the log-in page, enter the following:

<table>
<thead>
<tr>
<th>For This Field...</th>
<th>Do...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>Enter the default administrator user name.</td>
</tr>
<tr>
<td>Password</td>
<td>Use the administrator password from your Oracle Fusion Middleware 11g installation.</td>
</tr>
</tbody>
</table>

3. Click Login.

### 2.2.1 Finding Port Information

You can find port number information in the following ways:

- From Oracle WebLogic Server Administration Console
  1. Log in to the console.
  2. In the Domain Structure pane, shown in Figure 2–2, expand Environment and click Servers.
3. Note the Listen Port column for your server.
   - Or from `MW_HOME/user_projects/domains/your_domain_name/config/config.xml`

   ```xml
   <server>
     <name>soa_server1</name>
     <ssl>
       <name>soa_server1</name>
       <listen-port>8002</listen-port>
     </ssl>
     <machine>LocalMachine</machine>
     <listen-port>8001</listen-port>
   </server>
   ```

2.2.2 Accessing Oracle B2B Through Single Sign-On (SSO)

To log in, log out, and relog in to Oracle B2B using SSO in Oracle Identity Management, the `/b2bconsole` location must be added to the `mod_wl_ohs.conf` file of Oracle HTTP Server as follows:

```xml
<Location /b2bconsole>
  SetHandler weblogic-handler
  # PathTrim /weblogic
  ErrorPage http://WEBLOGIC_HOME:WEBLOGIC_PORT/
</Location>
```

This is in addition to the setting required for the `/b2b` location:

```xml
<Location /b2b>
  SetHandler weblogic-handler
  # PathTrim /weblogic
  ErrorPage http://WEBLOGIC_HOME:WEBLOGIC_PORT/
</Location>
```
2.2.3 Enabling the weblogic User for Logging in to Oracle B2B

For the weblogic user in Oracle Internet Directory (OID) to log in to Oracle B2B as an administrator and search for users, the OID Authenticator must have an Administrators group, and the weblogic user must be a member of that group.

To enable the weblogic user:

1. Create a weblogic user in OID using the LDAP browser. The `users.ldif` file is imported to OID as follows:

   ```
   dn: cn=weblogic,cn=Users,dc=us,dc=oracle,dc=com
   objectclass: inetorgperson
   objectclass: organizationalPerson
   objectclass: person
   objectclass: orcluser
   objectclass: orcluserV2
   objectclass: top
   sn: weblogic
   userpassword: welcome1
   uid: weblogic
   ```

2. Create an Administrators group in OID and assign the weblogic user to it. The `groups.ldif` file is imported to OID as follows:

   ```
   dn: cn=Administrators,cn=Groups,dc=us,dc=oracle,dc=com
   objectclass: groupOfUniqueNames
   objectclass: orclGroup
   objectclass: top
   owner: cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com
   uniquemember: cn=weblogic,cn=Users,dc=us,dc=oracle,dc=com
   ```

2.3 Using the Oracle B2B Interface

B2B activities are grouped as follows:

- Administration
- Partners
- Reports
- Metrics

2.3.1 Administration

Use the tabs of the Administration page, shown in Figure 2–3, to manage importing and exporting, document protocols, deployments, types, batching, callouts, purging, listening channels, and B2B configuration.
Figure 2–3 Administration Activities

See Part III, "Oracle B2B Administration" for more information.

2.3.2 Partners

Use the tabs of the Partners page, shown in Figure 2–4, to create and update trading partner information, create and update agreement information, add user information, associate documents with trading partners, set up channels, and configure the key store.

2.3.3 Reports

Use the tabs of the Reports page, shown in Figure 2–5, to create and view reports about the instance (run-time) data.
2.3.4 Metrics

Use the tabs of the Metrics page, shown in Figure 2–6, to see information about deployed agreements, such as lists of the active document types and trading partners, and run-time status, such as error messages and message counts.
2.4 Creating a B2B Transaction: An Overview of the Process Flow

Figure 2–7 shows the B2B process flow, which starts with creating B2B guideline files in Oracle B2B Document Editor and continues with using the Oracle B2B interface to create document definitions, configure trading partners, and create and deploy agreements.
Step 1: Create guideline files (ECS and optional XSD files) in Oracle B2B Document Editor

Using Oracle B2B Document Editor, shown in Figure 2–8, create transaction documents based on templates for hundreds of industry-standard protocols. The Oracle B2B Document Editor is required only when translation is needed. For XML documents, the editor is not used.

Figure 2–8 Creating a Document in Oracle B2B Document Editor

For information on Oracle B2B Document Editor, see the following:

- Chapter 3, "Creating Guideline Files"
- The Help menu of Oracle B2B Document Editor, shown in Figure 2–9.
Step 2: Create document definitions

Using the Administration > Document tab of Oracle B2B, shown in Figure 2–10, select from a list of document protocols, and then provide a document protocol version name, a document type name, and a document definition name. (For a Custom document, rather than selecting from the list of document protocols, you add a custom protocol name to the list in the Document Protocols folder.)
After selecting the ECS and optional XSD files you created in Step 1, you have created the document definition.

For more information, see Chapter 4, "Creating Document Definitions."

Step 3: Configure trading partners
Using the tabs of the Partners page of Oracle B2B, shown in Figure 2–11, add or update trading partner names, add identifiers and optional contact information, view parameters, add documents and delivery channels, and add key store information.

Figure 2–11 Configuring Trading Partners

For more information, see Chapter 5, "Configuring Trading Partners."

Step 4: Create agreements
Using the Partners > Agreement tab of Oracle B2B, shown in Figure 2–12, create an agreement that specifies the trading partners involved and associates the document definitions, channels, and identifiers with the agreement.
Step 5: Deploy agreements

Using the Administration > Deploy tab of Oracle B2B, shown in Figure 2–13, search for and deploy agreements.

For more information, see Chapter 6, "Creating and Deploying Trading Partner Agreements."
2.5 Using Oracle B2B in the Oracle JDeveloper Environment

By using an Oracle B2B binding component in a SOA composite application, you can create an end-to-end business process, such as sending a purchase order generated by a back-end application to a trading partner. Binding components establish the connection between a SOA composite application and the external world.

The B2B Configuration Wizard in Oracle JDeveloper enables you to add B2B binding components to a SOA composite application as follows:

- B2B is used as a service (inbound) to receive messages from trading partners and deliver them to SOA composite applications. Oracle B2B is the entry point to the SOA composite application.
- B2B is used as a reference (outbound) to send messages from the SOA composite application to partners.

As you follow the steps in the B2B Configuration Wizard, you are prompted to select a document definition created in Oracle B2B. Or, you can launch Oracle B2B from the wizard to create a document definition. This is the payload, or message, that you are receiving from trading partners or sending to trading partners.

**Note:** In the B2B Configuration Wizard, if SSL is enabled in the middleware (the B2B Web service), then the SSL port is detected by the wizard and the document definitions are retrieved using the SSL connection.
2.5.1 How To Use B2B Binding Components in a SOA Composite Application

To create a SOA composite application with a B2B binding component, do the following:

- **Task 1**, "Create a SOA Application and Project"
- **Task 2**, "Add Service Components"
- **Task 3**, "Add a B2B Binding Component"

See the following in *Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite* for more information on creating SOA composite applications:

- "Adding Wires"
- "Adding Security"
- "Deploying a SOA Composite Application"
- "Managing and Testing a SOA Composite Application"

**Task 1  Create a SOA Application and Project**

2. If Oracle JDeveloper is running for the first time, specify the location for the Java JDK.
3. Create a new SOA composite application, as described in Table 2–1.

### Table 2–1  SOA Composite Application Creation

<table>
<thead>
<tr>
<th>If Oracle JDeveloper...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Has no applications</td>
<td>In the Application Navigator in the upper left, click <strong>New Application</strong>.</td>
</tr>
<tr>
<td>For example, you are opening Oracle JDeveloper for the first time.</td>
<td></td>
</tr>
<tr>
<td>Has existing applications</td>
<td>From the <strong>File</strong> main menu or the <strong>Application</strong> menu:</td>
</tr>
<tr>
<td></td>
<td>1. Select <strong>New &gt; Applications</strong>. The New Gallery opens, where you can select different application components to create.</td>
</tr>
<tr>
<td></td>
<td>2. In the <strong>Categories</strong> tree, under the <strong>General</strong> node, select <strong>Applications</strong>. In the <strong>Items</strong> pane, select <strong>SOA Application</strong> and click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>

The Create SOA Application wizard starts.

4. On the Name your application page, you can optionally change the name and location for your web project. If this is your first application, from **Application Template**, select **SOA Application**. Accept the defaults for the package prefix, and click **Next**.
Using Oracle B2B in the Oracle JDeveloper Environment

5. On the Name your project page, you can optionally change the name and location for your SOA project. By default, Oracle JDeveloper adds the SOA project technology, the composite.xml that generates, and the necessary libraries to your model project. Click Next.

Note: Composite and component names cannot exceed 500 characters.

A project deployed to the same infrastructure must have a unique name across SOA composite applications. The uniqueness of a composite is determined by its project name. For example, do not perform the actions described in Table 2–2. During deployment, the second deployed project (composite) overwrites the first deployed project (composite).

Table 2–2 Restrictions on Naming a SOA Project—Caution: Doing This Overwrites the First Deployed Project

<table>
<thead>
<tr>
<th>Create an Application Named...</th>
<th>With a SOA Project Named...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application1</td>
<td>Project1</td>
</tr>
<tr>
<td>Application2</td>
<td>Project1</td>
</tr>
</tbody>
</table>

The Project SOA Settings page of the Create SOA Application wizard appears.

6. In the Configure SOA Settings page, click Empty Composite, and click Finish.

7. Select Save All from the File main menu.

See Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite for more information about creating a SOA application and project.
Task 2  Add Service Components
Service components implement the business logic or processing rules of your application.

1. From the Component Palette, select **SOA**.
2. From the **Service Components** list, drag a component into the designer.

   Figure 2–14 shows a BPEL process being added to the designer.

![Figure 2–14 Adding BPEL Process to Composite](image)

A specific dialog for the selected service components is displayed. Table 2–3 describes the available editors.

### Table 2–3 Starting Service Component Editors

<table>
<thead>
<tr>
<th>Dragging This Service Component...</th>
<th>Invokes The...</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPEL Process</td>
<td>Create BPEL Process dialog to create a BPEL process that integrates a series of business activities and services into an end-to-end process flow.</td>
</tr>
<tr>
<td>Business Rule</td>
<td>Create Business Rules dialog to create a business decision based on rules.</td>
</tr>
<tr>
<td>Human Task</td>
<td>Create Human Task dialog to create a workflow that describes the tasks for users or groups to perform as part of an end-to-end business process flow.</td>
</tr>
<tr>
<td>Mediator</td>
<td>Create Mediator dialog to define services that perform message and event routing, filtering, and transformations.</td>
</tr>
</tbody>
</table>

3. Configure the settings for a service component. For help with a service component dialog, click **Help** or press **F1**. Click **Finish**.
4. Click **OK**.
5. Select **Save All** from the **File** main menu.

See *Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite* for more information about adding service components.

Task 3  Add a B2B Binding Component
Add a service or a reference binding component.

1. From the Component Palette, select **SOA**.
2. Drag **B2B** to the **Exposed Services** or the **External References** swim lane.
   - Select **Exposed Services** for receiving inbound messages.
   - Select **External References** for sending outbound messages.

4. On the Service Name page, provide a name for the B2B service and click Next.

5. On the B2B Integration Type page, select an integration type, as described in Table 2–4.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>A B2B WSDL is generated for the SOA composite to communicate with Oracle B2B directly.</td>
</tr>
<tr>
<td>AQ</td>
<td>An AQ Adapter WSDL and JCA file are generated for the SOA composite to communicate with Oracle B2B through AQ queues.</td>
</tr>
<tr>
<td>JMS</td>
<td>A JMS Adapter WSDL and JCA file are generated for the SOA composite to communicate with Oracle B2B through JMS queues.</td>
</tr>
</tbody>
</table>

6. On the Application Server Connection page, do one of the following:
   - From the AppServer Connection list, select an application server connection and click Next.
   - Or, click New to create an application server connection. Follow the Create Application Server Connection Wizard.

   When the application server connection is established, the following information is displayed: the user name created for the application server connection, the host name for the server instance, and the SOA Server name. The SOA servers configured and running in Weblogic are displayed when you select an application server connection. After you select a SOA server, the SSL or HTTP port is retrieved and the B2B web service URL is generated for retrieving document definitions.

   You can also click Test B2B to verify that you can connect to your Oracle B2B installation.

7. On the Operation page, select Send or Receive, as described in Table 2–5.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send</td>
<td>For outbound messages</td>
</tr>
<tr>
<td>Receive</td>
<td>For inbound messages</td>
</tr>
</tbody>
</table>

8. On the Document Definition Handling page, select the option on the Basic tab or one of the options on the Advanced tab, as described in Table 2–6.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Tab</td>
<td>-</td>
</tr>
<tr>
<td>Import Schema from B2B</td>
<td>Imports the schema from Oracle B2B (the same option as on the Advanced tab)</td>
</tr>
</tbody>
</table>
You can also select **Attachment Support** for the **Default** integration type so that the B2B WSDL file includes a message part for the attachment.

9. On the Document Definition page, expand the tree to select a document definition. If you select a document definition with multiple root elements, then select a root element to use and click **OK**.

**Table 2–7** describes other options on the page.

### Table 2–7 Document Definition Page Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Search</strong></td>
<td>Enter a definition name. Partial strings are matched if you type the beginning of the definition name. Partial strings with wildcards cannot be used.</td>
</tr>
<tr>
<td><strong>Refresh</strong></td>
<td>Retrieves the document definition list from the B2B server. Refresh after a search to see all document definitions.</td>
</tr>
<tr>
<td><strong>B2B Configuration</strong></td>
<td>Opens a browser to Oracle B2B, using the connection specified on the Application Server Connection page. In Oracle B2B, you can create a document definition, include it in an agreement, and deploy the agreement. Then return to this dialog, click <strong>Refresh</strong>, and select the new document definition.</td>
</tr>
<tr>
<td><strong>Use Routing ID</strong></td>
<td>Selects all document definitions that use a document routing ID. See Section 7.9, “Using Document Routing IDs,” for more information.</td>
</tr>
</tbody>
</table>

If you selected the
- **Default** integration type, go to Step 10.
- **AQ** integration type, go to Step 11.
- **JMS** integration type, go to Step 15.

10. On the Finish page, click **Finish**.
11. On the Service Connection page, do one of the following:

- From the **Connection** list, select a database connection and click **Next**.
- Or, click **New** to create an application server connection. Follow the Create Application Server Connection Wizard.

Table 2–8 describes the information displayed when the database connection is established.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>The user name created for the database connection.</td>
</tr>
<tr>
<td>Driver</td>
<td>The JDBC driver is provided.</td>
</tr>
<tr>
<td>Connect String</td>
<td>The JDBC connection string is provided.</td>
</tr>
<tr>
<td>JNDI Name</td>
<td>Use the default Java Naming and Directory Interface (JNDI) name or specify a custom name. This connection enables you to configure the adapter during design time and to connect to the database server during run time.</td>
</tr>
<tr>
<td>Data Source</td>
<td>Enter the JNDI name that is used to look up the data source in data-sources.xml. If you are using data-sources.xml to get the connection, then this name is required.</td>
</tr>
<tr>
<td>XA Data Source</td>
<td>Select this option if the data source name is an XA data source. An XA data source can participate in an XA global transaction that can span multiple resources. In this transaction, the application server acts as the coordinating transaction manager with multiple databases (or other resources such as JMS), each of which is involved in a single transaction. If selected, the adapter becomes part of the XA transaction. Otherwise, it is a local transaction.</td>
</tr>
</tbody>
</table>

12. On the Queue Name page, select a database schema and a queue name.

Only queues for B2B of the type **IP_MESSAGE_TYPE** are available.

13. On the Queue Parameters page, do the following:

- For an enqueue operation, enter a recipient name or a list of recipients separated by commas. If you do not enter a recipient, then the message is sent to all subscribers of the queue. This field can be overridden on a per message basis by setting the RecipientList field in the outbound header. The default value is **b2buser**.

- For a dequeue operation, enter the following:
  - Consumer: The name of the agent subscribing to the queue. This field is required and is limited to 30 characters. The default value is **b2buser**.
  - Message Selector Rule: Optional filtering logic for messages to dequeue based on the message properties or message content (for example, priority < 5 or tab.user_data.amount > 10000). If a rule is supplied, then an agent using the consumer name and the message selector rule are created in the queue. The consumer name must be a new agent name, because the adapter does not change the message selector rule of a previously created agent. No validation is performed on the logic you enter.
  - Dequeue Condition: A Boolean expression similar to the **WHERE** clause of a SQL query. This expression can include conditions on message properties, user data properties (object payloads only), and PL/SQL or SQL.
functions. If more than one message satisfies the dequeue condition, then the order of dequeuing is indeterminate, and the sort order of the queue is not honored.

14. Click Finish.

15. On the JMS Provider page, select one of the following:

- **Oracle Enterprise Messaging Service (OEMS):** Oracle WebLogic JMS or Oracle Advanced Queuing
  
  OEMS is built on JMS and the J2EE Connector Architecture (JCA), which enables you to develop and integrate distributed applications in a service-oriented architecture environment. This messaging platform provides service for message persistence and recovery.

- **Third Party:** Persists messages in a third-party JMS provider, such as Tibco JMS or IBM WebSphere MQ JMS.

16. If you selected Oracle WebLogic JMS or Oracle Advanced Queuing, then the Service Connection page appears. On the Service Connection page, do one of the following:

- From the AppServer Connection list, select an existing application server connection and click Next.
- Or, click New to create an application server connection. Follow the Create Application Server Connection Wizard.

If you selected Third Party, then the JMS Connection page appears. On the JMS Connection page, provide the JMS Connection JNDI Name and click Next.

17. If you selected Send in Step 7, then the Produce Operation Parameters page appears. If you selected Receive in Step 7, then the Consume Operation Parameters page appears.

On the Produce Operation Parameters page, provide the following information and click Next:

- **Destination Name:** Enter the JNDI name of the queue or topic to produce the message or click Browse to select a name. The value of this field is typically the JNDI name.
- **Message Body Type:** Select the message body (payload) type.
  - Text Message: Use this option when the payload is a string.
  - Bytes Message: Use this option when the payload is an array of primitive bytes.
- **Delivery Mode:** Select a message delivery mode (only if WebLogic JMS was selected).
  - Persistent: Use this option for messages that are persisted to a file system or database.
  - Non-Persistent: Use this option for messages that are not persisted and are typically held in process memory only.
- **Priority:** Select a priority value, with 9 representing the highest priority and 0 representing the lowest priority.
- **Time To Live:** Enter a value that indicates the life span of the message. If no subscribers consume the message in the given time, then the message is not
delivered. There is no limit. A value of 0 indicates that there is no expiration time.

- **JNDI Name**: Displays the JNDI name based on your selection in the Destination Name field (only if WebLogic JMS or Advanced Queuing is selected).

On the Consume Operation Parameters page, provide the following information and click **Next**:

- **Destination Name**: Enter the JNDI name of the queue or topic to consume the message or click **Browse** to select a name.
- **Message Body Type**: Select the message body (payload) type.
  - Text Message: Use this option when the payload is a string.
  - Bytes Message: Use this option when the payload is an array of primitive bytes.
- **Message Selector**: Specify filtering logic that enables you to receive messages that match certain criteria. Enter an expression between 1 and 255 characters in length. Use SQL92 syntax in this field. The JMS server uses these criteria to filter messages received by this consumer. This works with variables defined in standard JMS headers and user-defined properties. You cannot use variables or elements that are in the payload of the message.
- **Use MessageListener**: This option is set to false by default if you selected Oracle Weblogic JMS on the JMS Provider page. It is not editable. Select true or false if you selected Oracle Advanced Queuing on the JMS Provider page (only if Oracle Advanced Queuing or Third Party is selected).
- **Durable Subscriber ID**: Enter an ID for receiving messages from a JMS topic. If you do not specify an ID, then you must have an active subscription session to receive messages. If you specify an ID for topics, then you receive messages even if you do not currently have an active subscription session. When a durable subscriber is disconnected from the JMS server, the server stores unexpired messages. When the durable subscriber reconnects, the server sends the unexpired messages that accumulated (only if Third Party is selected).
- **JNDI Name**: Displays the JNDI name based on your selection in the Destination Name field (only if WebLogic JMS or Advanced Queuing is selected).
- **Enable Streaming**: When you enable this feature, the payload is streamed to a database. Use this feature for large payloads. When you enable streaming, a corresponding Boolean property, StreamPayload, is appended to the ActivationSpec properties defined in the respective .jca file. If the StreamPayload property does not exist, then the default value false is assumed. The property is applicable when processing Raw messages, XMLType messages, and ADT type messages for which a payload is specified though an ADT attribute.

18. Click **Finish**.

See the following in Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite for more information on binding components:

- "Adding Service Binding Components"
- "Adding Reference Binding Components"
- "Getting Started with Binding Components"
2.5.2 About Using the JMS Integration Type in the B2B Configuration Wizard

If you select the JMS integration type, then you must use JMS properties prefixed with `jca.jms.JMSProperty`, as in `jca.jms.JMSProperty.property_name`. (In contrast, when you select the Default or AQ integration types in the B2B Configuration Wizard, message properties such as `MSG_ID`, `INREPLYTO_MSG_ID`, `FROM_PARTY`, `TO_PARTY`, `ACTION_NAME`, `MSG_TYPE`, `DOCTYPE_NAME`, and `DOCTYPE_REVISION` can be used in the SOA composite application without any changes.)

Table 2–9 lists the JMS properties to use with the Oracle JCA Adapter for JMS.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Name When Used with the Oracle JCA Adapter for JMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG_ID</td>
<td>jca.jms.JMSProperty.MSG_ID</td>
</tr>
<tr>
<td>INREPLYTO_MSG_ID</td>
<td>jca.jms.JMSProperty.INREPLYTO_MSG_ID</td>
</tr>
<tr>
<td>FROM_PARTY</td>
<td>jca.jms.JMSProperty.FROM_PARTY</td>
</tr>
<tr>
<td>TO_PARTY</td>
<td>jca.jms.JMSProperty.TO_PARTY</td>
</tr>
<tr>
<td>ACTION_NAME</td>
<td>jca.jms.JMSProperty.ACTION_NAME</td>
</tr>
<tr>
<td>MSG_TYPE</td>
<td>jca.jms.JMSProperty.MSG_TYPE</td>
</tr>
<tr>
<td>DOCTYPE_NAME</td>
<td>jca.jms.JMSProperty.DOCTYPE_NAME</td>
</tr>
<tr>
<td>DOCTYPE_REVISION</td>
<td>jca.jms.JMSProperty.DOCTYPE_REVISION</td>
</tr>
<tr>
<td>ATTACHMENT</td>
<td>jca.jms.JMSProperty.ATTACHMENT</td>
</tr>
</tbody>
</table>

For example, these sender properties and values are added in the Assign Values dialog of the Mediator component for an outbound composite, as shown in Figure 2–15.
2.5.3 The B2BX12OrderGateway Composite in the Fusion Order Demo

The SOA sample application, Fusion Order Demo, includes a B2B composite, B2BX12OrderGateway. In this demo, a trading partner sends an EDI purchase order request (850) to B2B. The trading partner does this by dropping a file into a directory on which B2B is listening through a listening channel. B2B takes the standard EDI 850 and transforms it into a canonical form, which is a common representation of the data to invoke the Store Front Service—the composite that provides access to the storefront data transaction support to update data for customers, orders, and products.

If the purchase order amount is greater than $2,000 (as determined by a business rule), then the purchase order is routed to a Human Workflow for manual approval. For orders under $2,000, approval is not required. After the order is approved, a purchase order response (855) is generated and sent back to the trading partner.

Figure 2–16 shows the B2BX12OrderGateway composite in Oracle JDeveloper.
Download the Fusion Order Demo application ZIP file (FusionOrderDemo_R1PS1.zip) from:


See Oracle Fusion Middleware Developer’s Guide for Oracle SOA Suite for a description of the SOA sample application.

2.6 What You May Need To Know About Using Oracle B2B

This section contains topics to help with troubleshooting.

2.6.1 Enabling Debug Mode at Run Time

Use Oracle Enterprise Manager Fusion Middleware Control to enable logging (SOA Infrastructure > Logs > Log Configuration). See Oracle Fusion Middleware Administrator’s Guide for Oracle SOA Suite for more information.

An alternative is to edit the logging.xml file at

$DOMAIN_HOME/config/fmwconfig/server/managed_server/

2.6.2 Logging Out: SSO Logout Configuration for Oracle Access Manager

In web.xml, the success_url parameter of

oracle.adf.share.security.authentication.AuthenticationServlet

must contain an SSO logout URL, such as
../access/oblix/lang/en-us/logout.html, to ensure that the URL is accessible and does not result in a 404 error.

Part II
Oracle B2B Process Flow

This part describes the Oracle B2B process flow.
This part contains the following chapters:

- Chapter 3, "Creating Guideline Files"
- Chapter 4, "Creating Document Definitions"
- Chapter 5, "Configuring Trading Partners"
- Chapter 6, "Creating and Deploying Trading Partner Agreements"
Creating Guideline Files

The first step in the Oracle B2B process flow, shown in Figure 3–1, is to create document guideline files.

Figure 3–1  Oracle B2B Process Flow

Oracle B2B Document Editor is a guideline creation and implementation application for defining and managing custom document definitions for Oracle B2B transactions.

This chapter contains the following topics:

- Section 3.1, "Introduction to Oracle B2B Document Editor"
- Section 3.2, "Installing Oracle B2B Document Editor"
- Section 3.3, "Creating Guideline Files: EDIFACT D98 Example"

For complete documentation on the document editor, see the Oracle B2B Document Editor Help menu.

3.1 Introduction to Oracle B2B Document Editor

Oracle B2B Document Editor is a guideline creation and implementation application for business-to-business (B2B) electronic commerce (e-commerce). Use the document editor to simplify developing, migrating, testing, distributing, and printing your electronic business (e-business) guideline documents. You can create new guideline documents or use the document editor’s comprehensive library of standards as templates.

Using an existing standard as a template, you can create new guidelines by changing the attributes of underlying segments, elements, and codes. You can also create a guideline file from a data file.

Figure 3–2 shows the types of available document guidelines: delimited flat file, EDI, HL7 2.x, HL7v3, NCPDP, ParserSchema, positional flat file (which includes SAP iDocs), RosettaNet, and XMLSchema.
In addition to using the RosettaNet document guidelines in the document editor, you can also download standard DTD files from the RosettaNet Web site.

After creating a custom guideline file, use the Oracle B2B interface to include the documents in the document definition, as shown in Figure 3–3. See Section 4.2, "Creating Document Definitions," for more information about this step.
In Figure 3–3, EDIFACT_D98A_ORDERS.xsd and EDIFACT_D98A_ORDERS.ecs are imported to create the document definition. The ECS file is required in B2B for translating and validating documents. The XSD is optional in B2B; however, it provides an easy reference to the document schema when modeling a SOA composite for sending and receiving the document.

3.2 Installing Oracle B2B Document Editor

Oracle B2B Document Editor is installed from the Oracle B2B Document Editor CD. Oracle B2B Document Editor runs on Microsoft Windows only (Win 2000, WinXP, Vista\(^1\) 32-bit and 64-bit, and Windows Server 2003), and requires the Microsoft .NET framework (installed automatically from the CD) for full support of W3C XML Schema guidelines.

Complete installation instructions are available from the Oracle B2B Document Editor Help menu by searching on installation and displaying the Preparation topic. A list of new features in this release of the document editor is also provided.

3.3 Creating Guideline Files: EDIFACT D98 Example

The following example describes how to create the guideline files—the ECS and XSD files—required to send an EDIFACT D98A purchase order, and how to generate and validate test data files based on the D98A–ORDERS guideline.

To create the EDIFACT transaction documents for this scenario, do the following:

- Task 1, "Create the ECS File"
- Task 2, "Create the XSD File"
- Task 3, "Generate Data Using the ECS File"
- Task 4, "Analyze the Data"

\(^1\) When using Microsoft Vista, do not install Oracle B2B Document Editor in the program folder, for which admin privilege is needed.
Task 1 Create the ECS File
Using an existing EDIFACT guideline (standard) as a template, create a purchase order guideline file called *orders.ecs*.

2. Click New Document and then EDI, as shown in Figure 3–4.

Figure 3–4 Creating a New Document in Oracle B2B Document Editor

3. Expand EDIFACT and D98A.
4. Select ORDERS - Purchase order message, as shown in Figure 3–5, and click Next.
5. Ensure that **Insert Envelope Segments** is *not* selected, as shown in Figure 3–6, and click **Finish**.
Oracle B2B Document Editor is preseeded with all versions of the interchange (envelope). Oracle B2B handles the envelope based on the settings.

To override the default preseeded envelope information, create a separate ECS file with only the required envelope information (without the transaction details) and then upload it using the envelope override option available in the document version.

6. (Optional) Edit the segment-level details.

No edits are needed for this scenario, as shown in Figure 3–7.
7. From File, select Save.
8. Accept the default directory and enter *orders.ecs* for the file name.

By default, the ECS file is saved to *My Documents\Oracle\Oracle B2B\Guidelines*.

**Task 2 Create the XSD File**

Using the guideline file in its internal format (the ECS file), create an XML schema definition file (the XSD format) to use with Oracle B2B.

1. From File, select Open.
2. Select orders.ecs and click Open.

3. From File, select Export.

4. In the Export Wizard, select Oracle B2B 2.0 from the list of export types, as shown in Figure 3–8, and click Next.

   Use the Oracle B2B 2.0 export type to provide a namespace of your choice, as in urn:oracle:b2b:EDIFACT/D98A/ORDERS for this example. (Use the Oracle B2B option to have a fixed namespace provided for you.)

   Figure 3–8 Selecting the Oracle B2B 2.0 Export Type

   ![Export Wizard]

5. In the Export Destination dialog, do the following and click Next.
   - Accept the default directory
   - Select Save guideline before exporting
   - Select Show advanced options

   The XSD file is saved with the ECS file in My Documents\Oracle\Oracle B2B\Guidelines.

6. In the XSD Namespace Options dialog, do the following and click Next.
   - Select Custom namespace
     - Provide a namespace, in this example, urn:oracle:b2b:EDIFACT/D98A/ORDERS

7. In the Templates Configuration dialog, click Next.

   No edits to the elements in the template are needed in this scenario.

8. In the Conversion Options dialog, shown in Figure 3–9, do the following and click Next.
Creating Guideline Files: EDIFACT D98 Example

**Figure 3–9 Selecting Conversion Options**

- Check the **Suppress Enumeration in XSD** option. This is recommended because code lists are in the ECS file. Suppressing enumeration reduces the XSD size considerably.
- Check the **Use this export module instead of default during XData generation** option.

9. In the Document Conversion Options dialog, accept the default, **Allow to use SegmentCount macro**, and click **Next**.
   The SegmentCount macro counts the number of segments. The data type of the XSD element is changed from numeric to string to enable the count.

10. Ensure that the **Launch Oracle B2B** option is not select (it is not needed in this scenario) and click **Next**.
    If you want to start Oracle B2B, enter the URL for your B2B interface (http://host_name:port/b2b).

11. In the Macro Nodes dialog, click **Next**.
    No macros are needed for any of the nodes in this scenario.
    If you see the message “Some characters were replaced in XSD names because they are not allowed,” click **OK**.

12. Click **Finish**.
    The *orders.xsd* file is created in Oracle B2B 2.0 format.

**Task 3 Generate Data Using the ECS File**
Using the Data Generator, create a test data file based on the guideline.
1. Click **Data Generator**.
2. Select **New Test Case** and click **Next**.
3. Click **Generate** and click **Next**.
   
   This step generates new data using the specified data dictionaries.
4. Select **From a guideline file**, select **ORDERS.ecs**, and click **Next**.
5. Select **Select Envelope Segments from the Standards Database** and click **Next**.
6. Select the **Syntax 3** envelope segment, as shown in **Figure 3–10**, and click **Next**.

**Figure 3–10 Selecting Envelope Segments from the Database**

7. Select **Use directly from the Standards Database** and click **Next**.
   
   The envelope segments are not incorporated in the guideline file.
8. Select **Mandatory + Percentage of optional data** and move the slider to indicate the percentage.
9. Select **User Option** and click **Next**.
10. Select **Any size** and click **Next**.
11. Select **Do not reset** and click **Next**.
12. Set the repeat count options, depending on how many messages you want generated.
13. Select any data dictionaries you want to use.
14. Accept the default delimiters and click Next.
15. Click Output Data file name, enter C:\D98A_ORDERS.dat and click Next.
   The DAT file opens, as shown in Figure 3–11.

**Figure 3–11  The DAT File**

16. Save and close the file.

**Task 4  Analyze the Data**

Using the Analyzer, validate the data file against the orders.ecs guideline file, and test the data file against the standard to check for required segments or elements that may be missing.

1. Click Analyzer, shown in Figure 3–12.
2. Browse for D98A_orders.dat and click Next.

3. Ensure that Show Advanced Options is selected, as shown in Figure 3–13, and click Next.

4. In the Clean Up Data File dialog, click Next.
   No preprocessing is needed in this scenario.

5. In the Data Structure dialog, click Next.
   The entire document is validated by default.

6. Select the guideline file (ECS file) against which to check the data, as shown in Figure 3–14. Do the following and click Next.
   - Select From a guideline file.
   - Select orders.ecs.
7. Select Select Envelope Segments from the Standards Database and click Next. The selected guideline file (ECS file) does not contain envelope segments.

8. Select the Syntax 3 envelope segment, as shown in Figure 3–15, and click Next.

9. Select Use directly from the Standards Database and click Next.
The envelope segments are not incorporated in the guideline file.

10. In the Analyzer Mode and Outputs dialog, accept the default settings, set **Generate XData (XML)** to **Always** and click **Next**.

The results, including any error messages, are displayed, as shown in **Figure 3–16**.

**Figure 3–16  Results from the Analyzer**

![Results from the Analyzer](image)

To view the data in XML format, click the **XML** icon in the upper right corner, as shown in **Figure 3–17**.

**Figure 3–17  Viewing the Data in XML Format**

![Viewing the Data in XML Format](image)

Use the **View as XML** (shown in **Figure 3–18**) and **View as HTML** options to view the data. Click the **Save Data As** icon to export the XML report as an XML file.
Figure 3–18  Analyzer Data
Creating Document Definitions

The second step in the Oracle B2B process flow, shown in Figure 4–1, is to create document definitions.

**Figure 4–1  Oracle B2B Process Flow**

A document definition specifies the document protocol—the document protocol version and document type—that is used to validate the message. The document definition can be an ECS file, in the case of EDI and HL7 messages, or an XSD/DTD, in the case of XML messages.

The same document definition is used by both the host and remote trading partner in a transaction. It must adhere to the standards for document protocols, protocol versions, and document types. This is straightforward when you use Oracle B2B Document Editor to create the document guideline files (Step 1 in Figure 4–1) and then the Oracle B2B interface to import those files when creating the document definition (Step 2 in Figure 4–1).

This chapter contains the following topics:

- **Section 4.1, "Introduction to Document Protocols"**
- **Section 4.2, "Creating Document Definitions"**
- **Section 4.3, "Deleting a Document Definition"**

For more information on document protocols, see Chapter 7, "Using Document Protocols."

### 4.1 Introduction to Document Protocols

**Figure 4–2** shows the document protocols supported in Oracle B2B. Using the Custom protocol and the many guideline documents in Oracle B2B Document Editor, you can define most protocols. When you add a new document protocol, it is always a Custom document.
As part of the document definition, you provide the document guideline files, which are typically created in Oracle B2B Document Editor. (For Custom documents, you cannot use Oracle B2B Document Editor.) If validation is enabled, then, at run time, the payload must conform to the document definition file type you use.

4.1.1 The Document Hierarchy

You can think of a document protocol as a hierarchy, as shown in Figure 4–3.

Figure 4–3 Document Hierarchy


Figure 4–4 shows a document protocol hierarchy as it applies to EDI X12.
In the Oracle B2B interface, as you create a document definition, the document protocol hierarchy is reflected in the definition:

*DocumentProtocol—Version—DocumentType—DocumentDefinitionName*

Example 4–1 shows the hierarchy reflected in the definition for an EDI EDIFACT document.

**Example 4–1  Document Definition Name for an EDI EDIFACT Document**

- Document protocol: EDI_EDIFACT
- Document protocol version: D98A
- Document type: ORDERS
- Document definition: ORDERS_def

The resulting document definition is

EDI_EDIFACT-D98A-ORDERS-ORDERS-def

Example 4–2 shows examples of document definitions for a Health Care 7 admit/visit notification and an X12 version 4010 purchase order, respectively.

**Example 4–2  Document Definition Names for HL7 and X12 Documents**

- HL7-2.3.1-ACK_A01-ACK_A01_Doc_Def
- EDI_X12-4010-850-850def

### 4.1.2 Document Protocols with Acknowledgments

For any message flow that involves an acknowledgment, Oracle B2B sends an acknowledgment only once. Resubmission does not generate another acknowledgment if the message has already been acknowledged. If further information about the message state is needed, then the trading partner must be notified by some other means (for example, e-mail).

### 4.2 Creating Document Definitions

After using Oracle B2B Document Editor to create the transaction set files, use the Oracle B2B interface to create the document definition and import the transaction set files.

**Note:** The document version, document type, and document definition are not editable after they are created. You must delete the specific document element (version, type, or definition) and create a new one. Updating the document elements after creation can lead to metadata inconsistency, metadata validation issues, and run-time errors.

**To create a document definition:**

1. Click the Administration tab.
2. Click the Document tab.
3. Select one of the document protocols, as shown in Figure 4–5.

Figure 4–6 shows an EDI EDIFACT document for illustration.
5. Enter a version name, provide document version parameters as applicable, and click **Save**.

The version is used for document identification and can be case sensitive and use a fixed syntax, depending on the protocol.

Figure 4–7 shows the document protocol version page for an EDI EDIFACT D98A document.

6. With the new version name selected, click **New Type**.

7. Enter a document type name, provide document type parameters as applicable, and click **Save**.

Figure 4–8 shows the document type parameters page for an EDI EDIFACT D98A document.
Creating Document Definitions

Figure 4–8  Entering Document Type Parameter Information

ED1_EDIFACT-D98A-NewDocumentType
Specify the document type for this version. After the new type is saved, you can create a new document definition.

* Document Type Name
  Description

Transaction
  * Functional Group Identifier Code
  Common Access Reference
  Controlling Agency
  Transaction Association Assigned Code

For parameter descriptions, see the following:

- Table 7–1, "Document Type Parameters for a Custom Document"
- Table 7–5, "Document Type Parameters for an EDI EDIFACT Document"
- Table 7–9, "Document Type Parameters for an EDI X12 Document"
- Table 7–12, "Document Type Parameters for an HL7 Document"
- Table 7–16, "Document Type Parameters for a RosettaNet Document"

8. With the new document type name selected, click New Definition.

9. Enter a document definition name and do the following:

- Browse for an optional definition (XSD) file for any of the document protocols.
- Browse for the required transaction set ECS file for the following protocols: EDI EDIFACT, EDI X12, HL7, and positional flat file.
- Provide document definition parameters as applicable.

Figure 4–9 shows the document definition parameters page for an EDI EDIFACT D98A document.
Deleting a Document Definition

4.3 Deleting a Document Definition

To delete a document definition, first delete all agreements that use that document definition and then remove the supported document definitions from the host and all remote trading partners that reference the definition.

For parameter descriptions, see the following:

- Table 7–2, "Document Definition Parameters for a Custom Document"
- Table 7–6, "Document Definition Parameters for an EDI EDIFACT Document"
- Table 7–10, "Document Definition Parameters for an EDI X12 Document"
- Table 7–13, "Document Definition Parameters for an HL7 Document"
- Table 7–14, "Document Definition Parameters for a Positional Flat File"
- Table 7–17, "Document Definition Parameters for a RosettaNet Document"
- Table 7–19, "Document Definition Parameters for a UCCNet Document"

10. Click Save.
The third step in the Oracle B2B process flow, shown in Figure 5–1, is to configure the trading partners.

**Figure 5–1 Oracle B2B Process Flow**

Configuring a trading partner includes creating a trading partner profile (providing values for identifiers, contact information, trading partner parameters, and Key Store information); adding trading partner users; adding document definitions and assigning sender and receiver roles, and configuring channel details, including security.

This chapter contains the following topics:

- **Section 5.1, "Introduction to Trading Partners"**
- **Section 5.2, "Creating Trading Partner Profiles"**
- **Section 5.3, "Adding Trading Partner Users"**
- **Section 5.4, "Adding Document Definitions"**
- **Section 5.5, "Configuring Channels"**
- **Section 5.6, "Using the Auto Create Agreement Feature"**
- **Section 5.7, "Using Identifiers for Trading Partner Lookup"**

### 5.1 Introduction to Trading Partners

In Oracle B2B, a transaction involves two trading partners, the host trading partner and a remote trading partner. The host trading partner is typically the organization where Oracle B2B is installed. The remote trading partner is the organization with whom the host trading partner conducts an e-business transaction. A trading partner can have host (back-end) applications, databases, or customers to involve in the transaction. Either the initiator of a transaction or the responder can be the host or the remote trading partner.
The host trading partner organization configures all the trading partners, host and remote. By using the trading partner users created for each remote trading partner by the host trading partner, remote partners can access their own data in Oracle B2B. Figure 5–2 shows the steps to configure a trading partner.

**Figure 5–2 Configuring Trading Partners**

5.2 Creating Trading Partner Profiles

Oracle B2B supplies a default host trading partner name, **MyCompany**, which you update to reflect your enterprise. After you create one or more remote trading partners, use the cloning feature to create new trading partners that participate in similar transactions. Cloning copies the source trading partner’s document definitions and delivery channels (except MLLP channels), but does not copy identifiers, contacts, and users. Renaming the delivery channel in the newly created trading partner is recommended.

After you create and configure a trading partner, the information is saved as a trading partner profile in Oracle Metadata Repository. Partner data can be exported to a ZIP file by using the **Export** button on the **Profile** tab.

To create a trading partner profile, do the following:

- **Task 1, "Update the Default Host Trading Partner Name"**
- **Task 2, "Add a Remote Trading Partner"**
- **Task 3, "Add Identifier Types and Values"**
- **Task 4, "Add Contact Information"**
- **Task 5, "Add a Trading Partner Parameter and Value"**
- **Task 6, "Provide Key Store Information for the Host Trading Partner"**
Task 1  Update the Default Host Trading Partner Name
Do this the first time you set up Oracle B2B.
1. Click the Partners link.
2. Click MyCompany.
3. Click Edit, as shown in Figure 5–3.

Figure 5–3  Editing the Host Trading Partner Profile

4. Provide the host trading partner name and optional icon file, and click OK.
The optional icon file must be a 16 x 16-pixel PNG file.
The host trading partner name appears in the Partner list.

Task 2  Add a Remote Trading Partner
Do this for each remote trading partner.
1. Click the Partners link.
2. Click Add, as shown in Figure 5–4.

Figure 5–4  Adding a Remote Trading Partner

3. Provide a partner name and click OK.
The remote trading partner name appears in the Partner list.
4. (Optional) Click Edit to add a 16 x16-pixel PNG file as an icon for the remote trading partner, as shown in Figure 5–5, and click OK.
Creating Trading Partner Profiles

**Figure 5–5  Editing a Remote Trading Partner Profile**

A variation on this task is to use the clone feature. If you have already created a trading partner that is similar to a trading partner you want to create, click the **Clone** icon, as shown in **Figure 5–6**, and provide the trading partner information that is not cloned: identifiers, contacts, and users. The Clone trading partner feature does not clone the MLLP delivery channel for a remote trading partner. The MLLP delivery channel must be created manually.

**Figure 5–6  Cloning a Remote Trading Partner**

---

**Note:** Use the **Delete** icon to delete a remote trading partner. However, you cannot delete a remote trading partner that is part of a deployed trading partner agreement. You must first delete the agreement.

---

**Task 3  Add Identifier Types and Values**

Identifier types enable Oracle B2B to identify a trading partner at run time. In general, the identification process is to identify the partner, then the document, and then the partner-document pair identifies the agreement. Oracle B2B provides each trading partner with a default identifier type, **Name**, whose value is the name of the trading partner.

Add identifier types and values for both the host and remote trading partners. See Chapter 9, "Creating Types," for how to create the types that you add here.

1. Click the **Partners** link.
2. Click the **Profile** tab.
3. Select a trading partner.
4. In the **Identifiers** area, click **Add**, as shown in **Figure 5–7**.
5. From the **Type** list, select an identifier type.

   See Table 9–1, ”**Identifier Types Defined in Oracle B2B**” for descriptions of the identifier types.

6. Provide a value.

7. Repeat Steps 4 through 6 as needed.

8. Click **Save**.

**Task 4 Add Contact Information**

To add optional contact information for a trading partner, use the preseeded types—Contact Name, Email, Fax, Phone. Or, you can create a contact type on the **Administration > Types** page. See Section 9.2, ”Creating Custom Contact Information Types,” for more information.

1. Click the **Partners** link.

2. Click the **Profile** tab.

3. In the **Contact Information** area, click **Add**.

4. Select from the list under **Type** and enter a value, as shown in Figure 5–8.

**Task 5 Add a Trading Partner Parameter and Value**

Before adding an optional trading partner parameter and value for a trading partner, you must create the parameter on the **Administration > Types** page. (If you have not already created a parameter, the **Add** icon does not appear.) See Chapter 9, ”Creating Types,” for more information.

1. Click the **Partners** link.
2. Click the **Profile** tab.

3. In the **Parameters** area, click **Add**, as shown in **Figure 5–9**.

![Figure 5–9 Adding Trading Partner Parameters and Values](image)

4. Select a parameter, as shown in **Figure 5–10**, and click **OK**.

![Figure 5–10 Selecting Trading Partner Parameters](image)

5. Click **Save**.

You can also update values for a specific trading partner on this page.

**Task 6  Provide Key Store Information for the Host Trading Partner**

Add an optional Key Store password and location for host trading partner security. If a digital signature, encryption, or SSL are enabled, you must specify a Key Store location. See **Task 5, "Configure Security"** for where you specify digital signatures and encryption, and **Table 5–6** for descriptions of security parameters.

You can choose any Key Store for Oracle B2B. If you are using SSL, using the same Key Store for both B2B and Oracle WebLogic Server SSL configuration is recommended to avoid SSL-related problems when exchanging messages with trading partners.

1. Click the **Partners** link.
2. Click the Profile tab.
3. Select the host trading partner.
4. In the Key Store section, provide a password and location, as shown in Figure 5–11.

Figure 5–11 Entering Key Store Information

<table>
<thead>
<tr>
<th>Key Store</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>*********</td>
<td></td>
</tr>
<tr>
<td>Confirm Password</td>
<td>*********</td>
<td></td>
</tr>
</tbody>
</table>

| Location | HOME/server/ib/DemoIdentity.jks |

5. Click Save.

---

**Note:** If you re-enter a Key Store password that you previously entered incorrectly (which produces errors trying to connect to the Key Store), then you must restart the server after entering the correct password.

---

### 5.3 Adding Trading Partner Users

The host trading partner administrator (the default login username-password combination) can add additional host and remote trading partner users. These users can log in to Oracle B2B and access their own trading partner data only.

The following roles are available:

- Administrator role—Provides access to all Oracle B2B functionality
- Monitor role—Provides access to reporting functionality only (use of the Reports link)

Users with the administrator role can access all B2B functions for their trading partner data only. No data for other trading partners is displayed. Users with the monitor role can access report functionality for their trading partner data only. No other links and no data for other trading partners are displayed. Oracle B2B also supports restricting access based on document type. See Section 1.4.2, "Restricting Access to Document Types," for more information.

To add users, do the following:

- Task 1, "Create a New User in the Identity Store"
- Task 2, "Add a User in the Oracle B2B Interface"
- Task 3, "Add Document Types That the User Has Permission to Access"

**Task 1 Create a New User in the Identity Store**

A user must exist in the Identity Store before you can provision the user in Oracle B2B. Although there are many tools that you can use to create users, one way is to use the Security Realms function in Oracle WebLogic Server Administration Console, as shown in Figure 5–12.
Adding Trading Partner Users

Figure 5–12 Oracle WebLogic Server Administration Console—Security Realms

Then, within the myrealm settings, the Users and Groups tab displays a table of all users in your realm. Click New, and then add a user and user password on the page shown in Figure 5–13.

Figure 5–13 Oracle WebLogic Server Administration Console—Adding a New User
Task 2  Add a User in the Oracle B2B Interface

The default administrator can add users. Host administrators and remote administrators can add users (remote administrators for their own data only) if they have been granted that permission by the default administrator.

1. Click the Partners link.
2. Click the Users tab.
3. Select a trading partner.
4. Click Add.
5. Provide the user name created in Task 1 and click Search.
   Enter the user name exactly as it was created.
6. Select the Monitor or Administrator role, shown in Figure 5–14, and click OK.

Figure 5–14  Assigning the Monitor or Administrator Role

Task 3  Add Document Types That the User Has Permission to Access

The default administrator can add document types for a user. Host administrators and remote administrators can add document types for a user (remote administrators for their own data only) if they have been granted that permission by the default administrator. If no document types are added here, then the user has access to all document types.

1. In the Supported Document Types area, shown near the bottom of Figure 5–15, click Add.
2. Select a document type and click Add, as shown in Figure 5–16.

3. Repeat Steps 1 and 2 as needed.

The document types that the user has permission to access are displayed in the Document Type Names column.

The document types in the Document Type Names column can also be deleted. If all types in the list are deleted, then the user has access to all document types.
5.4 Adding Document Definitions

The Oracle B2B host administrator creates all document definitions, which are automatically assigned to the host trading partner. The host administrator can assign any document definition to a remote trading partner. For both the host and remote trading partners, the sender and receiver for each document must be identified.

For information on updating a document definition after it has been added, see Section 7.8, "Changing Document Details."

---

**Note:** Document definitions that are automatically associated with the host trading partner must be deleted from the host trading partner profile (and also from the remote trading partner profile) before you can delete a document definition (from Administration > Document).

---

Consider the scenario in which Acme (buyer) sends a purchase order to GlobalChips. As part of this transaction, Acme also receives an acknowledgment that GlobalChips (seller) received the purchase order. Therefore, this EDIFACT transaction uses two document definitions, one for the purchase order and one for the functional acknowledgment. GlobalChips receives the purchase order and also sends the acknowledgment.

For information on creating a document definition—required before you can add it to the trading partner profile—see Chapter 4, "Creating Document Definitions."

To add document definitions, do the following:

- **Task 1, "Add Document Definitions"**

**Task 1 Add Document Definitions**

Add document definitions to both host and remote trading partner profiles. You can also change document type parameters and document version parameters for the remote trading partner on this page. See Chapter 7, "Using Document Protocols," for more information.

1. Click the Partners link.
2. Click the Documents tab.
3. Select a trading partner.
4. Click Add.
5. Expand the nodes, select a document definition as shown in Figure 5–17, and click Add.
5.5 Configuring Channels

A channel defines how a message is delivered. It specifies trading partner security characteristics, the transport protocol, the exchange protocol, any exchange protocol override elements, and, if defined, support for digital envelopes, encryption credentials, digital signatures, signing credentials, and validation.

When you configure an external delivery channel for the host trading partner, it is available for all remote trading partners when you create agreements. This avoids having to create a delivery channel multiple times, once for each remote trading partner. When you configure an external delivery channel for a remote trading partner, it is available for only that remote trading partner when you create agreements. When you configure an internal delivery channel for the host trading partner—for inbound messages to Oracle B2B using the AQ, File, or JMS transports—the channel is available for only the host trading partner when you create inbound agreements.
You can also create custom JMS exception queues by configuring JMS internal delivery channels for the host trading partner. See the following for more information:

- Table 15–1, "Configuration Settings"
- Section D.4, "Using a Custom Exception Queue for Error Message Delivery"

Table 5–1 lists the channels/exchange protocols available in Oracle B2B.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS2-1.1</td>
<td>Applicability Statement 2, version 1.1—specification for using EDI over the Internet. AS2 provides S/MIME support over HTTP or HTTPS. AS2 also works with non-EDI document types such as .xml, .txt, .doc, and .xls. AS2 is also called EDI over the Internet, or EDIJNT AS2.</td>
</tr>
</tbody>
</table>
| MLLP-1.0     | Minimum Lower Layer Protocol (MLLP) is a minimalistic OSI-session layer framing protocol. MLLP (and the TCP transport protocol) are available for remote trading partners only. It is used with HL7 or Custom documents. With MLLP, the same channel can be used for sending or receiving messages, and can be configured as either the server or the client. MLLP connections can be permanent or transient:
  - Features of a permanent connection:
    - Caches the socket based on the endpoint.
    - Only one socket per endpoint is created.
    - The socket is reused for future messages.
  - Features of a transient connection:
    - A new socket is created for each message.
    - A message is sent and the listener waits for the acknowledgment.
    - When the acknowledgment is received, the socket is closed.
  See Section 5.5.1, "About MLLP," for more information. |
| ebMS-2.0     | Electronic business Extensible Markup Language (ebXML) Messaging Service (ebMS)—specification used to exchange XML documents. ebMS is built on a SOAP Web services message format. Oracle B2B supports ebMS 1.0 and 2.0 and uses the HTTP, HTTPS, and Email transport protocols and the SOAP packaging protocol. The ebMS protocol supports correlation between documents. Oracle B2B also supports XMLDSig, XML Encrypt, and gZip-based compression for large documents. |
| RosettaNet-V02.00 | RosettaNet 2.0 does not include the proprietary aspects of RosettaNet 1.1, and adds support for multiple transfer protocols, hub-based routing, attachments, payload encryption, and more. |
| RosettaNet-01.10 | Implementation guidelines for creating software applications that provide for the reliable transport of PIPs in XML-format business documents between trading partners. Guidelines are provided for transport, routing, packaging, security, signals, and trading partner agreements. RosettaNet specifies the envelope or container format that remains constant when exchanging business documents (the payloads), whereas the document exchange choreography and the XML schemas vary based on which PIP and document type are used. The RosettaNet envelope format is also independent of the specific transfer protocol you use. |
| AS1-1.0      | Applicability Statement 1—specification for using EDI over SMTP. AS1 also works with non-EDI document types such as XML and TXT files.                                                                                   |
| Generic File-1.0 | Transport by which messages are sent to or received from a file in a local file system.                                                                                                                     |
| Generic AQ-1.0 | Transport by which messages are sent to or received from Oracle AQ single or multiconsumer queues.                                                                                                          |
Depending on the channel/transport protocol selected, you provide channel details such as transport protocol parameters, channel attributes, exchange protocol parameters, and security parameters, as shown in Figure 5–19.

### Table 5–1 (Cont.) Channels/Exchange Protocols Available in Oracle B2B

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic FTP-1.0</td>
<td>Transport by which messages are sent to or received from a file at a remote FTP server.</td>
</tr>
<tr>
<td>Generic SFTP-1.0</td>
<td>Transport by which messages are sent to or received from a file at a remote SFTP server.</td>
</tr>
<tr>
<td>Generic JMS-1.0</td>
<td>Transport by which messages are sent to or received from a JMS queue or topic. If a user name and password are not provided, the local JNDI is used, including in a clustered environment, provided that the destinations are distributed. Oracle B2B does not support javax.jms.ObjectMessage.</td>
</tr>
<tr>
<td>Generic HTTP-1.0</td>
<td>Transport by which messages are sent to or received from a Web server.</td>
</tr>
<tr>
<td>Generic Email-1.0</td>
<td>Transport by which messages are sent to or received from an e-mail server.</td>
</tr>
</tbody>
</table>

The transport protocol parameters define the properties specific to a given use of a protocol endpoint. The transport is responsible for message delivery using the selected transport protocol, mode (synchronous or asynchronous), server, and protocol endpoint address (trading partner address, such as a URI). Table 5–3 describes the transport protocol parameters.
The channel attributes define the communication interface between the host trading partner’s host application and its installation. Table 5–4 describes the channel attributes.

The exchange protocol parameters define the headers, acknowledgments, and packaging that puts the headers and payload together (the message exchange mechanism). Table 5–5 describes the exchange protocol parameters.

Security parameters define features such as signing, encryption, and digital signatures. Message encryption using an AES setting is preferable, where available. Table 5–6 describes the security parameters.

Note the following:

- No security parameters are specified for the Generic protocols—Generic File-1.0, Generic AQ-1.0, Generic FTP-1.0, Generic SFTP-1.0, Generic JMS-1.0, Generic HTTP-1.0, and Generic Email-1.0.
- Security parameters do not apply to the MLLP channel.

To configure a channel for a trading partner, do the following:

- Task 1, "Add a Channel"
- Task 2, "Provide Transport Protocol Parameters"
- Task 3, "Provide Channel Attributes"
- Task 4, "Provide Exchange Protocol Parameters"
- Task 5, "Configure Security"

**Task 1  Add a Channel**

Add a channel for the responder in a B2B transaction.

1. Click the Partners link.
2. Select a trading partner.
3. Click the Channels tab.
4. Click Add.
5. Enter a channel name.
6. Select a protocol, as described in Table 5–1.

Figure 5–20 shows the list of protocols.
7. Click **Save**.

Based on the delivery channel protocol you selected in Step 6, the applicable protocol is displayed in the **Transport Protocol** field, as shown in Table 5–2.

**Table 5–2** *Delivery Channels and Transport Protocols*

<table>
<thead>
<tr>
<th>Channel Protocol Selected...</th>
<th>Transport Protocol Displayed...</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS2-1.1</td>
<td>HTTP</td>
</tr>
<tr>
<td>ebMS-2.0, ebMS-1.0</td>
<td></td>
</tr>
<tr>
<td>RosettaNet-V02.00, RosettaNet-01.00</td>
<td></td>
</tr>
<tr>
<td>Generic HTTP-1.0</td>
<td></td>
</tr>
<tr>
<td>AS1-1.0</td>
<td>Email</td>
</tr>
<tr>
<td>Generic Email-1.0</td>
<td></td>
</tr>
<tr>
<td>MLLP-1.0</td>
<td>TCP</td>
</tr>
<tr>
<td>Generic File-1.0</td>
<td>File</td>
</tr>
<tr>
<td>Generic AQ-1.0</td>
<td>AQ</td>
</tr>
<tr>
<td>Generic FTP-1.0</td>
<td>FTP</td>
</tr>
<tr>
<td>Generic SFTP-1.0</td>
<td>SFTP</td>
</tr>
<tr>
<td>Generic JMS-1.0</td>
<td>JMS</td>
</tr>
</tbody>
</table>

**Task 2** *Provide Transport Protocol Parameters*

1. Click the **Transport Protocol Parameters** tab.

2. Provide transport protocol parameters, as described in Table 5–3, depending on the channel/transport protocols selected in Task 1.

*Table 5–3* describes the transport protocol parameters (listed in alphabetical order) and the protocols to which the parameters apply.
### Table 5–3 Transport Protocol Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional transport headers</td>
<td>The custom HTTP headers used to send messages to a trading partner</td>
<td>AS2 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-2.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic HTTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-V02.00 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-01.10 (optional)</td>
</tr>
<tr>
<td>Archival Directory</td>
<td>B2B channels move the processed files to this directory. By default, it is a</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>destructive read—processed files are deleted from the endpoint. In this case,</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>files are moved to the path provided.</td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td>Cache Connections</td>
<td>If enabled, file listing and processing of the file occur in the same session</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>(contrary to the default, in which listing and processing occur in different</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sessions).</td>
<td></td>
</tr>
<tr>
<td>Channel mask</td>
<td>To enable SSL for FTP, enter one of the following:</td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td></td>
<td>• Control—Encrypts the control channel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Data—Encrypts the data channel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Both—Encrypts both the data and control channels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The default is None (no SSL).</td>
<td></td>
</tr>
<tr>
<td>Cipher suites</td>
<td>Provide the preferred cipher for encryption.</td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td>Connection factory</td>
<td>The JNDI location or Java class name for the connection factory, as in</td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td></td>
<td>jms/b2b/B2BQueueConnectionFactory.</td>
<td></td>
</tr>
<tr>
<td>Connection Mode</td>
<td>Select from Client or Server.</td>
<td>MLLP-1.0 (required; for remote trading partners only)</td>
</tr>
<tr>
<td>Consumer</td>
<td>The client that receives the message.</td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td>Content type</td>
<td>The content type of the payload being sent over e-mail. The default content</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td>type is text/plain; other examples include application/xml and application/edi</td>
<td>Generic Email (optional)</td>
</tr>
<tr>
<td></td>
<td>This value is used only for the delivery channel (to send e-mail) and not</td>
<td></td>
</tr>
<tr>
<td></td>
<td>for the listening channel. On the listening channel side, intelligence is</td>
<td></td>
</tr>
<tr>
<td></td>
<td>built into the transport adapter to deal with different content types, so no</td>
<td></td>
</tr>
<tr>
<td></td>
<td>configuration is required.</td>
<td></td>
</tr>
<tr>
<td>Control port</td>
<td>Provide a value to change the default FTP port value (21)</td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td>Data port</td>
<td>The static port used for an active FTP connection</td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td>Datasource</td>
<td>The JNDI name of the database data source</td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td>Destination name</td>
<td>The JMS destination name</td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td>Destination Provider</td>
<td>Enables B2B to connect to JMS queues or topics available on remote servers.</td>
<td>Generic JMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>JNDI properties required to connected to the target server are expected as</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the value. Use ; (semicolon) as the separator for each key/value pair.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 5–3  (Cont.) Transport Protocol Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email ID</td>
<td>The destination e-mail</td>
<td>AS1 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (required)</td>
</tr>
<tr>
<td>Email Server</td>
<td>Select IMAP or POP3.</td>
<td>AS1 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (required)</td>
</tr>
<tr>
<td>Enable CCC</td>
<td>Enables B2B to authenticate in an SSL session and do the rest of the file transfer commands on a plain socket.</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td>Enable Marker</td>
<td>If enabled, creates a zero-byte file with the same name as the source, indicating completion of reading or writing. The file carries the same name as the source, but with the extension marker.</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (optional)-1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td>Encoding</td>
<td>The encoding to be used for the file transfer</td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td>Filename format</td>
<td>The following file name formats can be used:</td>
<td>Generic File (optional)</td>
</tr>
<tr>
<td></td>
<td>%FROM_PARTY%</td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td></td>
<td>%TO_PARTY%</td>
<td>Generic SFTP (optional)</td>
</tr>
<tr>
<td></td>
<td>%DOCTYPE_NAME%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%DOCTYPE_REVISION%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%MSG_ID%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%TIMESTAMP%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The following file name format can be used for ebMS documents only:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%ACTIONNAME%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>These file name formats can be used in any combination; for example,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%TO_PARTY%<em>%DOCTYPE_NAME%</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>%DOCTYPE_REVISION% _ .dat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>produces something like Acme_4010_850.dat. Any file extension is allowed.</td>
<td></td>
</tr>
<tr>
<td>Folder</td>
<td>An absolute directory path is recommended.</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (optional)</td>
</tr>
<tr>
<td>Folder name</td>
<td>An absolute directory path is recommended.</td>
<td>Generic File (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP (required)</td>
</tr>
<tr>
<td>Host name</td>
<td>The trading partner’s transport or e-mail server exchanging messages.</td>
<td>AS1 (required)</td>
</tr>
<tr>
<td></td>
<td>For the MLLP 1.0 protocol, if the connection mode is set to Server, then the host name must be the B2B server. If the connection mode is set to Client, then the host name must be the remote B2B server (MLLP server).</td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLLP-1.0 (required; for remote trading partners only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (required)</td>
</tr>
<tr>
<td>Is Binary</td>
<td>Treats the message as binary content, with no translation or validation.</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>Agreements are identified based on the file naming convention.</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td>Is Map Payload Alone</td>
<td>Indicates that the JMS map message contains only the payload</td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td>Is topic</td>
<td>Select to indicate that JMS is communicating with a topic (not a queue).</td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td>Is Van Mailbox</td>
<td>If enabled, B2B treats the endpoint as a VAN Mailbox and operates accordingly.</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
</tbody>
</table>
Table 5–3  (Cont.) Transport Protocol Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message type</td>
<td>Select a JMS message type: <strong>BYTES</strong>, <strong>TEXT</strong>, or <strong>MAP</strong>.</td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td>Minimum Age</td>
<td>Files arriving at the endpoint are processed after the time interval entered, in milliseconds.</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td>Pass phrase and Confirm pass phrase</td>
<td>If you enter a private key file location, and if the private key file is pass-phrase protected, then enter the pass phrase.</td>
<td>Generic SFTP (optional)</td>
</tr>
<tr>
<td>Password and Confirm Password</td>
<td>To use password authentication, provide a Key Store password, which is used for HTTP basic authentication.</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AS2 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-2.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic HTTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-V02.00 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-01.10 (optional)</td>
</tr>
<tr>
<td>Path</td>
<td>The absolute directory path where messages are sent from or received.</td>
<td>Generic SFTP (required)</td>
</tr>
<tr>
<td>Permanent Connection</td>
<td>When set to false (the default value), a message is sent on a new connection and the connection is closed after the ACK is received. As a receiver of the message, the connection is closed after the ACK is sent back to the trading partner. When set to true, a cached connection is used to exchange all the messages.</td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td>Polling interval</td>
<td>The time interval in seconds during which Oracle B2B polls the server for inbound messages.</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic File (not available)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP (not available)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP (not available)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (not available)</td>
</tr>
<tr>
<td>Port number (or Port)</td>
<td>AQ runs on default port 1521. SFTP runs on default port 22, which can be changed to another port. FTP runs on default port 21, which is not displayed. See the description of <strong>Control Port</strong> for how to change this port number. For the MLLP 1.0 protocol, if the connection mode is set to Server, then the port must be a valid TCP port number. If the connection mode is set to Client, then the port must be the same as the port used on the MLLP server.</td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLLP-1.0 (required; for remote trading partners only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP (required)</td>
</tr>
</tbody>
</table>
### Table 5-3  (Cont.) Transport Protocol Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preserve Filename</td>
<td>Retains the file name.</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td>Private key</td>
<td>To use public key authentication, provide the private key file location. You may also need to provide a pass phrase if the private key file is pass-phrase protected.</td>
<td>Generic SFTP (optional)</td>
</tr>
<tr>
<td>Queue name</td>
<td>The AQ queue name</td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td>Recipient</td>
<td>The AQ recipient</td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td>Send as attachment</td>
<td>If enabled, the message (payload) is sent as an e-mail attachment instead of the typical delivery in which the payload is the message body.</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (optional)</td>
</tr>
<tr>
<td>Sequence</td>
<td>Enable this property when delivering the incoming HL7 message in sequence to the back-end application is required.</td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td>SID</td>
<td>System ID to identify an Oracle database</td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td>Subject</td>
<td>The subject header of the e-mail message</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (optional)</td>
</tr>
<tr>
<td>Subscriber ID</td>
<td>The JMS subscriber ID is required if JMS is communicating with a topic.</td>
<td>Generic JMS</td>
</tr>
<tr>
<td>Timeout</td>
<td>Defines how long a transient MLLP connection keeps the socket open for the acknowledgment message. The default timeout value is 300 seconds. This parameter applies only to a transient MLLP connection (not to a permanent connection).</td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td>Transfer Type</td>
<td>Select <strong>binary</strong> or <strong>ascii</strong> for the file transfer mode.</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td>URL</td>
<td>The HTTP or HTTPS endpoint URL of the trading partner.</td>
<td>AS2 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-2.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic HTTP (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-V02.00 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-01.10 (required)</td>
</tr>
</tbody>
</table>
## Task 3  Provide Channel Attributes

1. Click the **Channel Attributes** tab.

2. Provide channel attributes, as described in Table 5–4, depending on the channel/transport protocols selected in Task 1.

Table 5–4 describes the channel attributes (listed in alphabetical order) and the protocols to which the attributes apply.
### Table 5–4  Channel Attributes

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ack Mode</td>
<td>Select <strong>Sync</strong>, <strong>Async</strong>, or <strong>None</strong>, for the mode in which the trading partner receives messages. Select <strong>None</strong> for all generic exchanges. For MLLP exchanges, select <strong>Sync</strong> or <strong>Async</strong> for a transient connection. Select <strong>None</strong> for a permanent connection.</td>
<td>AS1 (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) MLLP-1.0 (required; for remote trading partners only) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)</td>
</tr>
<tr>
<td>Compressed</td>
<td>Select for message compression.</td>
<td>AS1 (optional) AS2 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is available only with AS1 and AS2, although it may appear in the B2B interface for other protocols.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional</td>
<td>AS1 (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) MLLP-1.0 (optional; for remote trading partners only) Generic File (optional) Generic AQ (optional) Generic FTP (optional) Generic HTTP (optional) Generic HTTP (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional) Generic SFTP (optional) Generic JMS (optional) Generic Email (optional)</td>
</tr>
<tr>
<td>Enable/Disable Channel</td>
<td>The channel is the communication interface between the host trading partner’s host application and its installation.</td>
<td>MLLP-1.0 (required; for remote trading partners only)</td>
</tr>
</tbody>
</table>


Caution: While the B2B interface permits you to select invalid protocols when Internal is selected, do not select any protocols other than the generic protocols.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>Select this option if the channel is internal to the host trading partner’s enterprise.</td>
<td>If this option is checked, then only the generic protocols are valid: Generic File (optional) Generic AQ (optional) Generic FTP (optional) Generic HTTP (optional) Generic SFTP (optional) Generic JMS (optional) Generic Email (optional) If this option is not checked, all protocols are valid: AS1 (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) Generic File (optional) Generic AQ (optional) Generic FTP (optional) Generic HTTP (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional) MLLP-1.0 (optional; for remote trading partners only) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)</td>
</tr>
<tr>
<td>Response Mode</td>
<td>Select Sync, Async, or None.</td>
<td>AS1 (required) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) MLLP-1.0 (optional; for remote trading partners only) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)</td>
</tr>
</tbody>
</table>
**Table 5–4  (Cont.) Channel Attributes**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retry Count</td>
<td>The number of times that Oracle B2B retries to send the message.</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AS2 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-2.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic File (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic HTTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-V02.00 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-01.10 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (optional)</td>
</tr>
<tr>
<td>Retry Interval</td>
<td>The interval, specified in minutes, after which B2B attempts to resend a message. B2B tries to resend the message if the status of the message is not Complete. For a retry interval set to 2 minutes, the first retry may not be at the 120 second (2 minute) interval. This is because the first retry is done without considering the seconds at which the message was sent. For example, if the sent timestamp is 3:42:58 (3 hours, 42 minutes and 58 seconds), then 42 minutes is incremented by 2 minutes and the first retry is done at 3:44:00. Subsequent retries, if any, will follow at 3:46:00, then 3:48:00, and so on, in 2 minute intervals. For protocols with acknowledgments, B2B waits for the acknowledgment (formerly called the Time to Acknowledge parameter). If it is not received, the retry interval setting causes B2B to retry.</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AS2 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-2.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic File (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic HTTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-V02.00 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-01.10 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (optional)</td>
</tr>
<tr>
<td>Transport Callout</td>
<td>For the inbound message, B2B invokes the transport callout immediately after it receives a message from the transport. For the outbound message, B2B invokes the transport callout immediately before it sends a message to the transport.</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AS2 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-2.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic File (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic HTTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-V02.00 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-01.10 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic JMS (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email (optional)</td>
</tr>
</tbody>
</table>
Task 4  Provide Exchange Protocol Parameters

1. Click the Exchange Protocol Parameters tab.

2. Provide exchange protocol parameters, as described in Table 5–5, depending on the channel/transport protocols selected in Task 1.

Table 5–5 describes the exchange protocol parameters (listed in alphabetical order) and the protocols to which the parameters apply.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carriage Return Character</td>
<td>This value can be only one character. The carriage return character does not appear in the wire message payload. The default value is 0x0D (hexadecimal).</td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td>Custom Immediate ACK File</td>
<td>Browse for a file with a customized acknowledgment.</td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td>Discard HL7 ACK</td>
<td>Enable this property for the FA to be correlated at the transport level. This avoids the traditional message correlation that includes trading partner agreement identification, translation, and so on, thus improving performance. Because the ACK is stopped at the transport layer after correlation, it appears in the wire message report, but does not appear in the business message report. To enable the property, select one of the following codes. If the selected code is in the MSA.2 segment, then the ACK is stopped at the transport layer: AA—Application acknowledgment: Accept AE—Application acknowledgment: Error AR—Application acknowledgment: Reject CA—Application acknowledgment: Commit Accept CE—Application acknowledgment: Commit Error CR—Application acknowledgment: Commit Reject Selecting None does not enable this property. MSA.2 is the second element of MSA segment.</td>
<td>MLLP-1.0 (optional; for remote trading partners only)</td>
</tr>
<tr>
<td>Duplicate Elimination</td>
<td>If enabled, a duplicate elimination header is added for an outbound message. This flag does not apply to the inbound message flow.</td>
<td>ebMS-2.0 (optional)</td>
</tr>
<tr>
<td>End Block</td>
<td>This property is used to indicate the end of the message. Generally, End Block is sent after the message is sent to the trading partner.</td>
<td>MLLP-1.0 (optional; for remote trading partners only) For generic support for TCP only.</td>
</tr>
<tr>
<td>End Block Character</td>
<td>This value can be only one character. The end block character does not appear in the wire message payload. The default value is 0x1C (hexadecimal).</td>
<td>MLLP-1.0 (optional; for remote trading partners only) For generic support for TCP only.</td>
</tr>
<tr>
<td>Header Length</td>
<td>This property defines the header size, which is prefixed to the actual data. (This includes the start block, message length, and padded header).</td>
<td>MLLP-1.0 (optional; for remote trading partners only) For generic support for TCP only.</td>
</tr>
</tbody>
</table>
Configuring Channels

Identify TP by Delivery Channel

The trading partner is identified using the delivery channel.

Enable this parameter to identify an incoming message by the delivery channel configured for the remote trading partner (rather than by using the MLLP ID). This feature serves as an anonymous trading partner, for situations when identifying the sender is not important. If this parameter is not checked, then the MLLP ID (or some document-level identifier such as HL7 Message Application ID or HL7 Message Facility ID to identify the agreement) is required for MLLP exchanges.

Immediate ACK

Note: The MLLP immediate ACK of an incoming business message (with control number 1017, for example), prefixes the control number with A, as in A1017. This indicates to the trading partner that it is an ACK control number. If the prefixed string exceeds the permissible length (for example, if any validation rules are violated at the receiving end), use the Map ACK Control ID parameter.

An immediate acknowledgment is generated and transmitted in the TCP transport layer instead of the document layer. It is an alternative to the functional acknowledgment. It is available when the turnaround time of a functional acknowledgment is undesirable (for example, for some business-critical health care applications), because the functional acknowledgment captures translation and validation errors.

Oracle B2B can send an immediate acknowledgment in the following modes:

- Default: B2B parses the incoming HL7 message and generates an acknowledgment from it. In this mode, B2B can send the acknowledgment to the sending application with correlation details (for example, the control number from the incoming message, the sending application, and so on.) Hence, the trading partner application can correlate the incoming acknowledgment message. If mapping the MSH.10 of the ACK with the MSH.10 of the incoming business message is required, then enable the Map ACK Control ID property. By default, an Immediate ACK is a generic ACK. If generating an ACK with a trigger event is required, then enable the Map Trigger Event property.

- Simple: B2B sends the predefined acknowledgment message to the sender and does not parse the message.

- Custom: B2B sends the custom HL7 acknowledgment message based on a configurable file content. If this mode is selected, then specifying the file in the Custom Immediate ACK File property is required.

- Negative: Select this option to send an immediate ACK only in the case of exceptions.

Map ACK Control ID

Select to enable the mapping of the MSH.10 message header of the business message to the MSH.10 message header of the immediate acknowledgment.

3. Click Save.

**Task 5  Configure Security**

1. Click the Security tab.

2. Provide security parameters, as described in Table 5–6, depending on the channel/transport protocols selected in Task 1.

   Table 5–6 describes the security parameters (listed in alphabetical order) and the protocols to which the parameters apply.

   **Note:** The Digital Signature and Encryption lists are populated with the available certificates when the Key Store location is provided for the host trading partner. See Task 6, “Provide Key Store Information for the Host Trading Partner” for more information.
### Table 5–6 Security Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ack Signed</td>
<td>Select this option to ensure that the responder acknowledges receipt of the messages; nothing needs to be provided.</td>
<td>AS1 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AS2 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-2.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ebMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-V02.00 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RosettaNet-01.10 (optional)</td>
</tr>
<tr>
<td>Digital Signature</td>
<td>To use a digital signature certificate, the Key Store must have the corresponding private key.</td>
<td>AS1</td>
</tr>
<tr>
<td></td>
<td>If <strong>Message Signed</strong> is selected, then select the following for AS1:</td>
<td>AS2</td>
</tr>
<tr>
<td></td>
<td>SMIME 3.0 with SHA1 - RSA</td>
<td>ebMS-2.0</td>
</tr>
<tr>
<td></td>
<td>If <strong>Message Signed</strong> is selected, then select one of the following for AS2:</td>
<td>ebMS-1.0</td>
</tr>
<tr>
<td></td>
<td>SMIME 3.0 with MD5 - RSA</td>
<td>RosettaNet-V02.00</td>
</tr>
<tr>
<td></td>
<td>SMIME 3.0 with SHA1 - RSA</td>
<td>RosettaNet-01.10</td>
</tr>
<tr>
<td></td>
<td>If <strong>Message Signed</strong> is selected, then select one of the following for ebMS-2.0 and ebMS-1.0:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XMLDSIG with SHA1 - RSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XMLDSIG with SHA1 - DSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If <strong>Message Signed</strong> is selected, then select one of the following for RosettaNet-V02.00:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMIME 3.0 with MD5 - RSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMIME 3.0 with SHA1 - RSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMIME 2.0 with MD5 - RSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMIME 2.0 with SHA1 - RSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XMLDSIG with SHA1 - RSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XMLDSIG with SHA1 - DSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If <strong>Message Signed</strong> is selected, then select one of the following for RosettaNet-01.10:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMIME 3.0 with MD5 - RSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMIME 3.0 with SHA1 - RSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMIME 2.0 with MD5 - RSA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMIME 2.0 with SHA1 - RSA</td>
<td></td>
</tr>
</tbody>
</table>
3. Click Save.

Note: For AS1, B2B supports only SHA1 for signing. MD5 is not supported for AS1 signing.

5.5.1 About MLLP

An MLLP delivery channel is established by a two-way handshake between the server and client. It is always bidirectional, unlike other transports, and is used for both sending and receiving messages. An MLLP delivery channel is configured for the remote trading partner only, and is configured as a server socket or a client socket. As a server socket, the channel accepts connections on the specified port. As a client socket, the channel establishes a connection on the specified IP address and port. For either socket type, you specify a permanent or transient connection type. A permanent connection, once established, is cached and serves as a channel for the message exchange throughout the lifecycle of the endpoint. A transient connection serves as a channel only for exchanging one set of messages comprised of the business message and its acknowledgment. See Section 5.5.1.1, "Overriding the Connection Mode."

A recommended configuration is for the sender to configure the MLLP client delivery channel and for the receiver to configure the MLLP server channel. For example, if Acme wants to send an HL7, Custom, or positional flat file message to GlobalChips, Acme can have the client MLLP permanent channel and GlobalChips can have the
server MLLP permanent channel. MLLP connection types (permanent and transient) for the server and client must match (both permanent or both transient). However, in some cases the sender can have the server channel and receiver can have the client channel provided the connection is pre-established.

Because MLLP is a bidirectional channel, it is not considered to be a listening channel and the same MLLP delivery channel can be used for both sending and receiving messages.

Because MLLP operates in single delivery channel mode by default, simply select a delivery channel under the remote trading partner when creating an agreement. If operating in a non-single MLLP delivery channel mode is required, select a different MLLP delivery channel in the other agreements.

5.5.1.1 Overriding the Connection Mode
To override the connection mode for a message without changing the configuration manually, set the following properties:

For changing from a transient to a permanent connection:

`CONNMODE: Permanent`

For default integration:

`b2b.connMode: Permanent`

You can also change from a permanent connection to a transient connection.

5.5.1.2 Generic Support for TCP
MLLP uses SB (start byte), EB (end byte) and CR to interpret a message. To interpret a message using the length of the data or the start string and end string instead of SB and EB, Oracle B2B provides a generic solution for TCP.

For generic support for TCP, use with the following parameters on the Exchange Protocol Parameters tab (shown in Figure 5–21): Start Block, End Block, Header Length, Message Length Index, and Retain Header.

**Figure 5–21 Parameters for Generic TCP**
Configuring Channels

Note: When you create a generic TCP channel using the MLLP protocol, the parameters on the Exchange Protocol Parameters tab appear as shown in Figure 5–21. After creating the channel, two subtabs appear, with MLLP-specific and generic TCP-specific parameters grouped under them.

See Table 5–5, "Exchange Protocol Parameters" for descriptions of these parameters.

Table 5–7 describes how Oracle B2B processes messages using MLLP when data is sent or received using the parameters that support generic TCP.

<table>
<thead>
<tr>
<th>Generic TCP Solution</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send or receive data by specifying a start block and end block</td>
<td>Use the Start Block and End Block parameters available on the Exchange Protocol Parameters tab when you select MLLP-1.0 for a remote trading partner. See Table 5–5 for descriptions of the Start Block and End Block parameters. Example: <code>&lt;start block&gt;Data&lt;end block&gt;</code></td>
</tr>
<tr>
<td>Send or receive data by specifying a start block, end block, and data length</td>
<td>Use the Start Block, End Block, Message Length Index, and Header Length parameters available on the Exchange Protocol Parameters tab when you select MLLP-1.0 for a remote trading partner. See Table 5–5 for descriptions of the parameters. Example: <code>&lt;start block&gt;&lt;length&gt;Data&lt;end block&gt;</code></td>
</tr>
<tr>
<td>Send or receive data by specifying the data length</td>
<td>Use the Message Length Index and Header Length parameters available on the Exchange Protocol Parameters tab when you select MLLP-1.0 for a remote trading partner. See Table 5–5 for descriptions of the Message Length Index and Header Length parameters. Example: <code>&lt;length&gt;Data</code> Example: <code>&lt;length+header&gt;Data</code> That is, 15HDRDATADATADATA, where you configure: Message Length Index=1-2 Header length=5 15 is the length start after end index of Message Length Index. HDR is the header.</td>
</tr>
<tr>
<td>Send or receive data by specifying the start block and data size</td>
<td>Use the Start Block, Message Length Index, and Header Length parameters available on the Exchange Protocol Parameters tab when you select MLLP-1.0 for a remote trading partner. See Table 5–5 for descriptions of the Start Block, Message Length Index, and Header Length parameters. Note: In this case, the start block is part of the header and the minimum message length index must be more than the start block size. Example: <code>&lt;start block&gt;&lt;length&gt;Data</code></td>
</tr>
<tr>
<td>Retain the back-end application header and B2B will not add the start block, data size, and end block.</td>
<td>To send data to the trading partner without adding a header and retain the back-end application header, select the Retain Header property. See Table 5–5 for a description of Retain Header parameter.</td>
</tr>
</tbody>
</table>

5.5.1.3 Dynamic Endpoints

The dynamic IP feature of MLLP provides flexibility to dynamically change the endpoints associated with a delivery channel. This is done by overriding the IP address of the delivery channel through the `actionName/eventName` attribute in the message enqueue header.

For example:

```
eventName=DynamicIP:GlobalChips:IP_address:port_number
```
or

```
actionName=DynamicIP:GlobalChips:IP_address:port_number
```

This feature is also available in B2B composites (as part of the SOA Service Component Architecture (SCA) assembly model) using the following syntax:

```
b2b.toDynamicIP=GlobalChips:IP_address:port_number
```

The `b2b.toDynamicIP` property is set in a normalized message property that is sent to B2B.

Oracle B2B generates a unique control number for each message. For a broadcasting case involving multiple dynamic endpoints corresponding to the same trading partner, the back-end application must provide the control number. Oracle B2B stores and uses the dynamic endpoint details for correlation of the acknowledgment. No additional configuration is required.

### 5.5.1.4 Using a Transport Callout to Extract Custom Headers

To extract a custom header for outbound messages, add the `CUSTOM_HEADER` property in the `actionName` property from the back-end application. This property will be available in the callout as a `CUSTOM_HEADER` parameter of `CalloutMessage`. You can retrieve the property in the callout for your usage.

For example:

```
eventName= CUSTOM_HEADER:your_value
```

For default integration:

```
b2b.customHeader= your_value
```

To extract a custom header for inbound messages, set the `CUSTOM_HEADER` property as the `CalloutMessage` parameter in the callout. The property will be available as part of the `actionName` properties in the back-end application. See Example 12-1, "Setting and Getting the CUSTOM_HEADER Property" for details.

### 5.5.1.5 Message Sequencing in MLLP Exchanges

Exchanging messages in sequence can be challenging in a multi-threaded system, because the first message produced may not necessarily arrive at the destination first. For enterprises with this business requirement, B2B provides a sequencer and a dispatcher. The sequencer sequences a message based on arrival time. The dispatcher dispatches the sequenced message. Message sequencing is available for outbound and inbound directions.

#### 5.5.1.5.1 Outbound Message Sequencing

To enable sequencing for an outbound message, for AQ delivery channels, enqueue the message by setting `ACTION_NAME` to `TARGET:sequence_target_name` when assigning the `jca.aq.HeaderDocument`. However, when using the `ENQUEUE` utility that is provided with the `b2b.jar`, set `eventName` (not `ACTION_NAME`) to `TARGET:sequence_target_name`; for example, `eventName=TARGET:sequence1`.

To enable sequencing when using the default channel, use `b2b.sequencingTarget = sequence_target_name`.

To dispatch the sequenced message, configure the `Outbound Dispatcher Count` parameter, shown in Figure 5–22.
By default, the value is 0, which is the setting for sequencing without dispatching (stacking). Depending on the message load, set Outbound Dispatcher Count to the appropriate value.

5.5.1.5.2 Inbound Message Sequencing

To enable sequencing for an inbound message, select the Sequence flag for the MLLP delivery channel, as shown in Figure 5–23.

To dispatch the sequenced message, configure the Inbound Dispatcher Count parameter, as shown in Figure 5–22.

5.5.1.5.3 Sequencing Without Dispatching

Trading partner downtime is typically handled by stacking messages in the back-end application, which requires the entire message processing in B2B after the downtime. This leads to underutilizing the B2B application during downtime and overloading when the trading partner comes up. This affects the regular message flow, because there is a surge in message processing.

Upon trading partner delivery failure, the corresponding messages are marked not to be picked up by the dispatcher, resulting in stacking the messages in B2B instead of the back-end application. To process the messages, set the following properties:

- Auto Stack Handler = true
- Auto Stack Handler Interval = interval (in seconds)

The Auto Stack Handler and Auto Stack Handler Interval parameters are shown in Figure 5–22. When set to true, the stacked message are eligible for delivery by the dispatcher during an appropriate interval. It is also possible to specify the variable interval with a comma-separated value to Auto Stack Handler Interval.
5.6 Using the Auto Create Agreement Feature

In the Partner area, shown in Figure 5–24, you can use the Auto Create Agreement icon to create an agreement for a remote trading partner.

Figure 5–24 The Auto Create Agreement Feature

This feature creates one agreement for each document definition associated with the selected remote trading partner. You can further customize the agreement on the Agreement tab. See Chapter 6, "Creating and Deploying Trading Partner Agreements," for more information about the Agreement tab.

5.7 Using Identifiers for Trading Partner Lookup

Identifiers available in design-time data are used to look up trading partners. Identifiers do not need to be part of a deployed, active agreement. The appropriate document and exchange identifiers are used for lookup; for example:

- For the AS2-1.1 exchange protocol, the AS2 identifier is used.
- For the EDI X12 document protocol, the Sender Group ID and Sender Interchange ID are used.
The final steps in the Oracle B2B process flow, shown in Figure 6–1, are to create and deploy the agreement.

Figure 6–1  Oracle B2B Process Flow

A trading partner agreement defines the terms that enable two trading partners, the initiator and the responder, to exchange business documents. It identifies the trading partners, trading partner identifiers, document definitions, and channels.

This chapter contains the following topics:

- Section 6.1, "Introduction to Agreements"
- Section 6.2, "Creating an Agreement"
- Section 6.3, "Deploying an Agreement"
- Section 6.4, "Deleting and Exporting Agreements"

See the following for more information:

- Chapter 8, "Managing Deployments," for how to export agreements and manage deployment states
- Chapter 10, "Importing and Exporting Data," for how to export agreements

6.1 Introduction to Agreements

An agreement consists of two trading partners—the host trading partner and one remote trading partner, and represents one type of business transaction between those partners. For example, if Acme and GlobalChips participate in both EDIFACT and RosettaNet exchanges with each other, you create an agreement for each of the exchanges. Some exchanges are bidirectional, requiring an agreement for each direction.
For example, if Acme sends a sales order to GlobalChips using a Custom document sent using the Generic File protocol, you create an agreement for the outbound direction, where Acme sends the order, and for the inbound direction, where Acme is the receiver. A change to a component of an agreement (for example, a change to the document definition) is effective automatically in the agreement.

Creating an agreement is the last step in the design of a B2B transaction. Before you create an agreement, you must have already created the document definitions and configured the trading partners. See Chapter 4, "Creating Document Definitions," and Chapter 5, "Configuring Trading Partners," for more information.

6.2 Creating an Agreement

Figure 6–2 shows the Oracle B2B interface for working with agreements. Click a remote trading partner name to see its agreements with the host trading partner.

Figure 6–3 shows the steps to create an agreement.
Creating an Agreement

Creating and Deploying Trading Partner Agreements

6-3

Figure 6–3  Steps to Creating an Agreement (Workflow Overview)

Step 1: Identify the remote trading partner
The host trading partner is automatically included in an agreement, so you need only identify the remote trading partner. You can do this in two ways: select the partner from the Partners region before adding the agreement, or select the host trading partner, click Add in the Agreements region, and click the Select Partner icon in the New Agreement region.

Step 2: Select the document definition
The document definition is selected for the host trading partner, as reflected in the Select Document Definition dialog, shown in figure Figure 6–4.

Figure 6–4  Selecting the Document Definition

<table>
<thead>
<tr>
<th>Select Document Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partner</td>
</tr>
<tr>
<td>Acme → GlobalChips</td>
</tr>
<tr>
<td>Acme → GlobalChips</td>
</tr>
<tr>
<td>Acme → GlobalChips</td>
</tr>
<tr>
<td>Acme → GlobalChips</td>
</tr>
</tbody>
</table>

For an exchange in which you need both outbound and inbound agreements, do the following:

- For the outbound agreement, select the document definition in which the host trading partner is the sender (Acme → GlobalChips in Figure 6–4)
- For the inbound agreement, select the document definition in which the host trading partner is the receiver (Acme ← GlobalChips in Figure 6–4)

Step 3: Provide the agreement ID and name
Provide any agreement identifier and agreement name. These fields can have the same value if you need only one for tracking purposes.
Step 4: Select validation, translation, and functional acknowledgment options

Table 6–1 describes the validation, translation, and functional acknowledgments available when you create an agreement.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validate</td>
<td>Select to enable validation of the document against the configured ECS file.</td>
</tr>
<tr>
<td>Translate</td>
<td>Select to enable the translation of XML to native format and vice versa (for EDI and HL7, for example). If Translate is not selected (no translation), then B2B cannot correlate the business message with the functional acknowledgment, irrespective of the value of the B2B Handle FA property. See Section B.1, &quot;Properties To Set in Fusion Middleware Control,&quot; for information about the property.</td>
</tr>
<tr>
<td>Functional Ack</td>
<td>Select to enable the functional acknowledgment for success or error criterion.</td>
</tr>
</tbody>
</table>

Step 5: Select the channel for the remote trading partner

A list of channels that you created when you set up the remote trading partner is available. (Listening channels are not part of an agreement.)

Step 6: Add identifiers

Identifier types for the host and remote trading partners are listed. Select the identifiers that apply to this agreement. You can shift-click to select multiple identifiers.

For outbound agreements, use the identifier types listed in Table 6–2 with the exchange protocols.

<table>
<thead>
<tr>
<th>Exchange Protocol</th>
<th>Identifier Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic File-1.0</td>
<td>Name</td>
</tr>
<tr>
<td>Generic FTP-1.0</td>
<td>Name</td>
</tr>
<tr>
<td>Generic SFTP-1.0</td>
<td>Name</td>
</tr>
<tr>
<td>Generic AQ-1.0</td>
<td>Name</td>
</tr>
<tr>
<td>Generic JMS-1.0</td>
<td>Name</td>
</tr>
<tr>
<td>AS2-1.1</td>
<td>Name, AS2 Identifier</td>
</tr>
<tr>
<td>AS1-1.0</td>
<td>Name, AS1 Identifier</td>
</tr>
<tr>
<td>ebMS-1.0, ebMS-2.0</td>
<td>Name, ebMS Identifier</td>
</tr>
<tr>
<td>RosettaNet-V02.00</td>
<td>Name, DUNS</td>
</tr>
<tr>
<td>MLLP exchange</td>
<td>Name, MLLP ID</td>
</tr>
<tr>
<td>Generic HTTP-1.0</td>
<td>Name, Generic Identifier</td>
</tr>
<tr>
<td>Generic Email-1.0</td>
<td>Name, Generic Identifier</td>
</tr>
</tbody>
</table>

See Chapter 9, "Creating Types," for more information about identifier types.
Step 7: Save and validate the agreement
Clicking **Save** also validates the agreement.

**To create an agreement:**
1. Click the **Partners** tab.
2. In the **Agreements** region, click **Add**.
3. Click **Select Partner**.
4. Select a remote trading partner.
5. Click **Select Document Definition**.
6. Select a document definition for the initiator.
7. Provide an agreement ID and name.
8. Select from the validation, translation, and functional acknowledgment options, as described in Table 6–1.
9. Provide an optional description, a callout (if previously created), and start and end dates.
   Use callouts to transform the formats of messages exchanged between remote and host trading partners. See Chapter 12, "Managing Callouts."
   An agreement cannot be deployed after an end date entered here because the agreement will have expired.
10. For the host trading partner, click **Add** and select identifiers.
11. For the remote trading partner, select a channel.
12. In the remote trading partner, click **Add** and select identifiers.
13. Click **Save**.

After you create an agreement, it is ready to be deployed. It is listed on the **Administration > Deploy** page. See Section 6.3, "Deploying an Agreement," to continue.

### 6.3 Deploying an Agreement

Deployment is the process of activating an agreement from the design-time repository to the run-time repository.

After deploying an agreement, use the **Manage Deployments** tab and the **Reports** tab. See the following for more information:

- Chapter 8, "Managing Deployments"
- Chapter 16, "Creating Reports"

After you create, save, and validate an agreement, you can deploy it as follows:

- From the same page (**Partners > Agreement** tab), using the **Deploy** button (see Figure 6–2)
- From the **Administration > Deploy** page, as shown in Figure 6–5. Use this option to select multiple agreements to deploy at the same time.
Deploying an Agreement

To deploy an agreement from the Deploy tab:
1. Click the Administration tab.
2. Click the Deploy tab.
3. Use the search parameters to find the agreement you want to deploy and click Search.
4. Highlight one or more agreements and click Deploy.

6.3.1 Redeploying an Agreement

If you deploy a previously deployed agreement, the first version is moved to an inactive state and the most recently deployed agreement is active.
6.4 Deleting and Exporting Agreements

Only agreements in the draft state can be deleted. Purging an agreement returns its status to the draft state. Agreements that have deployed versions in active, inactive, or retired states cannot be deleted.

An agreement can be exported to a ZIP file by using the Export button on the Agreement tab.
This part describes how to use Oracle B2B administration features.

This part contains the following chapters:

- Chapter 7, "Using Document Protocols"
- Chapter 8, "Managing Deployments"
- Chapter 9, "Creating Types"
- Chapter 10, "Importing and Exporting Data"
- Chapter 11, "Batching EDI Messages"
- Chapter 12, "Managing Callouts"
- Chapter 13, "Purging Data"
- Chapter 14, "Configuring Listening Channels"
- Chapter 15, "Configuring B2B System Parameters"
A document protocol defines the document type of the message payload. Oracle B2B document protocols are shown in Figure 7–1.

Figure 7–1  Oracle B2B Document Protocols

You can define nearly any protocol by using the Custom protocol and the many guideline documents in Oracle B2B Document Editor.

This chapter contains the following topics:

- Section 7.1, "Using the Custom Document Protocol"
- Section 7.2, "Using the EDI EDIFACT Document Protocol"
- Section 7.3, "Using the EDI X12 Document Protocol"
- Section 7.4, "Using the HL7 Document Protocol"
- Section 7.5, "Using the Positional Flat File Document Protocol"
- Section 7.6, "Using the RosettaNet Document Protocol"
- Section 7.7, "Using the UCCNet Document Protocol"
- Section 7.8, "Changing Document Details"
- Section 7.9, "Using Document Routing IDs"

For related information, see the following:

- Chapter 3, "Creating Guideline Files"
- Chapter 4, "Creating Document Definitions"
7.1 Using the Custom Document Protocol

Oracle B2B supports custom document protocols to create documents needed for proprietary transactions. With XML messages, you have the advantage of schema enforcement (XSDs).

When creating a Custom document, you specify rules to identify the incoming document. For XML documents, specify an XPath expression and a value, which is the expected result of the expression.

Document Version Parameters
No parameters need to be set when you create the document version for a Custom document.

Document Type Parameters
When you create a Custom document type, you can set ebXML messaging service (ebMS) parameters to identify the ebXML document. Figure 7–2 shows the document type parameters for a Custom document.

Figure 7–2 Document Type Parameters for a Custom Document

Table 7–1 describes the document type parameters for a Custom document.

Table 7–1 Document Type Parameters for a Custom Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ebMS Tab</td>
<td></td>
</tr>
<tr>
<td>Action name</td>
<td>The action name for the ebXML header, which is also an identification criteria for inbound and outbound messages. ebMS documents require an action name to avoid run-time errors.</td>
</tr>
<tr>
<td>Service name</td>
<td>The service name for the ebXML header, which is also an identification criteria for inbound messages. ebMS documents require a service name to avoid run-time errors.</td>
</tr>
</tbody>
</table>
Table 7–1 (Cont.) Document Type Parameters for a Custom Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service type</td>
<td>The service type for the ebXML header, which is also an identification criteria for inbound messages. ebMS documents require a service type to avoid run-time errors.</td>
</tr>
<tr>
<td>From Role</td>
<td>The trading partner that sends the message. A value provided here overrides the Identifiers values supplied on the Profile tab.</td>
</tr>
<tr>
<td>To Role</td>
<td>The trading partner that receives the message. A value provided here overrides the Identifiers values supplied on the Profile tab.</td>
</tr>
</tbody>
</table>

**Document Definition Parameters**

When you create a Custom document definition, select the file type—XML or Flat—and set parameters in the tabbed areas. *Figure 7–3* shows the document definition parameters for an XML-type Custom document.

*Figure 7–3* Document Definition Parameters for an XML-Type Custom Document

![Document Definition](image)

Custom-1.0-PurchaseOrder-PurchaseOrder_Def

Enter the document definition name and select the required definition file.

- **Document Definition Name**: PurchaseOrder_Def
- **Description**: 

  ![Description](image)

  **Definition**: Custom_PurchaseOrder.xsd

  ![Definition](image)

- **Root XSD Name**: 

  ![Root XSD Name](image)

- **Identification Type**: XML

  ![Identification Type](image)

- **Identification Expression (XPath)**: /"local-name()=PurchaseOrder"

  ![Identification Expression](image)

- **Identification Value**: 

  ![Identification Value](image)

- **DTD/XSD NamespaceConversion**: None

  ![DTD/XSD NamespaceConversion](image)

*Figure 7–4* shows the document definition parameters for a flat-file Custom document.

*Figure 7–4* Document Definition Parameters for a Flat-Type Custom Document

![Flat-Type Custom Document](image)
Figure 7–4  Document Definition Parameters for a Flat-File Custom Document

Table 7–2 describes the document definition parameters for a Custom document.

Table 7–2  Document Definition Parameters for a Custom Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XML Tab</strong></td>
<td>(Available if XML is selected from Identification Type)</td>
</tr>
<tr>
<td>Identification Expression (XPath)</td>
<td>Locates a node in the XML payload</td>
</tr>
<tr>
<td>Identification Value</td>
<td>Provides the value to match in the node identified by the Identification Expression. If the values match, then the document is successfully identified. If the value is left blank, then Oracle B2B checks for the existence of the node and the document is successfully identified.</td>
</tr>
<tr>
<td>DTD/XSD NamespaceConversion</td>
<td>Select from None, Both, Inbound, or Outbound.</td>
</tr>
<tr>
<td><strong>Routing Tab</strong></td>
<td>-</td>
</tr>
<tr>
<td>Document Routing ID</td>
<td>Sets the consumer name to the back-end application</td>
</tr>
<tr>
<td><strong>Flat Tab</strong></td>
<td>-</td>
</tr>
<tr>
<td>Identification Start Position</td>
<td>Used in combination with the end position to retrieve a value from the payload between the start and end positions</td>
</tr>
<tr>
<td>Identification End Position</td>
<td>Used in combination with the start position to retrieve a value from the payload between the start and end positions</td>
</tr>
<tr>
<td>Identification Value</td>
<td>A value between the start and end positions</td>
</tr>
</tbody>
</table>

7.1.1 How to Configure the XPath Expression for a Custom XML Document

The XPath expression identifies a Custom XML document. You configure the XPath expression when you specify the document type parameters.

The options when configuring an XPath expression are as follows:
Using the Custom Document Protocol

- Option 1: Specify the XPath and the Matching Value
- Option 2: Check for the Existence of a Node
- Option 3: Check the Value of an Attribute

### 7.1.1 Option 1: Specify the XPath and the Matching Value

Assume that the transaction ID is 12345. Set the parameters as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification Value</td>
<td>12345</td>
</tr>
<tr>
<td>Identification Expression</td>
<td>//*[local-name() = 'TransactionID']/text()</td>
</tr>
</tbody>
</table>

Oracle B2B compares the value of **Identification Expression** in the payload to the value specified in **Identification Value**. If the values match, then the document is identified successfully and the corresponding document type and document protocol version are used to identify the agreement. **Example 7-1** shows an excerpt of the XML payload for this option.

**Example 7-1 Specify the XPath and the Matching Value**

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Message xmlns:ns1="http://www.example1.org" xmlns:ns2="http://www.example2.org"
  xmlns="http://www.example3.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns="http://www.example4.org">
  <MessageHeader>
    <Source>201944019</Source>
    <Destination>205704856</Destination>
    <TransactionID>123456</TransactionID>
    <Version>1-0-0</Version>
  </MessageHeader>
  <Body>
    <ns:Case xsi:schemaLocation="http://www.example4.org" ns:caseCategoryID="1">
      <ns1:OfficialProvisionNumber>String</ns1:OfficialProvisionNumber>
    </ns:Case>
  </Body>
</Message>
```

### 7.1.1.2 Option 2: Check for the Existence of a Node

Assume that you are checking for the existence of a node called *registerCommand*. Set the parameters as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification Value</td>
<td>Leave blank.</td>
</tr>
<tr>
<td>Identification Expression</td>
<td>/<em>[local-name() = 'envelope']/body/transaction/command/</em>[local-name() = 'registerCommand']</td>
</tr>
</tbody>
</table>

When the **Identification Value** field is left blank, Oracle B2B checks for the node identified in **Identification Expression**. If a node in the payload matches, then the document is identified successfully. **Example 7-2** shows an excerpt of the XML payload for this option.
Example 7–2  Check for the Existence of a Node

```
<uccnet:envelope xmlns:eanucc="http://www.ean-ucc.org/schemas/1.3/eanucc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:uccnet="http://www.uccnet.org/schemas/2.2/uccnet"
    communicationVersion="2.2"
    xsi:schemaLocation="http://www.uccnet.org/schemas/2.2/uccnet
    http://www.testregistry.net/xmlschema/uccnet/2.2/Envelope.xsd">
    <messageHeader>
        <messageIdentifier>
            <value>791:1_EB3CDC749A1F2B2ABE03014906CC4605A</value>
        </messageIdentifier>
        <userId>oraclesupXSD</userId>
        <representingParty>
            <gin>0060974050142</gin>
        </representingParty>
    </messageHeader>
    <body>
        <transaction>
            <entityIdentification>
                <uniqueCreatorIdentification>856</uniqueCreatorIdentification>
                <globalLocationNumber>
                    <gin>0060974050142</gin>
                </globalLocationNumber>
            </entityIdentification>
            <command>
                <uccnet:registerCommand>
                    <registerCommandHeader type="ADD" />
                </uccnet:registerCommand>
            </command>
        </transaction>
    </body>
</uccnet:envelope>
```

7.1.1.3  Option 3: Check the Value of an Attribute

Assume that the value of the country attribute is **US**. Set the parameters as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification Value</td>
<td>US</td>
</tr>
<tr>
<td>Identification Expression</td>
<td>//*/@country</td>
</tr>
</tbody>
</table>

Oracle B2B compares the value of the country attribute to the value set for Identification Value. If the values match, then the document is identified successfully. Example 7–3 shows an excerpt of the XML payload for this option.

Example 7–3  Check the Value of an Attribute

```
<?xml version="1.0" encoding="windows-1252" ?>
<MyAddress country="US" xmlns="http://www.example.org"
    xsi:schemaLocation="PO.xsd">
    <name>B2B Buyer</name>
    <street>100 Oracle Parkway</street>
    <city>Redwood City</city>
    <state>CA</state>
    <zip>94065</zip>
</MyAddress>
```
7.2 Using the EDI EDIFACT Document Protocol

Oracle B2B supports message exchanges using UN/EDIFACT, the United Nations Electronic Data Interchange for Administration, Commerce and Transport. These standards prescribe the formats, character sets, and data elements used in purchase orders and invoices.

Oracle B2B supports all versions and document types of EDI EDIFACT. Table 7–3 lists a few of the transaction sets supported in Oracle B2B.

<table>
<thead>
<tr>
<th>Set</th>
<th>Description</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDERS</td>
<td>Purchase Order Message</td>
<td>D98A</td>
</tr>
<tr>
<td>ORDRSP</td>
<td>Purchase Order Response Message</td>
<td>D98A</td>
</tr>
<tr>
<td>CONTRL</td>
<td>Syntax and Service Report Message</td>
<td>D3</td>
</tr>
</tbody>
</table>

For information about the organization that created and maintains the UN/EDIFACT standards, go to http://www.unece.org

Document Version Parameters

When you create an EDI EDIFACT document version, you can set various parameters. Figure 7–5 shows document version parameters for an EDI EDIFACT document.

Table 7–4 describes the document version parameters for an EDI EDIFACT document.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interchange Tab</td>
<td>-</td>
</tr>
<tr>
<td>Create UNA</td>
<td>Select from <strong>always</strong>, <strong>never</strong>, or <strong>delimiter-based</strong>. If <strong>delimiter-based</strong> is selected, then UNA is created if the specified delimiters are different from the EDIFACT default value. The <strong>Never</strong> option does not generate UNA for outbound EDIFACT documents, even if nondefault delimiters are used. The <strong>Never</strong> option for inbound messages cannot work for B2B if an EDIFACT document is received without UNA and with nondefault delimiters.</td>
</tr>
<tr>
<td>Syntax Identifier</td>
<td>Coded identification of the agency controlling syntax and syntax level used in an interchange. EDI position UNB 010 010 S001 0001. The value <strong>UNOB</strong> is supplied.</td>
</tr>
<tr>
<td>Syntax Version Number</td>
<td>Version number of the syntax identified in the syntax identifier (0001). EDI position UNB 010 020 S001 0002. The value <strong>1</strong> is supplied.</td>
</tr>
<tr>
<td>Service Code List Directory Version Number</td>
<td>Version number of the service code list directory. EDI position UNB 010 030 S001 0030.</td>
</tr>
<tr>
<td>Character Encoding</td>
<td>Coded identification of the character encoding used in the interchange. To be used as specified in the partners’ interchange agreement, for the purpose of identifying the character repertoire encoding technique used in the interchange (when the default encoding defined by the character repertoire’s associated character set specification is not used). EDI position UNB 010 040 S001 0133.</td>
</tr>
<tr>
<td>Interchange Date</td>
<td>Local date when an interchange or a group was prepared. EDI position UNB 030 010 S004 0017. The value <strong>#SystemDate(YYMMDD)#</strong> is supplied.</td>
</tr>
<tr>
<td>Interchange Time</td>
<td>Local time of day when an interchange or a group was prepared. EDI position UNB 030 020 S004 0019. The value <strong>#SystemTime(HHMM)#</strong> is supplied.</td>
</tr>
<tr>
<td>Recipient’s Reference/Password</td>
<td>Reference or password to the recipient’s system or to a third-party network as specified in the partners’ interchange agreement. To be used as specified in the partners’ interchange agreement. It may be qualified by data element 0025. EDI position UNB 060 010 S005 0022.</td>
</tr>
<tr>
<td>Recipient’s Reference/Password Qualifier</td>
<td>Qualifier for the recipient’s reference or password. To be used as specified in the partners’ interchange agreement. EDI position UNB 060 020 S005 0025.</td>
</tr>
<tr>
<td>Application Reference</td>
<td>Identification of the application area assigned by the sender, to which the messages in the interchange relate; for example, the message type, if all the messages in the interchange are of the same type. Identification of the application area (for example, accounting, purchasing) or of the message type, as applicable. EDI position UNB 070.</td>
</tr>
<tr>
<td>Processing Priority Code</td>
<td>Code determined by the sender requesting processing priority for the interchange. To be used as specified in the partners’ interchange agreement. EDI position UNB 080.</td>
</tr>
<tr>
<td>Interchange Agreement Identifier</td>
<td>Identification by name or code of the type of agreement under which the interchange takes place. Name or code to be specified in the partners’ interchange agreement. EDI position UNB 100.</td>
</tr>
<tr>
<td>Test Indicator</td>
<td>Indication that the structural level containing the test indicator is a test. EDI position UNB 110.</td>
</tr>
<tr>
<td>Interchange ecs File</td>
<td>Use the <strong>Browse</strong> button to find an ecs file to override the standard file. If not provided, the B2B-provided default file (interchange ecs file of the syntax version number, UNB 010 020) is used.</td>
</tr>
<tr>
<td>Group Tab</td>
<td>-</td>
</tr>
<tr>
<td>Create Functional Group</td>
<td>Indication of function group (UNG) creation. The value <strong>TRUE</strong> is supplied.</td>
</tr>
<tr>
<td>Date of Group Preparation</td>
<td>Local date when an interchange or a group was prepared. EDI position UNG 040 010. The system date stamp is supplied.</td>
</tr>
<tr>
<td>Time of Group Preparation</td>
<td>Local time of day when an interchange or a group was prepared. EDI position UNG 040 020. The system time stamp is supplied.</td>
</tr>
</tbody>
</table>
Using the EDI EDIFACT Document Protocol

Document Type Parameters
When you create an EDI EDIFACT document type, you can set various parameters. Figure 7–6 shows the document type parameters for an EDI EDIFACT document.

Table 7–4 (Cont.) Document Version Parameters for an EDI EDIFACT Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlling Agency</td>
<td>Code identifying a controlling agency. EDI position UNG 070 010. The value UN is supplied.</td>
</tr>
<tr>
<td>Group Association Assigned Code</td>
<td>Code assigned by the association responsible for the design and maintenance of the message type concerned that further identifies the message. EDI position UNG 070 030.</td>
</tr>
<tr>
<td>Application Password</td>
<td>Password to the recipient's division, department or sectional application system/process. EDI position UNG 080.</td>
</tr>
<tr>
<td>Group ecs File</td>
<td>Use the Browse button to find an ecs file to override the standard file. If not provided, the B2B-provided default file is used.</td>
</tr>
<tr>
<td>Delimiters Tab</td>
<td>A delimiter is characterized by two levels of separators and a terminator assigned by the sender. Delimiters are also called service characters, data delimiters, or message delimiters. They are specified in the interchange header and cannot be used in a data element value elsewhere in the interchange. In an EDI file, the segment delimiter, the element delimiter, and the subelement delimiter are used. Note: Click Select Hexadecimal Characters next to any of the delimiter fields to provide values.</td>
</tr>
<tr>
<td>Segment Delimiter</td>
<td>EDIFACT segment delimiter. The value 0x27 is supplied.</td>
</tr>
<tr>
<td>Element Delimiter</td>
<td>EDIFACT element delimiter. The value 0x2b is supplied.</td>
</tr>
<tr>
<td>Subelement Delimiter</td>
<td>EDIFACT subelement delimiter. The value 0x3a is supplied.</td>
</tr>
<tr>
<td>Decimal Separator</td>
<td>EDIFACT decimal separator. The value 0x2e is supplied.</td>
</tr>
<tr>
<td>Release Character</td>
<td>EDIFACT release character. The value 0x3f is supplied.</td>
</tr>
<tr>
<td>Replacement Character</td>
<td>EDIFACT replacement character. The value 0x7c is supplied.</td>
</tr>
<tr>
<td>Repeating Separator</td>
<td>EDIFACT repeating separator. The value 0x2a is supplied.</td>
</tr>
<tr>
<td>Miscellaneous Tab</td>
<td>-</td>
</tr>
<tr>
<td>Check Duplicate Control Number</td>
<td>When this property is selected (set to true), messages with duplicate interchange control numbers are rejected, meaning that the state of the incoming message is set to ERROR.</td>
</tr>
<tr>
<td>Ignore Envelope Parameters</td>
<td>Use this option to provide a list of envelope elements, separated by commas, to be ignored during look-up validation. The possible values depend on the identifiers used in the agreement. Possible values include InterchangeSenderID, InterchangeReceiverID, GroupReceiverID, GroupSenderID, TransactionAssociationAssignedCode, InterchangeReceiverQual, InterchangeSenderQual, and InterchangeControlVersion.</td>
</tr>
</tbody>
</table>
**Figure 7–6 Document Type Parameters for an EDI EDIFACT Document**

Table 7–5 describes the document type parameters for an EDI EDIFACT document.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transaction Tab</strong></td>
<td></td>
</tr>
<tr>
<td>Controlling Agency</td>
<td>Code identifying the agency controlling the specification, maintenance and publication of the message type. EDI position UNH 020 040 0051.</td>
</tr>
<tr>
<td>Transaction Association Assigned Code</td>
<td>Code, assigned by the association responsible for the design and maintenance of the message type concerned, which further identifies the message. EDI position UNH 020 050 0059.</td>
</tr>
<tr>
<td>Common Access Reference</td>
<td>Reference serving as a key to relate all subsequent transfers of data to the same business case or file. EDI position UNH 030 0068.</td>
</tr>
</tbody>
</table>

**Document Definition Parameters**

When you create an EDI EDIFACT document definition, you can set various parameters. Figure 7–7 shows document definition parameters for an EDI EDIFACT document.
Using the EDI X12 Document Protocol

7.3 Using the EDI X12 Document Protocol

Oracle B2B supports message exchanges using American National Standards Institute (ANSI) X12. These standards prescribe the formats, character sets, and data elements used in documents such as purchase orders and invoices.

Oracle B2B supports all versions and document types of EDI X12. Table 7–7 lists a few of the transaction sets supported in Oracle B2B.

<table>
<thead>
<tr>
<th>Set</th>
<th>Description</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>850</td>
<td>Purchase Order</td>
<td>4010</td>
</tr>
<tr>
<td>855</td>
<td>Purchase Order Acknowledgment</td>
<td>4010</td>
</tr>
<tr>
<td>997</td>
<td>Functional Acknowledgment</td>
<td>4010</td>
</tr>
</tbody>
</table>

For information about the organization that created and maintains the ANSI X12 standards, go to

http://www.ansi.org
Document Version Parameters

When you create an EDI X12 document version, you can set various parameters. Figure 7–8 shows document version parameters for an EDI X12 document.

Figure 7–8 Document Version Parameters for an EDI X12 Document

Table 7–8 describes the document version parameters for an EDI X12 document.

Table 7–8 Document Version Parameters for an EDI X12 Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interchange Tab</td>
<td>-</td>
</tr>
<tr>
<td>Authorization Information Qualifer</td>
<td>Code to identify the type of information in the authorization information. EDI position ISA 01. The value 00 is supplied.</td>
</tr>
<tr>
<td>Security Information Qualifer</td>
<td>Code to identify the type of information in the security information. EDI position ISA 03. The value 00 is supplied.</td>
</tr>
<tr>
<td>Interchange Date</td>
<td>Date of the interchange. EDI position ISA 09. The system date stamp is supplied (#SystemDate(YYMMDD)#).</td>
</tr>
<tr>
<td>Interchange Time</td>
<td>Time of the interchange. EDI position ISA 10. The system time stamp is supplied (#SystemTime(HHMM)#).</td>
</tr>
<tr>
<td>Interchange Control Standard/Repetition Separator</td>
<td>Code to identify the agency responsible for the control standard used by the message that is enclosed by the interchange header and trailer. EDI position ISA 11. The value U is supplied.</td>
</tr>
<tr>
<td>*Interchange Control Version Number</td>
<td>Code specifying the version number of the interchange control segments. EDI position ISA 12. The value 00401 is supplied.</td>
</tr>
<tr>
<td>Usage Indicator</td>
<td>Code to indicate whether data enclosed by this interchange envelope is in test or production. EDI position ISA 15. The value P, for production, is supplied.</td>
</tr>
</tbody>
</table>
Using the EDI X12 Document Protocol

Table 7–8  (Cont.) Document Version Parameters for an EDI X12 Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interchange ecs File</td>
<td>Use the <strong>Browse</strong> button to find an ecs file to override the standard file. If not provided, the B2B-provided default file (interchange ecs file of the interchange control version, ISA 12) is used.</td>
</tr>
<tr>
<td><strong>Group Tab</strong></td>
<td></td>
</tr>
<tr>
<td>Functional Group Date</td>
<td>Date sender generated a functional group of transaction sets. EDI position GS 04. The system date stamp is supplied (<strong>#SystemDate(CCYYMMDD)#</strong>).</td>
</tr>
<tr>
<td>Functional Group Time</td>
<td>Time when the sender generated a functional group of transaction sets (local time at sender's location). EDI position GS 05. The system time stamp is supplied (<strong>#SystemTime(HHMM)#</strong>).</td>
</tr>
<tr>
<td>Responsible Agency Code</td>
<td>Code used in conjunction with data element 480 to identify the issuer of the standard. EDI position GS 06. The value X is supplied.</td>
</tr>
<tr>
<td>Version/Release/Industry Identifier Code</td>
<td>Code indicating the version, release, subrelease, and industry identifier of the EDI standard being used, including the GS and GE segments; if the code in DE455 in GS segment is X, then in DE 480 positions 1-3 are the version number; positions 4-6 are the release and subrelease, level of the version; and positions 7-12 are the industry or trade association identifiers (optionally assigned by user); if the code in DE455 in GS segment is T, then other formats are allowed.</td>
</tr>
<tr>
<td>Group ecs File</td>
<td>Use the <strong>Browse</strong> button to find an ecs file to override the standard file. If not provided, the B2B-provided default file (group ecs file of EDI X12 version) is used.</td>
</tr>
<tr>
<td><strong>Delimiters Tab</strong></td>
<td></td>
</tr>
<tr>
<td>Segment Delimiter</td>
<td>The value <strong>0x7e</strong> is supplied.</td>
</tr>
<tr>
<td>Element Delimiter</td>
<td>The value <strong>0x2a</strong> is supplied.</td>
</tr>
<tr>
<td>Subelement Delimiter</td>
<td>The value <strong>0x5c</strong> is supplied.</td>
</tr>
<tr>
<td>Decimal Separator</td>
<td>The value <strong>0x2e</strong> is supplied.</td>
</tr>
<tr>
<td>Replacement Character</td>
<td>The value <strong>0x7c</strong> is supplied.</td>
</tr>
<tr>
<td>Repeating Separator</td>
<td>The value <strong>0x5e</strong> is supplied.</td>
</tr>
<tr>
<td><strong>Miscellaneous Tab</strong></td>
<td></td>
</tr>
<tr>
<td>Check Duplicate Control Number</td>
<td>When this property is selected (set to true), messages with duplicate interchange control numbers are rejected, meaning that the state of the incoming message is set to ERROR.</td>
</tr>
<tr>
<td>Ignore Envelope Parameters</td>
<td>Use this option to provide a list of envelope elements, separated by commas, to be ignored during look-up validation. The possible values depend on the identifiers used in the agreement. Possible values include InterchangeSenderID, InterchangeReceiverID, GroupReceiverID, GroupSenderID, TransactionAssociationAssignedCode, InterchangeReceiverQual, InterchangeSenderQual, and InterchangeControlVersion.</td>
</tr>
</tbody>
</table>

**Document Type Parameters**

When you create an EDI X12 document type, you can set various parameters. **Figure 7–9** shows the document type parameters for an EDI X12 document.
Using the EDI X12 Document Protocol

Figure 7–9   Document Type Parameters for an EDI X12 Document

Table 7–9 describes the document type parameters for an EDI X12 document.

Table 7–9   Document Type Parameters for an EDI X12 Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Tab</td>
<td></td>
</tr>
<tr>
<td>*Functional Group Identifier Code</td>
<td>Uniquely identifies a transaction set GS 01. Required.</td>
</tr>
<tr>
<td>Implementation Convention Reference</td>
<td>Reference assigned to identify Implementation Convention. EDI position ST 03.</td>
</tr>
<tr>
<td>Transaction Purpose Code</td>
<td>Code identifying the purpose of the transaction set. EDI position BEG/BGN 01.</td>
</tr>
</tbody>
</table>

Document Definition Parameters
When you create an EDI X12 document definition, you can set various parameters. Figure 7–10 shows document definition parameters for an EDI X12 document.
7.4 Using the HL7 Document Protocol

Oracle B2B implements the Health Level 7 (HL7) version 2.x and version 3 standards (version 3 supports Custom document protocols) to exchange documents containing health care information using the Generic exchange or MLLP exchange. When using HL7, the standard Oracle B2B features, such as validation, translation, automatic generation of outbound envelope headers, and acknowledgments, are available.

**Note:** While HL7 BATCH and FILE envelopes are supported, batching is not supported in this release.
For information about the organization that created and maintains the HL7 standards, go to http://www.hl7.org

**Document Version Parameters**
When you create an HL7 document version, you can set various parameters. Figure 7–11 shows document version parameters for an HL7 document.

**Figure 7–11  Document Version Parameters for an HL7 Document**

| Table 7–11 describes the document version parameters for an HL7 document. |

**Table 7–11  Document Version Parameters for an HL7 Document**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Header Tab</td>
<td>-</td>
</tr>
<tr>
<td>Security</td>
<td>In some applications of HL7, this field is used to implement security features.</td>
</tr>
<tr>
<td>Processing ID</td>
<td>MSH.11 - This field is used to decide whether to process the message as defined in HL7 Application (level 7) processing rules. The first component defines whether the message is part of a production, training, or debugging system (refer to HL7 table 0103 - Processing ID for valid values). The second component defines whether the message is part of an archival process or an initial load (refer to HL7 table 0207 - Processing mode for valid values). This allows different priorities to be given to different processing modes.</td>
</tr>
<tr>
<td>Accept Acknowledgement Type</td>
<td>B2B checks the payload (MSH.15) of an incoming message to see if an ACK has to be generated. In some HL7 Systems, MSH.15 is not sent in the payload at all and it is expected that an ACK is still sent.</td>
</tr>
</tbody>
</table>
Table 7–11  (Cont.) Document Version Parameters for an HL7 Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Acknowledgment Type</td>
<td>MSH.16. The value AL (always) is supplied.</td>
</tr>
<tr>
<td>Country Code</td>
<td>Sets the country of origin for the message. The value US is supplied.</td>
</tr>
<tr>
<td>Character Set</td>
<td>Sets the character set for the entire message. The value ASCII is supplied.</td>
</tr>
<tr>
<td>Internationalization Code Identifier</td>
<td>MSH.19</td>
</tr>
<tr>
<td>Internationalization Code Text</td>
<td>MSH.19</td>
</tr>
<tr>
<td>Internationalization Coding System Name</td>
<td>MSH.19</td>
</tr>
<tr>
<td>Internationalization Code Alternate Identifier</td>
<td>MSH.19</td>
</tr>
<tr>
<td>Internationalization Code Alternate Text</td>
<td>MSH.19</td>
</tr>
<tr>
<td>Internationalization Code Alternate Coding System Name</td>
<td>MSH.19</td>
</tr>
<tr>
<td>International Version Identifier</td>
<td>MSH.12</td>
</tr>
<tr>
<td>International Version ID Text</td>
<td>MSH.12</td>
</tr>
<tr>
<td>International Version ID Coding System Name</td>
<td>MSH.12</td>
</tr>
<tr>
<td>International Version ID Alternate Identifier</td>
<td>MSH.12</td>
</tr>
<tr>
<td>International Version ID Alternate Text</td>
<td>MSH.12</td>
</tr>
<tr>
<td>International Version ID Alternate Coding System Name</td>
<td>MSH.12</td>
</tr>
</tbody>
</table>

**Batch Header Tab**
- Create Batch Header: Check the box to create batch headers.
- Batch Header ecs File: Use the Browse button to find an ecs file to override the standard file. If not provided, the B2B-provided default file is used.
- Batch Security: BHS.8
- Batch Date: BHS.7. The system date-time stamp is supplied (#SystemDateTime(CCYYMMDDHHMM)#).

**File Header Tab**
- Create File Header: Check the box to enable.
- File Header ecs File: Use the Browse button to find an ecs file to override the standard file. If not provided, the B2B-provided default file is used.
- File Security: FHS.8
- File Date: FHS.7. The system date-time stamp is supplied (#SystemDateTime(CCYYMMDDHHMM)#).
Using the HL7 Document Protocol

Table 7–11 (Cont.) Document Version Parameters for an HL7 Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimiter Tab</td>
<td>Click Select Hexadecimal Characters next to any of the delimiter fields to provide values. See Table 7–4 for more about delimiters.</td>
</tr>
<tr>
<td>Element Delimiter</td>
<td>A single character that follows the segment identifier and separates each data element in a segment except the last. The value (0x7c) is supplied.</td>
</tr>
<tr>
<td>Escape Character</td>
<td>The value (0x5c) is supplied.</td>
</tr>
<tr>
<td>Repeating Separator</td>
<td>A service character used to separate adjacent occurrences of a repeating data element, or to separate multiple occurrences of a field. The value (0x7e) is supplied.</td>
</tr>
<tr>
<td>Segment Delimiter</td>
<td>A syntax character indicating the end of a segment (a logical grouping of data fields) within a message. The value (0x0d) is supplied.</td>
</tr>
<tr>
<td>Subcomponent Delimiter</td>
<td>The value (0x26) is supplied.</td>
</tr>
<tr>
<td>Subelement Delimiter</td>
<td>The value (0x5e) is supplied.</td>
</tr>
<tr>
<td>Miscellaneous Tab</td>
<td>-</td>
</tr>
<tr>
<td>Ignore Envelope Parameters</td>
<td>Use this option to provide a list of envelope elements, separated by commas, to be ignored during look-up validation. The possible values depend on the identifiers used in the agreement. For an HL7 agreement, the possible values include MessageSendingApp, MessageReceivingApp, MessageSendingFacility, and MessageReceivingFacility.</td>
</tr>
</tbody>
</table>

Document Type Parameters

When you create an HL7 document type, you can set various parameters. Figure 7–12 shows the document type parameters for an HL7 document.

Figure 7–12 Document Type Parameters for an HL7 Document

Table 7–12 describes the document type parameters for an HL7 document.
Table 7–12  Document Type Parameters for an HL7 Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Tab</td>
<td>-</td>
</tr>
<tr>
<td>HL7 Generic ACK</td>
<td>If selected, Oracle B2B sends a generic ACK immediately upon receiving an HL7 message.</td>
</tr>
<tr>
<td>Map ACK Control ID</td>
<td>Select to enable mapping the MSH.10 of the business message to the MSH.10 of the acknowledgment. Note: This Map ACK Control ID parameter is for the functional ACK.</td>
</tr>
<tr>
<td>Accept Acknowledgement</td>
<td>A functional acknowledgment is generated when MSH.15 has no value. Select <code>None</code> to take no action. Acknowledgment generation is dependent on the value in MSH.15 of the business message. Select <code>AL</code> (always) to generate the acknowledgment under any conditions. Select <code>ER</code> (error/reject) to generate the acknowledgment when the message errors or is rejected. Select <code>SU</code> (successful completion) to generate the acknowledgment when the message is successfully processed.</td>
</tr>
</tbody>
</table>

Document Definition Parameters

When you create an HL7 document definition, you can set various parameters. Figure 7–13 shows document definition parameters for an HL7 document.

Table 7–13 describes the document definition parameters for an HL7 document.

Table 7–13  Document Definition Parameters for an HL7 Document

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Tab</td>
<td>-</td>
</tr>
<tr>
<td>Transaction Set ecs File</td>
<td>Use the <code>Browse</code> button to find the ecs file.</td>
</tr>
<tr>
<td>Document Routing ID</td>
<td>Sets the consumer name to the back-end application</td>
</tr>
</tbody>
</table>
Notes on Using HL7

- No business message is produced for an HL7 immediate acknowledgment (transport-level acknowledgment). When using AS2, you see one acknowledgment business message for MDN (transport-level acknowledgment), and for ebMS, you see one acknowledgment business message in the business message report. In summary, because immediate acknowledgments are sent at the transport level, the entry is available only in the wire message report and not in the business message report.

- Negative acknowledgment messages indicating errors in an HL7 exchange may be truncated because of the 80-character length limitation in HL7 versions 2.1 through 2.5.

7.5 Using the Positional Flat File Document Protocol

Oracle B2B supports message exchange for positional flat files and SAP iDocs (intermediate documents (text files) used with SAP applications). This adds capabilities beyond handling XML files and traditional EDI files based on various XML and EDI standards.

Document Version Parameters
No parameters need to be set when you create the document version for a positional flat file.

Document Type Parameters
No parameters need to be set when you create the document type for a positional flat file.

Document Definition Parameters
When you create a document definition for a positional flat file, you can set various parameters. Figure 7–14 shows document definition parameters for a positional flat file.
Oracle B2B implements the nonproprietary, XML-based RosettaNet standards to exchange documents over the Internet. RosettaNet standards prescribe when information should be exchanged, acknowledged, or confirmed, and how messages in an exchange should be packaged and physically exchanged between trading partners. In addition to using the RosettaNet document guideline files in Oracle B2B Document Editor, you can also download standard DTD files from the RosettaNet Web site.

A RosettaNet DTD, when used with Oracle B2B in a SOA composite application, must be converted to an XSD. An AQ Adapter added to the composite application can convert the inbound DTD to an XSD and manipulate the data as needed. Likewise, the

Table 7–14 describes the document definition parameters for a positional flat file.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transaction</strong></td>
<td>-</td>
</tr>
<tr>
<td>Transaction Set ecs File</td>
<td>Use the <strong>Browse</strong> button to find the ecs file.</td>
</tr>
<tr>
<td><strong>Identification</strong></td>
<td>-</td>
</tr>
<tr>
<td>Identification Value</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Identification Start Position</td>
<td>Used in combination with the end position to retrieve a value from the payload between the start and end positions</td>
</tr>
<tr>
<td>Identification End Position</td>
<td>Used in combination with the start position to retrieve a value from the payload between the start and end positions</td>
</tr>
<tr>
<td><strong>Routing</strong></td>
<td>-</td>
</tr>
<tr>
<td>Document Routing ID</td>
<td>Sets the consumer name to the back-end application</td>
</tr>
</tbody>
</table>
AQ Adapter can convert the outbound XSD to a DTD for Oracle B2B to send the message out.

RosettaNet standards are specified by using of the RosettaNet Partner Interface Process (PIP), RosettaNet Dictionaries, and RNIF. Oracle B2B supports all PIPs. (The RosettaNet Technical Dictionary is not supported in Oracle B2B.)

For information about the RosettaNet consortium and its history, and for a complete list of PIP clusters and segments, go to

http://www.rosettanet.org

7.6.1 PIPs

A PIP is an XML-based dialog that defines the business processes between trading partners. It defines the structure, sequence of steps, roles (buyer and seller) activities, data elements, values, and value types for each business document message exchanged between trading partners.

Using PIP 3A4 as an example, you can see how a PIP defines a dialog between trading partners, as shown in Figure 7–15.

**Figure 7–15  PIP 3A4 Message Exchange Between Buyer and Seller**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3       | Order manage cluster, with which trading partners can:  
  - Order catalog products  
  - Create custom orders  
  - Manage product distribution and delivery  
  - Support product returns and financial transactions |
| 3A      | Quote and order entry segment |
| 3A4     | Specific PIP type, which supports:  
  - Submittal of a purchase order by a buyer  
  - Submittal of an acceptance purchase order by a seller  
  - Ability of a buyer to cancel or change a purchase order based on the acknowledgment response |
Document Version Parameters
No parameters need to be set when you create the document version for a RosettaNet document.

Document Type Parameters
When you create a RosettaNet document type, you can set various parameters. Figure 7–16 shows document type parameters for a RosettaNet document.

Figure 7–16  Document Type Parameters for a RosettaNet Document

Table 7–16 describes document type parameters for a RosettaNet document.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Header Tab</td>
<td></td>
</tr>
<tr>
<td>*From Role</td>
<td>The trading partner that sends the message (in Partner Role Description of the PIP).</td>
</tr>
<tr>
<td>*To Role</td>
<td>The trading partner that receives the message (the role the trading partner receiving the message plays in the PIP).</td>
</tr>
<tr>
<td>*From Service</td>
<td>The service that sends the message.</td>
</tr>
<tr>
<td>*To Service</td>
<td>The service to which the message is sent.</td>
</tr>
<tr>
<td>*Business Transaction Name</td>
<td>The name of the business transaction is required.</td>
</tr>
<tr>
<td>*Business Action</td>
<td>The name of the business action is required. The value must be consistent with the Global Business Action Code.</td>
</tr>
</tbody>
</table>
Using the RosettaNet Document Protocol

**Table 7–16 (Cont.) Document Type Parameters for a RosettaNet Document**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Time to Perform for Collaboration</td>
<td>The time to perform the business action is required.</td>
</tr>
<tr>
<td>*Collaboration Name</td>
<td>The RosettaNet collaboration name signifies the business transaction between trading partners (the roles as buyer and seller) depending on a common transaction. Required.</td>
</tr>
<tr>
<td>*Collaboration Code</td>
<td>The textual form of the abbreviated collaboration name. Required.</td>
</tr>
</tbody>
</table>

**Document Definition Parameters**

When you create a RosettaNet document definition, you can set various parameters. Figure 7–17 shows the document definition parameters for a RosettaNet document.

**Figure 7–17 Document Definition Parameters for a RosettaNet Document**

Table 7–17 describes the document definition parameters for a RosettaNet document.
7.6.2 RosettaNet Validation

RosettaNet validation compares the elements in RosettaNet XML-format business documents to the requirements specified in the RosettaNet Message Guideline specification to determine their validity. This specification defines requirements for details such as element datatypes, element lengths, element value lists, and element cardinality. PIPs that require RosettaNet dictionary validation are also validated when a dictionary is present.

The minimum validation-level requirements on the sections of a RosettaNet XML-format business document are as follows. These requirements cover the preamble, delivery header, service header, and service content sections of a document. Documents not following one or more of these requirements are identified as invalid.

1. The XML-format business document requires compliance with its DTD.
2. Elements with datatypes, lengths, or both that are specified in the RosettaNet Message Guideline specification require validation against this specification.
3. An element’s list of values specified in the entity instance list in the corresponding RosettaNet Message Guideline specification requires validation against this specification.
4. If the Message Guideline specification defines the cardinality specification of an element differently from the corresponding DTD specification, the Message Guideline specification takes precedence.
5. If a PIP requires dictionary validation, and a dictionary is included, the service content requires validation against the dictionary as a part of action performance.

6. Cross-tag validation is based on message guidelines.

7.7 Using the UCCNet Document Protocol

Oracle B2B implements UCCNet, which enables trading partners—typically retailers and suppliers in the retail and consumer goods industries—to exchange documents with UCCNet. Table 7–18 lists the UCCNet document types supported in Oracle B2B.

<table>
<thead>
<tr>
<th>Table 7–18</th>
<th>UCCNet Document Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td></td>
</tr>
<tr>
<td>registerCommand</td>
<td></td>
</tr>
<tr>
<td>confirmCommand</td>
<td></td>
</tr>
</tbody>
</table>
Using the UCCNet Document Protocol

Table 7–18 (Cont.) UCCNet Document Types

<table>
<thead>
<tr>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>linkCommand</td>
</tr>
<tr>
<td>checkComplianceCommand</td>
</tr>
<tr>
<td>documentCommand</td>
</tr>
<tr>
<td>documentIdentificationCommand</td>
</tr>
<tr>
<td>notificationStateCommand</td>
</tr>
<tr>
<td>queryCommand</td>
</tr>
<tr>
<td>registerLinkCommand</td>
</tr>
<tr>
<td>publicationCommand</td>
</tr>
<tr>
<td>publishCommand</td>
</tr>
<tr>
<td>catalogueItemMaintenanceCommand</td>
</tr>
<tr>
<td>priceCommand</td>
</tr>
<tr>
<td>validateCommand</td>
</tr>
<tr>
<td>registerOwnershipCommand</td>
</tr>
<tr>
<td>subscriptionCommand</td>
</tr>
<tr>
<td>notifyCommand</td>
</tr>
<tr>
<td>response</td>
</tr>
</tbody>
</table>

For information about the organization that created and maintains the UCCNet standards, go to

http://www.lsync.org

Document Version Parameters
No parameters need to be set when you create the document version for a UCCNet document.

Document Type Parameters
No parameters need to be set when you create the document type for a UCCNet document.

Document Definition Parameters
When you create a UCCNet document definition, you can set various parameters. Figure 7–18 shows document definition parameters for a UCCNet document.
Using the UCCNet Document Protocol

Figure 7–18  Document Definition Parameters for a UCCNet Document

Table 7–19 describes the document definition parameters for a UCCNet document.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XML Tab</strong></td>
<td></td>
</tr>
<tr>
<td>Identification Expression (XPath)</td>
<td>Locates a node in the XML payload</td>
</tr>
<tr>
<td>Identification Value</td>
<td>Provides the value to match in the node identified by the Identification Expression. If the values match, then the document is successfully identified. If the value is left blank, then Oracle B2B checks for the existence of the node and the document is successfully identified.</td>
</tr>
<tr>
<td><strong>Routing Tab</strong></td>
<td></td>
</tr>
<tr>
<td>Document Routing ID</td>
<td>Sets the consumer name to the back-end application</td>
</tr>
</tbody>
</table>

7.7.1 Creating a 1Sync Document

The 1Sync document protocol helps in the data synchronization between seller and buyer, which enables the transfer of product and location information with the continuous synchronization of the data over time.

Use the Custom document protocol or the UCCNet document protocol to create a 1Sync XML document.
Using the UCCNet Document Protocol

**Note:** The GS-1 organization has changed the standard name from UCCNet to 1Sync. Use either the seeded UCCNet document protocol or create a new Custom document protocol, 1Sync, as illustrated in the figure. The functionality is the same.

Figure 7–19 shows a document definition for a 1Sync document, using the Custom document protocol.

**Figure 7–19  1Sync Document Definition**

You can correlate 1Sync request and response messages as follows:

- Use the document routing ID on the **Routing** tab. The routing ID 1Sync_64_catalogueRequest is shown Figure 7–20.

**Figure 7–20  The Routing Tab for a 1Sync Document Definition**

See the following for more information:
Section 7.1, "Using the Custom Document Protocol"

Section 1.7, "Oracle B2B Samples and Cookbooks." A 1Sync example is provided with the B2B samples.

### 7.8 Changing Document Details

Document details—document protocol versions and document type parameters—can be changed for a remote trading partner from the Partners > Documents tab. Host administrators can change any remote trading partner’s document details here (host administrators must change their own data on the Administration > Document tab), and remote administrators can change document details for their own data, if the remote administrator has been granted access to those document types. See Section 1.4.2, "Restricting Access to Document Types," for more information.

Figure 7–21 shows the Version tab in the Document Details section, where parameters for the document protocol version can be changed.

**Figure 7–21 Changing Document Details**

Delimiters, and parameters such as Interchange Control Version Number, Interchange Date, and Interchange Time are typically changed for different remote trading partners.

Figure 7–22 shows the Document Type tab, where parameters for the document type can be changed.
Figure 7–22 Changing Document Details

Use the **Override Version Param** and **Override DocType Param** parameters to indicate that override values are provided. Document type parameter values set for a remote trading partner take precedence over the default document type parameter values set for the document definition when the document was created on the **Administration > Document** tab.

To **override** document details:
1. Click the **Partners** tab.
2. Click the **Documents** tab.
3. Select a remote trading partner.
4. Select a document definition.
5. Select the override types that apply:
   - **Override Version Param**
   - **Override DocType Param**
6. Provide values to override values on the **Version** tabs or the **Document Type** tabs, or both.
7. Click **Save**.

### 7.8.1 Changing Document Definitions After Deploying an Agreement

Changes to a document definition after an agreement is deployed are not reflected in the trading partner’s profile. Use the **Document Details** area on the **Partners >**
Documents tab to change document protocol version and document type parameters. Then redeploy the agreement.

7.8.2 Changing Document Definitions After Importing Metadata

If you import B2B metadata and then change the document from the Administration > Document tab, then you must also make the same changes to the supported document definition for the host and remote trading partners from the Partners > Documents tab. Use the Version, Document Type, and Definitions tabs under Document Details to make the changes.

7.9 Using Document Routing IDs

A document routing ID is useful in two circumstances: when enqueuing to an AQ queue and when using B2B documents in a SOA composite application. If you set a document routing ID for messages enqueued to an AQ queue (inbound only), then the AQ consumer name is set to the document routing ID. Within a SOA composite application, if you use a document routing ID in your B2B binding component instead of the document definition, then all messages with the same document routing ID are routed to the same SOA composite.

This is useful if you have many different document definitions, but you want them to be handled the same way. The WSDL uses the document routing ID instead of the document definitions. In a SOA composite application, the B2B Configuration Wizard provides an option to use the document routing ID instead of selecting a document definition, as shown in Figure 7–23.

*Figure 7–23  Document Routing ID Option in Oracle JDeveloper*

When using AQ, if you set the routing ID value instead of using the default b2buser, then do not set it to a numeric value. Use a combination of alphabetic and numeric values.
Deploying an agreement is the process of validating and activating a set of run-time data that is used for run-time transactions.

This chapter contains the following topics:

- Section 8.1, "Introduction to Agreement Deployment States"
- Section 8.2, "Managing Deployed Agreements"

See Chapter 6, "Creating and Deploying Trading Partner Agreements," for more information about how to deploy an agreement.

### 8.1 Introduction to Agreement Deployment States

You can manage the state of a deployment—Active, Inactive, Retired, or Purged—as shown in Figure 8–1. You can also search on the deployed agreements in the run-time repository, as well as export an agreement.

**Figure 8–1 Managing a Deployed Agreement**

---

<table>
<thead>
<tr>
<th>Agreement</th>
<th>User</th>
<th>State</th>
<th>First Deployed Data</th>
<th>Last Deployed Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Active</td>
<td>8/4/2009 1:30 PM</td>
<td>8/4/2009 1:30 PM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Active</td>
<td>8/4/2009 1:30 PM</td>
<td>8/4/2009 1:30 PM</td>
</tr>
</tbody>
</table>
8.2 Managing Deployed Agreements

A deployed agreement is initially in the Active state. Table 8–1 describes the deployment states.

<table>
<thead>
<tr>
<th>Table 8–1 Deployed Agreement States</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State</strong></td>
</tr>
<tr>
<td>Active</td>
</tr>
<tr>
<td>Inactive</td>
</tr>
<tr>
<td>Retired</td>
</tr>
<tr>
<td>Purged</td>
</tr>
</tbody>
</table>

Table 8–2 describes the information displayed for a deployed agreement.

<table>
<thead>
<tr>
<th>Table 8–2 Deployed Agreements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field</strong></td>
</tr>
<tr>
<td>Agreement</td>
</tr>
<tr>
<td>User</td>
</tr>
<tr>
<td>State</td>
</tr>
<tr>
<td>First Deployed Date</td>
</tr>
<tr>
<td>Last Deployed Date</td>
</tr>
</tbody>
</table>

8.2.1 Searching for Deployed Agreements

Use the search parameters described in Table 8–3 to search for deployed agreements.
Table 8–3  Search Parameters for Searching on Deployed Agreements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a string that is contained in the agreement name, equals the name, or is at the end of the name.</td>
</tr>
<tr>
<td>Responding Partner</td>
<td>Enter a string that is contained in the responding partner name, equals the name, or is at the end of the name.</td>
</tr>
<tr>
<td>Initiating Partner</td>
<td>Enter a string that is contained in the initiating partner name, equals the name, or is at the end of the name.</td>
</tr>
<tr>
<td>*State</td>
<td>Select from All, Active, Inactive, or Retire.</td>
</tr>
<tr>
<td>Document Definition</td>
<td>Select from one of the document definitions you previously created. See Chapter 4, “Creating Document Definitions,” for more information.</td>
</tr>
</tbody>
</table>

- Click **Reset** to return the search parameters shown in Table 8–3 to their previous settings.
- Click **Advanced** to select additional search parameters, as shown in Figure 8–2.

Figure 8–2  Advanced Search Parameters

If you select the document search parameters from the Add Fields list, use them as follows: Select a document protocol name first to populate the list of document protocol versions; next select a document protocol version to populate the list of document types; and then select a document type to populate the list of document definitions.

The Saved Search feature is not available.
8.2.2 Changing the Deployment State

To change the deployment state:
1. Click the Administration link.
2. Click the Manage Deployments tab.
3. Select an agreement.
4. Click one of the available actions:
   - If the state is Active, then Inactive is available.
   - If the state is Inactive, then Active or Retire is available.
   - If the state is Retired, then Purge is available.

8.2.3 Exporting an Active Agreement

You can export active agreements. For agreements that use HTTPS or digital signature and encryption, the key store password of the host trading partner is not included as part of the export file. This is because a key store is specific to each computer. Therefore, when the export file is imported on a different computer, you must re-create the key store password and update the key store location (if needed) for the host trading partner in the B2B interface. If the export file is imported back or the key store and its location have not changed on the target computer, then the key store password and location may be identical to the first key store and key store password you used. This applies only to the host trading partner.

---

**Caution:** Do not manually edit exported files. If you do so, Oracle B2B cannot guarantee their integrity.

---

To export an active agreement:
1. Click the Administration link.
2. Click the Manage Deployments tab.
3. Select an agreement (or multiple agreements).
4. Click Export.

The system-provided file name is MDS_EXPORT_DD_MM_YYYY.zip. You can choose whether you want to open the file or save it, in which case you can specify a file name and download location. Each agreement is a separate ZIP file within MDS_EXPORT_DD_MM_YYYY.zip.

Exporting can take some time based on the agreement metadata.
You can create identifier types, contact information types, and trading partner parameter types. With custom types, Oracle B2B can meet individual specifications for document exchange, contact information, and trading partner parameters.

This chapter contains the following topics:

- Section 9.1, "Creating Custom Identifier Types"
- Section 9.2, "Creating Custom Contact Information Types"
- Section 9.3, "Creating Custom Trading Partner Parameter Types"

See Section 5.2, "Creating Trading Partner Profiles," for information on adding custom types and values to a trading partner profile.

### 9.1 Creating Custom Identifier Types

Identifier types, or identifiers, help in identifying a trading partner (as exchange identifiers) or can be used to define additional inputs for various document protocols.

Oracle B2B has preseeded many of the commonly required identifiers. A new custom identifier can be created as required.

**To create an identifier type:**

1. Click the Administration link.
2. Click the Types tab.
3. In the Identifiers area, click Add.
4. Provide a name and optional description, as shown in Figure 9–1.
Creating Custom Identifier Types

Figure 9–1 Creating an Identifier Type

5. Click Save.

See Task 3, "Add Identifier Types and Values" on page 5-4 for how to add the new type and a value to a trading partner’s profile.

Oracle B2B provides predefined identifiers for the supported document protocols, as listed in Table 9–1. You can delete unused types to further customize your B2B environment. A type that is used by a trading partner cannot be deleted.

Table 9–1 Identifier Types Defined in Oracle B2B

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS1 Identifier</td>
<td>The specification for using EDI over SMTP to transmit data using e-mail. AS1 also works with non-EDI document types such as XML and TXT files. The AS1 Identifier and the Name identifier are required for AS1 exchanges.</td>
</tr>
<tr>
<td>AS2 Identifier</td>
<td>An alias for the service address (specified by the AS2-From/AS2-To fields) inside an AS2 transaction. The value can be any unique name that a trading partner recognizes. The AS2 Identifier and the Name identifier are required for AS2 exchanges.</td>
</tr>
<tr>
<td>DUNS</td>
<td>A unique, sequentially-generated, nine-digit number that is obtained from Dun and Bradstreet, formally as a D-U-N-S number. The DUNS Identifier and the Name identifier are required for RNIF exchanges.</td>
</tr>
<tr>
<td>EDI Group ID</td>
<td>Used to identify multiple branches within a trading partner’s company. The group ID can be the same as the interchange ID.</td>
</tr>
<tr>
<td>EDI Group ID Qualifier</td>
<td>Similar to EDI Interchange ID Qualifier, this qualifier applies to EDIFACT only (not X12).</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EDI Interchange ID</td>
<td>A unique identifier for a trading partner that can come from different sources. For example, if the trading partner has a Dun &amp; Bradstreet number, that number can be used for the interchange ID. In most cases, the selected VAN assigns the interchange ID.</td>
</tr>
<tr>
<td>EDI Interchange ID Qualifier</td>
<td>Informs the network of the type of interchange ID that follows. Typical qualifiers include ZZ, indicating that the interchange ID that follows is mutually defined; 01, indicating that the interchange ID is the trading partner’s Dun and Bradstreet number; 12, indicating that the interchange ID is a telephone number.</td>
</tr>
<tr>
<td>EDI Interchange Internal ID</td>
<td>Identifies the trading partner based on the EDI interchange internal ID.</td>
</tr>
<tr>
<td>EDI Interchange Internal Sub ID</td>
<td>Identifies the trading partner based on the EDI interchange internal sub-ID.</td>
</tr>
<tr>
<td>Generic Identifier</td>
<td>The IP address to use for identifying trading partners if you are using the generic exchange protocol (EDI X12 over Generic Exchange, EDI EDIFACT over Generic Exchange, or Custom Document over Generic Exchange) with the HTTP or HTTPS transport protocol. Do not enter the host name. The Generic Identifier and the Name identifier are required for Generic HTTP and Generic Email exchanges. Wildcard characters are not permitted in the IP address.</td>
</tr>
<tr>
<td>HL7 Batch Application ID</td>
<td>Identifies the trading partner based on the HL7 batch application ID. BHS.3 and BHS.5 have the same definition as the corresponding field in the MSH segment.</td>
</tr>
<tr>
<td>HL7 Batch Application Universal ID</td>
<td>Identifies the trading partner based on the HL7 batch application universal ID.</td>
</tr>
<tr>
<td>HL7 Batch Application Universal ID Type</td>
<td></td>
</tr>
<tr>
<td>HL7 Batch Facility ID</td>
<td>Identifies the trading partner based on the HL7 batch facility ID.</td>
</tr>
<tr>
<td>HL7 Batch Facility Universal ID</td>
<td>Identifies the trading partner based on the HL7 batch facility universal ID.</td>
</tr>
<tr>
<td>HL7 Batch Facility Universal ID Type</td>
<td></td>
</tr>
<tr>
<td>HL7 File Application ID</td>
<td>Identifies the trading partner based on the HL7 file application ID. FSH.3 and FSH.5 have the same definition as the corresponding field in the MSH segment.</td>
</tr>
<tr>
<td>HL7 File Application Universal ID</td>
<td>Identifies the trading partner based on the HL7 file application universal ID.</td>
</tr>
<tr>
<td>HL7 File Application Universal ID Type</td>
<td></td>
</tr>
<tr>
<td>HL7 File Facility ID</td>
<td>Identifies the trading partner based on the HL7 file facility ID. This field further describes the sending/receiving application. The facility ID can have an organizational entity, unit, product or vendor's identifier.</td>
</tr>
<tr>
<td>HL7 File Facility Universal ID</td>
<td>Identifies the trading partner based on the HL7 file facility universal ID.</td>
</tr>
<tr>
<td>HL7 File Facility Universal ID Type</td>
<td></td>
</tr>
<tr>
<td>HL7 Message Application ID</td>
<td>Identifies the sending/receiving application.</td>
</tr>
<tr>
<td>HL7 Message Application Universal ID</td>
<td>For outbound messages, this field is used to override the Message Application Universal ID, which is MSH.3 for the sender and MSH.5 for the receiver. For inbound messages, this field is used for lookup.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Creating Custom Contact Information Types

Oracle B2B provides a centralized location for trading partner contact information. After you create a type, you can add it to a trading partner’s profile and change its value.

You can create any type of contact information. You may want to create types for contact names, e-mail addresses, telephone and fax numbers, and so on. You can deleted unused types to further customize your B2B environment. A type that is used by a trading partner cannot be deleted.

To create a contact information type:
1. Click the Administration link.
2. Click the Types tab.
3. In the Contact Information area, click Add.
4. Provide a name for the contact information type, an optional description, and click Save.

The string that you provide in the Name field is displayed in a list under the Type field on the Partners > Profile page.

See Task 4, "Add Contact Information" on page 5-5 for how to add the new type and a value to a trading partner’s profile.

Creating Custom Trading Partner Parameter Types

Trading partner parameter types are string types. After you create a type, you can add it to a trading partner’s profile and change its value.

### Table 9–1 (Cont.) Identifier Types Defined in Oracle B2B

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL7 Message Application Universal ID Type</td>
<td>For outbound messages, this field is used to override the Message Application Universal ID Type, which is MSH.3 for the sender and MSH.5 for the receiver. For inbound messages, this field is used for lookup.</td>
</tr>
<tr>
<td>HL7 Message Facility ID</td>
<td>Identifies the trading partner based on the HL7 message facility ID.</td>
</tr>
<tr>
<td>HL7 Message Facility Universal ID</td>
<td>For outbound messages, this field is used to override the Message Facility Universal ID, which is MSH.4 for the sender and MSH.6 for the receiver. For inbound messages, this field is used for lookup.</td>
</tr>
<tr>
<td>HL7 Message Facility Universal ID Type</td>
<td>For outbound messages, this field is used to override Message Facility Universal ID, which is MSH.4 for the sender and MSH.6 for the receiver. For inbound messages, this field is used for lookup.</td>
</tr>
<tr>
<td>MLLP ID</td>
<td>The TCP/IP Minimum Lower Layer Protocol (MLLP) is the standard for HL7. The MLLP ID and the Name identifier are required for MLLP exchanges.</td>
</tr>
<tr>
<td>Name</td>
<td>Identifies the trading partner by its name. The value for this type is automatically supplied when you create or edit the trading partner name, for example, Acme or GlobalChips. The Name identifier is required for Generic File, Generic FTP, Generic SFTP, Generic AQ, and Generic JMS exchanges.</td>
</tr>
<tr>
<td>ebMS Identifier</td>
<td>This type, OASIS ebXML Messaging Services (ebXML), specifies a secure and reliable way to exchange messages using HTTP, HTTPS, SOAP, XMLDsig, and XMLOEncrypt. The ebMS Identifier and the Name identifier are required for ebMS exchanges.</td>
</tr>
</tbody>
</table>
To create a trading partner parameter type and default value:

1. Click the Administration link.
2. Click the Types tab.
3. In the Trading Partner Parameters area, click Add.
4. Provide the following information and click Save.
   - Name (required)
   - Default Value (optional)
   - Group Name (optional)
   - Display Name (optional; however, the value of Display Name, not Name, appears when you add this type to a trading partner profile)
   - Description (optional)

See Task 5, "Add a Trading Partner Parameter and Value" on page 5-5 for how to add the new type and a value to a trading partner’s profile.

There are no predefined trading partner parameter types. You may want to create a type named Country, for example. Then the value—a specific country code—can be configured for each trading partner. You can deleted unused types to further customize your B2B environment. A type that is used by a trading partner cannot be deleted.
For design-time data, use the Oracle B2B interface to import and export B2B repositories.

This chapter contains the following topics:

- Section 10.1, "Importing and Exporting the Design-Time Repository"
- Section 10.2, "What Is Copied When You Import or Export from the Import/Export Tab"
- Section 10.3, "About the Exported File"


10.1 Importing and Exporting the Design-Time Repository

Oracle B2B design-time data can be exported and saved to a ZIP file. The ZIP file can be imported back into Oracle B2B so that the data is available in the B2B interface. This is useful when migrating data from a test environment to a production environment.

---

**Caution:** Do not manually edit exported files. If you do so, Oracle B2B cannot guarantee their integrity.

---

You can also export data from other areas of the Oracle B2B interface:

- Click **Partners > Profile** to export trading partner data. See Section 5.2, "Creating Trading Partner Profiles," for more information.
- Click **Partners** and then select an agreement to export. See Section 6.4, "Deleting and Exporting Agreements," for more information.
- Click **Administration > Manage Deployments** to export deployed agreements. See Section 8.2.3, "Exporting an Active Agreement," for more information.

You can import sample files that use the following document types: Custom, EDI EDIFACT, EDI X12, HL7, and RosettaNet. See Section 1.7, "Oracle B2B Samples and Cookbooks," for the download location and information about the scenarios presented in the samples.

Figure 10–1 shows the **Import/Export** tab, where you import and export design-time data.
Figure 10–1 Importing and Exporting Data

When you import metadata, the updates to your existing B2B are incremental unless you select the Replace Existing Metadata option. To delete all existing data before importing metadata, use the Purge tab under the Administration link. See Chapter 13, "Purging Data," for more information.

**Caution:** Complete export operations without interruption or idle time. Leaving the browser idle for more than a few minutes during export operations can cause file corruption.

**To import data:**
1. Click the Administration link.
2. Click the Import/Export tab.
3. Click Browse to find the metadata repository ZIP file.
   The default name for exported metadata is MDS_EXPORT_DD_MM_YEAR.zip.
   If you are importing a ZIP file that contains multiple ZIP files within it, you must unzip the containing file and import each ZIP file separately. Individual ZIP files are created when you export multiple agreements at the same time.
4. If you select Replace Existing Metadata, then current metadata in the Metadata Service (MDS) repository is overwritten. If it is not selected, then only new data is copied to the MDS repository.
5. Click Import.
   Depending on the size of the design-time repository contents, this process can take time.

**To export data:**

**Caution:** Do not manually edit exported files.

1. Click the Administration link.
2. Click the **Import/Export** tab.

3. **Select Entire Repository** or **Active Agreements**.

   The entire repository includes all data in the B2B design-time repository—agreements in all states, all trading partner configurations, and so on.

   Active agreements are all deployed agreements that are not inactive, retired, or purged.

4. **(Optional) Narrow the list of agreements by using the **Search** option.**
   
   a. Select **Agreement** or **Document Type**.
   
   b. Enter part or all of an agreement name or document type name and click **Search**.
   
   c. Click **Search**.
   
   d. Select one or more agreements from the search results.

      If you select multiple agreements, each agreement is exported in its own ZIP file, and all the individual ZIP files are contained in the export ZIP file.

5. Click **Export**.

6. **Select Open with** or **Save to Disk**.

   The system-provided file name is MDS_EXPORT_DD_MM_YYYY.zip. As shown in **Figure 10–2**, you can choose whether you want to open the file or save it, in which case you can specify a file name and download location.
Figure 10–2  Exporting Data

Clicking **Import** imports whatever is in the export file (that is, the file that was previously exported), which can possibly include B2BUser and ParameterValue objects. A warning message is displayed to indicate that, if the file contains credential- and policy-related data, then the credential and policy stores must also be imported.

User information—including user permissions for document-type access (see Section 1.4.2, "Restricting Access to Document Types")—is not copied when you export a repository.

ParameterValue objects for passwords are copied when you export a repository.

The B2B import and export functionality is separate from the credential store and policy store import and export functionality. Use the Oracle WebLogic Server tools to import and export identity, credential, and policy stores.

Passwords are not copied when you import a repository. Passwords must be re-created in the destination B2B instance. Passwords are not copied when you export the design-time repository.

Callout library JAR files are not copied during import or export. See Table 12–1, "Callout Details" for more information.
If you export the design-time repository and then continue to make changes to the repository contents in the Oracle B2B interface, and if you later import the exported file (the contents of which are now older), then updates are as follows:

- If **Replace Existing Metadata** is not checked during import, then new data created in the Oracle B2B interface after the file was exported is left untouched.
- If **Replace Existing Metadata** is checked during import, then data updated or deleted after the file was exported is overwritten with the older contents of the imported file.

If an import fails, then the changes are rolled back and the design-time repository remains unchanged. A message appears indicating that the import was unsuccessful.

### 10.3 About the Exported File

Design-time repository contents that are exported to a file represent a copy of the current data. This file is no longer accessible for changes with the Oracle B2B user interface until it is imported back into Oracle B2B. Do not manually edit exported files.

#### 10.3.1 Exported ZIP Files Containing Agreement Names in Multibyte Character Languages

If you select multiple agreements to export, and any of those agreement names are in a multibyte character language, then in the export ZIP file, which contains a separate ZIP file for each agreement, the ZIP file names for the agreement names in multibyte characters are garbled. This affects your ability to import the ZIP file back into Oracle B2B. Use one of the following approaches for working with this type of file:

- To import a ZIP file containing multiple agreements, in which one or more of the agreement names are in a multibyte character language, use a UTF-8-based unzip tool, such as WinZip version 11.2, to unzip the export file. Then import the individual ZIP files into B2B.
- Alternatively, you can export agreement names that use a multibyte character language one at a time (one per ZIP file). Then import the individual ZIP files as you normally would.
11

Batching EDI Messages

For outbound messages, use the Oracle B2B interface to batch, schedule, and send outbound EDI X12 and EDI EDIFACT messages. (Inbound messages to Oracle B2B are automatically debatched.)

This chapter contains the following topics:

- Section 11.1, "Setting Up a Batch"
- Section 11.2, "Managing Batched Messages"

See the following for more information about EDI:

- Section 7.2, "Using the EDI EDIFACT Document Protocol"
- Section 7.3, "Using the EDI X12 Document Protocol"

11.1 Setting Up a Batch

Batching is often used to group messages by document type; for example, you may want to send out a batch of purchase orders or a batch of invoices, to one or more trading partners. You can also batch multiple document types, sent to one or more trading partners.

When batching more than 1000 documents, where the payload size is greater than 30 KB, specific settings are required. See Section A.1.9, "Settings for Batching a Large Number of Documents."

Figure 11–1 shows where you set up a batch transmission of EDI messages.
To set up a batch, do the following:

- **Task 1, “Search for Agreements to Batch”**
- **Task 2, “Create the Batch”**
- **Task 3, “Schedule the Batch”**

### Task 1 Search for Agreements to Batch

1. Click the **Administration** link.
2. Click the **Schedule Batch** tab.
3. Use the search parameters described in Table 11–1 to identify which agreements you want to batch.

Use the document search parameters as follows: Select a document protocol name first to populate the list of document protocol versions; next select a document protocol version to populate the list of document types; and then select a document type to populate the list of document definitions.

### Table 11–1 Search Parameters for Creating a Batch

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match All or Any</td>
<td>If you select <strong>All</strong>, then fields with values are matched using an and condition. If you select <strong>Any</strong>, then fields with values are matched using an or condition.</td>
</tr>
<tr>
<td>Responding Partner</td>
<td>Select <strong>Starts With</strong>, <strong>Contains</strong>, <strong>Equals</strong>, or <strong>Ends With</strong>, and type the appropriate portion of the name of the responding trading partner.</td>
</tr>
<tr>
<td>Agreement</td>
<td>Select <strong>Starts With</strong>, <strong>Contains</strong>, <strong>Equals</strong>, or <strong>Ends With</strong>, and type the appropriate portion of the name of the agreement.</td>
</tr>
<tr>
<td>Document Protocol Name</td>
<td>Select <strong>EDI_EDIFACT</strong> or <strong>EDI_X12</strong>.</td>
</tr>
<tr>
<td>Document Protocol Version</td>
<td>Select a document protocol version that you previously created.</td>
</tr>
</tbody>
</table>
4. Click **Search**.
   
   Active, deployed agreements (outbound) that meet your search criteria are displayed.

5. Go to **Task 2, "Create the Batch"**.

**Task 2  Create the Batch**

1. Enter a name for the batch.
2. Select the agreements you want to batch.
3. Click **Create Batch**.
4. Go to **Task 3, "Schedule the Batch"**.

**Task 3  Schedule the Batch**

1. Click **Launch Scheduler**.
2. Select the **Non-Repeating Event** tab or the **Repeating Event** tab.
3. For a nonrepeating event, do one of the following:
   - Enter the date in the format shown in the Scheduler dialog and click **OK**.
   - Or, click the **Calendar** icon, specify a date and time, and click **OK**.
   
   For a repeating event (see Figure 11–2), enter details on the interval to trigger the event by specifying the minutes, hour, month, year, and date details. Then click **OK**.

**Figure 11–2  Scheduling the Batch**

You can see the batches you create on the **Manage Batch** tab.
Managing Batched Messages

11.2 Managing Batched Messages

Figure 11–3 shows where you can search for batches that you previously created, view details of a batch, and disable, update, or delete a batch.

Figure 11–3  Managing Batched EDI Messages

For the Update batch action, only the documents definitions selected can be updated. Ensure that you reselect all the documents that are to be part of the batch and not just the new ones.

In some cases, B2B may not pick up the batched messages when you update the batching schedule. If you see that batched messages are not being picked up, delete the batch and create a new batch schedule with the same name as the previous batch. The same name must be used so that B2B picks up the previous messages in WAIT_BATCH status.
This chapter describes how to create and use Java callouts, which transform the formats of messages exchanged between the host and remote trading partners. You can use callouts to invoke an XSLT style sheet, and any Java program in general.

This chapter contains the following topics:

- Section 12.1, "Introduction to Callouts"
- Section 12.2, "Creating a Callout"
- Section 12.3, "Including a Callout in an Agreement"
- Section 12.4, "Implementing a Callout"

### 12.1 Introduction to Callouts

Callouts are used in environments in which a host trading partner application does not use the same message format as the remote trading partner. For example, a remote trading partner sends a RosettaNet XML-formatted purchase order request to a host trading partner, as shown in Figure 12–1.
In this example, the host application of the host trading partner is an Oracle E-Business Suite application that does not use RosettaNet XML-formatted messages. To enable communication between these two different formats, you create two callouts, as follows:

- One callout, `callout_inbound`, for example, transforms the RosettaNet XML-formatted purchase order request into an Oracle E-Business Suite XML format understood by the Oracle E-Business Suite application. The Oracle E-Business Suite application, in turn, responds to the request message with a purchase order acceptance message in Oracle E-Business Suite XML format.

- The other callout, `callout_outbound`, for example, transforms the Oracle E-Business Suite XML format back into a RosettaNet XML-formatted message for the remote trading partner.

These two callouts are then associated with the two agreements created for this exchange, as follows:

- Include `callout_outbound` in the agreement for the outbound message, that is, the agreement for the initiating purchase order request.

- Include `callout_inbound` in the agreement for the inbound message, that is, the agreement for the responding purchase order acceptance.

Because a document definition is a component of an agreement, a callout is associated with a specific document definition.
This purchase order example depicts a simple association of one callout to one agreement. In reality, however, the same callout can be included in many different agreements by changing the value of one or more callout parameters. See Figure 12–3 for where you add parameters and see Table 12–2 for a list of parameter attributes.

12.1.1 Transport Callouts

Another type of callout is the transport callout, which is associated with a channel. For the inbound message, B2B invokes the transport callout immediately after it receives a message from the transport. For the outbound message, B2B invokes the transport callout immediately before it sends a message to the transport. Transport callouts can be selected in the channel configuration, as shown in Figure 12–2, and can be used with any protocol.

You can use transport callouts to extract custom headers for inbound and outbound messages using the MLLP protocol. Example 12–1 shows how to set and get the CUSTOM_HEADER property in the callout.

Example 12–1 Setting and Getting the CUSTOM_HEADER Property

```java
import java.util.*;
import oracle.tip.b2b.callout.*;
import oracle.tip.b2b.callout.exception.*;

public class SampleCallout implements Callout {
    public void execute(CalloutContext context, List input, List output)
        throws CalloutDomainException, CalloutSystemException {
        try {
            CalloutMessage cmIn = (CalloutMessage)input.get(0);
            String s = cmIn.getBodyAsString();
```
// for getting the CUSTOM_HEADER
Properties params = (Properties)cmIn.getParameters();
String customHeader = (String)params.get("CUSTOM_HEADER");

// for setting the CUSTOM_HEADER
CalloutMessage cmOut = new CalloutMessage(s);
    cmOut.setParameter("CUSTOM_HEADER", "your_value");
    output.add(cmOut);

} catch (Exception e) {
    throw new CalloutDomainException(e);
}
}

See Section 5.5.1.4, "Using a Transport Callout to Extract Custom Headers," for more information.

Transport callouts are created like other callouts, from the Callout tab, as described in Section 12.2, "Creating a Callout." Although a transport callout is not added to an agreement, all transport callouts appear in the Callouts list on the Agreement tab; therefore, it is available for selection. To avoid confusion, when you create a transport callout, provide a name that indicates its type so that you do not select it from the Callouts list on the Agreement tab.

### 12.1.2 Creating a Callout Library JAR File

If the callout JAR file provided with Oracle B2B is not sufficient for your needs, you can create your own callout JAR file outside of Oracle B2B, following the standards described in the Oracle Fusion Middleware B2B Callout Java API Reference. Use the Configuration tab of the Administration link to specify the directory location of this external JAR file. It is recommended that you create an external JAR file for your callouts; do not bundle your callouts with b2b.jar.

**Note:** MySampleCallout is a restricted keyword and should not be used. It is already packaged into b2b.jar.

### 12.2 Creating a Callout

To create a callout, provide callout details—the implementation class name and library name—and callout parameters, as shown in Figure 12–3.
You can create multiple callouts with the same name if you assign them different implementation names. You cannot delete a callout that is included in an agreement.

Table 12–1 lists the callout details that you provide.

**Table 12–1 Callout Details**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Implementation Class</td>
<td>Enter the class file name without .class.</td>
</tr>
</tbody>
</table>

**Note:** Oracle B2B includes a predefined class file named XSLTCalloutImpl that you can use for XML-to-XML transformations.
Callout parameters are similar in concept to global variables to which you can assign local values that are applicable only to a specific callout use. Or, you can create a callout parameter and assign it a default value that is applicable to all callout uses. Changes to callout parameters for an existing callout affect all agreements that use that callout.

Table 12–2 lists the optional callout parameter attributes.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a parameter name.</td>
</tr>
<tr>
<td>Type</td>
<td>Select from Integer, Float, String, Boolean, or Date types. The format for the Date type is MM/DD/YYYY. Note: Changing a type can invalidate the parameter default value.</td>
</tr>
<tr>
<td>Value</td>
<td>Enter a value. If Encrypted is set to True, then this value is encrypted.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Select True or False.</td>
</tr>
<tr>
<td>Encrypted</td>
<td>Select True or False.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter an optional description.</td>
</tr>
</tbody>
</table>

After you create a callout, it is available to include in an agreement. See Section 12.3, "Including a Callout in an Agreement," for more information. If you change a callout after it is deployed with an agreement, a server restart is required.
To create a callout:
1. Click Administration, and then Callout.
2. In the Callout section, click Add.
3. Enter a name for the callout.
   (You may want to indicate if you are creating a transport callout in the name.)
4. Enter callout details, as described in Table 12–1.
5. (Optional) Click Add in the Parameters section.
6. Enter a parameter name and attributes, as described in Table 12–2.
7. Click Save.
   You can edit the details, parameters, or parameter values at any time, but not the callout name.

### 12.3 Including a Callout in an Agreement

After you create a callout, it is available to include in an agreement, as shown in Figure 12–4.

**Figure 12–4 Specifying a Callout in an Agreement**

To include a callout in an agreement:
1. Click Partners.
2. Click an agreement name.
3. Select a callout.
4. Click Save.

To update the value of a callout parameter for a specific agreement:
1. Click Partners.
2. Click an agreement name.
3. Select a callout.
4. Click Callout Details.
5. Enter a value for the parameter name, as shown in Figure 12–5.
Figure 12–5  Entering Callout Details

<table>
<thead>
<tr>
<th>Callout Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Implementation Class</td>
</tr>
<tr>
<td>Library Name</td>
</tr>
<tr>
<td>Timeout</td>
</tr>
</tbody>
</table>

6. Click OK.

12.4 Implementing a Callout

Example 12–2 shows how an incoming XML document is transformed to another XML document. The directory structure is `oracle.tip.callout`. In this example, note that setting the output CalloutMessage in the output list is required (output.add(cmOut)).

Example 12–2  Code Example of an XML-to-XML Transformation

```java
import java.io.*;
import java.net.*;
import java.util.*;
import oracle.xml.parser.v2.*;
import oracle.tip.b2b.callout.Callout;
import oracle.tip.b2b.callout.CalloutMessage;
import oracle.tip.b2b.callout.CalloutContext;
import oracle.tip.b2b.callout.exception.*;

/**
 * This sample callout transforms the incoming XML document
 * to another XML document. It also shows how to generate
 * Functional Ack and Error message.
 */
public class XSLTCalloutImpl implements Callout {
    public void execute(CalloutContext context,
                        List input,
                        List output)
        throws CalloutDomainException, CalloutSystemException {
        try {
            // (1) Retrieve the callout properties from CalloutContext
            String xsltFile = context.getStringProperty("xsltFile");

            // (2) Get the input callout message
            CalloutMessage cmIn = (CalloutMessage)input.get(0);

            // (3) Process the message
```
// instantiate a stylesheet
URL xslURL = new URL("file:///" + xsltFile);
XSLProcessor processor = new XSLProcessor();
XSLStylesheet xsl = processor.newXSLStylesheet(xslURL);

// parser input XML content
DOMParser parser = new DOMParser();
parser.setPreserveWhitespace(true);
parser.parse(new StringReader(cmIn.getBodyAsString()));
XMLDocument xml = parser.getDocument();
processor.showWarnings(true);
processor.setErrorStream(System.err);

// Transform the document
StringWriter strWriter = new StringWriter();
processor.processXSL(xsl, xml, new PrintWriter(strWriter));

// (4) Create a output callout message
CalloutMessage cmOut =
    new CalloutMessage(strWriter.getBuffer().toString());
strWriter.close();

// create Functional Ack callout message
// this is an optional step
CalloutMessage fa = new CalloutMessage(*set FA payload here*);
fa.setParameter("functional_ack", "true");
// setting your own doctype and revision
// set the doc type name and revision as defined in b2b ui
fa.setParameter("doctype_name", "fa");
fa.setParameter("doctype_revision", "1.0");

// create Error callout message
// this is an optional step
CalloutMessage err = new CalloutMessage(* set the payload that causes this error */);
err.setParameter("error_message", "true");
err.setParameter("error_desc", "set the error desc");

    output.add(cmOut);
    output.add(fa);
    output.add(err);

    // (5) Throw an exception, if any
    catch (Exception e) {
        throw new CalloutDomainException(e);
    }
}
Use the Oracle B2B interface to purge design metadata and instance data. This chapter contains the following topics:

- Section 13.1, "Purging Design Metadata and Instance Data"

See the following for alternate methods of purging:

- Chapter 18, "B2B Command-Line Tools"

### 13.1 Purging Design Metadata and Instance Data

Use the Oracle B2B interface to purge design metadata and instance data. Design metadata contains partner profile data, identifiers, document definitions, channels, and agreements. When you purge this data, predefined data that is part of the installation (the host trading partner name, protocols, and identification types, for example) is not purged. Instance data is created during run time when messages are processed. Instance, or run-time, data contains the business messages and message-related data.

Specific instance data can be purged from the **Business Message** tab of the **Reports** link. See Section 16.1.2, "Purging Messages," for more information.

Purging does not remove artifacts that B2B creates in the Credential Store, such as passwords. See Oracle Fusion Middleware Security Guide for more information about the Credential Store.

With an instance message purge, you can optionally purge control number information. Control numbers are used in EDI (X12 and EDIFACT) and HL7 message standards. B2B keeps track of control numbers for inbound and outbound messages. For outbound messages, B2B generates the control numbers in a sequence from an internal control number table. Because purging instance data and control numbers resets the sequence (the control number table is reset), an outbound message after a purge may have the same control number as a message before the purge. If this is undesirable, do not purge control numbers.

Purging is useful for:

- Managing disk space and improving performance
- Removing repositories on a test system

---

**Caution:** Purging is an irreversible operation. Ensure that you first archive any important data.
To purge design metadata or instance data:

1. Click the Administration tab, and then the Purge tab.

2. (Optional if purging instance data.) Select Purge Control Number to reset the sequence.

3. Click Purge Design Metadata or Purge Instance Data, as shown in Figure 13–1.

Figure 13–1 Purging Design Metadata or Instance Data

If you select Purge Design Metadata, then the message Do you want to delete all the design metadata from the repository permanently? appears.

If you select Purge Instance Data, then the message Do you want to delete all the runtime data from the repository permanently? appears.

4. Click Yes.
A listening channel is used to send messages to Oracle B2B. A listening channel listens on an endpoint for messages. If a listening channel is marked as internal, then it can be used by any internal business application. If it is used as an external channel, then any trading partner can send a message to Oracle B2B using this channel.

This chapter contains the following topics:

- Section 14.1, "Adding a Listening Channel and Protocol"
- Section 14.2, "Using Transport Protocols"
- Section 14.3, "Adding Listening Channel Details"
- Section 14.4, "Configuring a Listening Channel"

### 14.1 Adding a Listening Channel and Protocol

Listening channels are used globally. You do not need to select a listening delivery channel in an agreement. Listening channels are used for any trading partner to send inbound messages to Oracle B2B or for any back-end business application to send outbound messages to Oracle B2B.

When you add a listening channel, you also specify the protocol that the channel uses, as shown in Figure 14–1.
By using a global listening channel, you can keep all messages in one directory from which Oracle B2B pulls. This approach is useful for File, FTP, and SFTP (SSH FTP) exchanges.

Table 14–1 describes the listening channel protocols supported by Oracle B2B.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS1-1.0</td>
<td>Applicability Statement 1 (AS1) provides S/MIME and uses SMTP to transmit data using e-mail. Security, authentication, message integrity, and privacy are assured by the use of encryption and digital signatures. Use nonrepudiation to make it impossible for the intended recipient of a message to deny having received it. AS1 works with almost any type of data.</td>
</tr>
<tr>
<td>Generic File-1.0, Generic AQ-1.0, Generic FTP-1.0, Generic SFTP-1.0, Generic JMS-1.0, Generic Email-1.0</td>
<td>Using the Generic options, you can send messages with or without security. The Generic exchange protocol supports MIME and S/MIME, including S/MIME 3.0-based signing and encryption. There is no receipt acknowledgment support with the Generic protocols (the acknowledgment mode must be set to None).</td>
</tr>
</tbody>
</table>
14.2 Using Transport Protocols

The transport protocol used to send the message is determined by the listening channel you select, as shown in the Channel Details area in Figure 14–2.

Figure 14–2 Channel Details: The Transport Protocol

Table 14–2 describes the transport protocols available in Oracle B2B.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>Use Email for AS1 and Email listening channels.</td>
</tr>
<tr>
<td>File</td>
<td>The File transport enables files to be picked up from a shared file directory.</td>
</tr>
<tr>
<td>AQ</td>
<td>Oracle AQ provides secure, bidirectional, asynchronous communication. The location of the application location is transparent, using any number of Oracle connectivity options, including OCI, JDBC, or PL/SQL. Both XML and non-XML message payloads are supported.</td>
</tr>
<tr>
<td>FTP</td>
<td>FTP enables files to be passed with FTP between applications. FTP runs on default port 21. To change to another port, provide the value in the Control Port field. To enable SSL, use the Channel Mask field. The default is None (no SSL).</td>
</tr>
<tr>
<td>SFTP</td>
<td>SFTP enables files to be passed using SSH FTP. SFTP runs on default port 22, which can be changed to another port. SFTP supports two modes of authentication, password authentication and public key authentication. To use password authentication, provide a password, which is used for authentication. To use public key authentication, provide the private key file location. You may also need to provide a pass phrase if the private key file is pass-phrase protected.</td>
</tr>
<tr>
<td>JMS</td>
<td>JMS enables applications to send and receive messages to and from the queues and topics administered by any Java Message Service (JMS) provider, including Oracle WebLogic JMS and non-Oracle providers such as MQSeries JMS (IBM). If a user name and password are not provided, the local JNDI is used, including in a clustered environment, provided that the destinations are distributed. Oracle B2B does not support javax.jms.ObjectMessage.</td>
</tr>
</tbody>
</table>
14.3 Adding Listening Channel Details

Listening channel details include transport protocol parameters, channel attributes, exchange protocol parameters, and security specifications.

14.3.1 Transport Protocol Parameters

A transport protocol defines the properties specific to a given use of a protocol endpoint. The transport is responsible for message delivery using the selected transport protocol, mode (synchronous or asynchronous), server, and protocol endpoint address (the trading partner address, such as a URI). Table 14–3 describes the transport protocol parameters and lists the protocols to which the parameters apply.

<table>
<thead>
<tr>
<th>Protocol/Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archival Directory</td>
<td>B2B channels move the processed files to this directory. By default,</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>it is a destructive read—processed files are deleted from the endpoint.</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>In this case, files are moved to the path provided.</td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td>Cache Connections</td>
<td>If enabled, file listing and processing of the file occur in the same</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>session (contrary to the default, in which listing and processing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>occur in different sessions).</td>
<td></td>
</tr>
<tr>
<td>Channel mask</td>
<td>To enable SSL for FTP, enter one of the following:</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>■ Control—Encrypts the control channel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Data—Encrypts the data channel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Both—Encrypts both the data and control channels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The default is None (no SSL).</td>
<td></td>
</tr>
<tr>
<td>Cipher suites</td>
<td>Sets of ciphers defined in SSL.</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td>Connection factory</td>
<td>The JNDI location or Java class name for the connection factory, as in</td>
<td>Generic JMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>jms/b2b/B2BQueueConnectionFactory.</td>
<td></td>
</tr>
<tr>
<td>Consumer</td>
<td>The client that receives the message.</td>
<td>Generic AQ-1.0 (optional)</td>
</tr>
<tr>
<td>Content type</td>
<td>The content type of the payload being sent over e-mail. The default content</td>
<td>AS1-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>type is text/plain; other examples include application/xml and</td>
<td>Generic Email-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>application/edi. This value is used only for the delivery channel (to send</td>
<td></td>
</tr>
<tr>
<td></td>
<td>e-mail) and not for the listening channel. On the listening channel side,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>intelligence is built into the transport adapter to deal with different</td>
<td></td>
</tr>
<tr>
<td></td>
<td>content types, so no configuration is required.</td>
<td></td>
</tr>
<tr>
<td>Control port</td>
<td>Provide a value to change the default FTP port value (21)</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td>Data port</td>
<td>For active FTP connections, use this option to configure the static/fixed</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>data port of the FTP server.</td>
<td></td>
</tr>
<tr>
<td>Datasource</td>
<td>The JNDI name of the JDBC data source to access AQ queues.</td>
<td>Generic AQ-1.0 (optional)</td>
</tr>
<tr>
<td>Destination name</td>
<td>The JMS destination name.</td>
<td>Generic JMS-1.0 (optional)</td>
</tr>
<tr>
<td>Destination Provider</td>
<td>Enables B2B to connect to JMS queues or topics available on remote servers.</td>
<td>Generic JMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>JNDI properties required to connected to the target server are</td>
<td></td>
</tr>
<tr>
<td></td>
<td>expected as the value. Use ; (semicolon) as the separator for each key/value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pair.</td>
<td></td>
</tr>
<tr>
<td>Email ID</td>
<td>The e-mail address to which messages are delivered (similar to specifying</td>
<td>AS1-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td>the path for a file channel or queues in AQ or JMS).</td>
<td>Generic Email-1.0 (required)</td>
</tr>
<tr>
<td>Email Server</td>
<td>Select IMAP or POP3.</td>
<td>AS1-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td>Select IMAP or POP3.</td>
<td>Generic Email-1.0 (required)</td>
</tr>
<tr>
<td>Enable CCC</td>
<td>Enables B2B to authenticate in an SSL session and do the rest of the file</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>transfer commands on a plain socket.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 14–3 (Cont.) Transport Protocol Parameters

<table>
<thead>
<tr>
<th>Protocol/Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Marker</td>
<td>If enabled, creates a zero-byte file with the same name as the source, indicating completion of reading or writing. The file carries the same name as the source, but with the extension marker.</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (optional)-1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td>Encoding</td>
<td>The encoding used in B2B to convert the contents of the inbound files.</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td>Filename format</td>
<td>The following filename formats can be used:</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>%FROM_PARTY%</td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>%TO_PARTY%</td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td>%DOCTYPE_NAME%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%DOCTYPE_REVISION%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%MSG_ID%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%TIMESTAMP%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This filename format can be used for ebMS documents only:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%ACTIONNAME%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>These formats can be used in any combination; for example,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>%TO_PARTY%<em>%DOCTYPE_NAME%</em>%DOCTYPE_REVISION%.dat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>produces something like Acme_4010_850.dat. Any file extension is allowed.</td>
<td></td>
</tr>
<tr>
<td>Folder</td>
<td>An absolute directory path is recommended.</td>
<td>ASI-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email-1.0 (optional)</td>
</tr>
<tr>
<td>Folder name</td>
<td>An absolute directory path is recommended.</td>
<td>Generic File-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (required)</td>
</tr>
<tr>
<td>Host name</td>
<td>The trading partner’s transport or e-mail server exchanging messages.</td>
<td>ASI-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email-1.0 (required)</td>
</tr>
<tr>
<td>Is Binary</td>
<td>Treats the message as binary content, with no translation or validation.</td>
<td>This parameter is not available with Generic File-1.0, Generic FTP-1.0, and Generic SFTP-1.0, although it appears in the B2B interface for these protocols.</td>
</tr>
<tr>
<td>Minimum Age</td>
<td>Files arriving at the endpoint are processed after the time interval entered, in milliseconds.</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td>Pass phrase and Confirm pass phrase</td>
<td>If you enter a private key file location, and if the private key file is pass-phrase protected, then enter the pass phrase.</td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td>Password and Confirm Password</td>
<td>To use password authentication, provide a key store password, which is used for authentication.</td>
<td>ASI-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic JMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email-1.0 (optional)</td>
</tr>
</tbody>
</table>
### Table 14–3  (Cont.) Transport Protocol Parameters

<table>
<thead>
<tr>
<th>Protocol/Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path</td>
<td>The absolute directory path where messages are sent from or received.</td>
<td>Generic SFTP-1.0 (required)</td>
</tr>
</tbody>
</table>
| Polling interval   | The time interval in seconds during which Oracle B2B polls the server for inbound messages. | AS1-1.0 (optional)  
|                    | | Generic File-1.0 (optional)  
|                    | | Generic AQ-1.0 (optional)  
|                    | | Generic FTP-1.0 (optional)  
|                    | | Generic SFTP-1.0 (optional)  
|                    | | Generic JMS-1.0 (optional)  
|                    | | Generic Email-1.0 (optional)  |
| Port number        | AQ runs on default port 1521.  
|                    | SFTP runs on default port 22, which can be changed to another port.  
|                    | FTP runs on default port 21, which is not displayed. See the description of Control Port for how to change this port number. | Generic AQ-1.0 (optional)  
|                    | | Generic SFTP-1.0 (required)  |
| Preserve Filename  | Retains the file name. | Generic File-1.0 (optional)  
|                    | | Generic FTP-1.0 (optional)  
|                    | | Generic SFTP-1.0 (optional)  |
| Private key        | To use public key authentication, provide the private key file location. You may also need to provide a pass phrase if the private key file is pass-phrase protected. | Generic SFTP-1.0 (optional)  |
| Queue name         | The AQ queue name. | Generic AQ-1.0 (optional)  |
| Recipient          | The value used when delivering a message to the AQ queue. For example, if you set the recipient to testuser, then the message can be consumed only by the consumer with the name testuser (in other words, the recipient is on the sending side and the consumer is on the listening side). | Generic AQ-1.0 (optional)  |
| Send as attachment | If enabled, the message (payload) is sent as an e-mail attachment instead of the typical delivery in which the payload is the message body. | This parameter is not available with AS1-1.0 and Generic Email-1.0, although it appears in the B2B interface for these protocols.  
| SID                | System ID to identify an Oracle database. | Generic AQ-1.0 (optional)  |
| Subject            | The subject header of the e-mail message. | This parameter is not available with AS1-1.0 and Generic Email-1.0, although it appears in the B2B interface for these protocols.  
| Subscriber ID      | The JMS subscriber ID is required if JMS is communicating with a topic. | Generic JMS-1.0  |
| Transfer Type      | Select binary or ascii for the file transfer mode. | Generic FTP-1.0 (optional)  |
| Use JMS ID         | Uses the JMS message ID as the B2B message ID. This facilitates correlation at the JMS level. | Generic JMS-1.0 (optional)  |
| Use proxy          | Select this option if a proxy server is used. | Generic FTP-1.0 (optional)  
|                    | | Generic SFTP-1.0 (optional)  |
| User name          | The user name (login name) to connect to the target servers. This value is optional for AQ and JMS because B2B can use the configured JNDI data sources to connect to queues. | AS1-1.0 (required)  
|                    | | Generic AQ-1.0 (optional)  
|                    | | Generic FTP-1.0 (required)  
|                    | | Generic SFTP-1.0 (required)  
|                    | | Generic JMS-1.0 (optional)  
|                    | | Generic Email-1.0 (required)  |
14.3.2 Channel Attributes

The channel is the communication interface between the host trading partner’s host application and its installation. **Table 14–4** describes the channel attributes and lists the protocols to which the attributes apply.

<table>
<thead>
<tr>
<th>Protocol/Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ack Mode</td>
<td>Select <strong>Sync</strong>, <strong>Async</strong>, or <strong>None</strong> for the mode in which the trading partner receives messages. Select <strong>None</strong> for all generic exchanges.</td>
<td>This parameter is <em>not</em> available with AS1-1.0, although it appears in the B2B interface for this protocols.</td>
</tr>
<tr>
<td>Description</td>
<td>Provide an optional description.</td>
<td>AS1-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic JMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email-1.0 (optional)</td>
</tr>
<tr>
<td>Enable/Disable Channel</td>
<td>The channel is the communication interface between the host trading partner’s host application and its installation.</td>
<td>AS1-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic File-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic JMS-1.0 (required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email-1.0 (Required)</td>
</tr>
<tr>
<td>Internal</td>
<td>Select this option if the channel is internal to the host trading partner’s enterprise. This feature is disabled for AS1.</td>
<td>Generic File-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic AQ-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic FTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic SFTP-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic JMS-1.0 (optional)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Email-1.0 (optional)</td>
</tr>
<tr>
<td>Response Mode</td>
<td>Select <strong>Sync</strong>, <strong>Async</strong>, or <strong>None</strong>,</td>
<td>This parameter is <em>not</em> available with AS1-1.0, although it appears in the B2B interface for this protocols.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
14.3.3 Exchange Protocol Parameters

The exchange protocol defines the headers, acknowledgments, and packaging that puts the headers and payload together (the message exchange mechanism). The exchange protocol also defines signing and compression. Table 14–5 describes the exchange protocol parameters and lists the protocols to which the parameters apply.

<table>
<thead>
<tr>
<th>Protocol/Parameter</th>
<th>Description</th>
<th>Protocol Used With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signed and Compressed</td>
<td>If selected, the message is first signed, and then compressed. If <em>not</em> selected, the message is first compressed, and then signed.</td>
<td>AS1-1.0 (optional)</td>
</tr>
</tbody>
</table>

Table 14–5 Exchange Protocol Parameters

14.3.4 Security Parameters

Security parameters are not available for any of the protocols, although the B2B interface displays security parameters for the AS1-1.0 protocol, as described in Table 14–6.
14.4 Configuring a Listening Channel

To configure a listening channel, add a listening channel protocol, and then transport protocol parameters, channel attributes, exchange protocol parameters, and security parameters, depending on the channel protocol you selected.

To add a listening channel protocol:
1. Click the Administration link.
2. Click the Listening Channel tab.
3. Click Add.
4. Provide a name for the listening channel.
5. Select a protocol.

Figure 14–3 shows the list of protocols.
See Table 14–1 for a description of the protocols.

The transport protocol that appears under Channel Details is based on your protocol selection in Step 5.

6. Click Save.

**To add transport protocol parameters:**

1. Click the Transport Protocol Parameters tab.

2. Provide transport protocol parameters, depending on the channel/transport protocols.

   Table 14–3 describes the transport protocol parameters (listed in alphabetical order) and the protocols to which the parameters apply.

3. Click Save.

**To add channel attributes:**

1. Click the Channel Attributes tab.

2. Provide channel attributes, depending on the channel/transport protocols selected.

   Table 14–4 describes the channel attributes (listed in alphabetical order) and the protocols to which the attributes apply.

3. Click Save.
To add exchange protocol parameters:

1. Click the Exchange Protocol Parameters tab.

2. Provide exchange protocol parameters, depending on the channel/transport protocols selected.
   
   Table 14–5 describes the exchange protocol parameters (listed in alphabetical order) and the protocols to which the attributes apply.

3. Click Save.

This chapter contains the following topics:
- Section 15.1, "Setting Configuration Parameters"

### 15.1 Setting Configuration Parameters

Table 15–1 shows the configuration settings available in the Oracle B2B interface.
Table 15-1 describes the configuration parameters.
### Table 15–1 Configuration Settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgment</td>
<td>-</td>
</tr>
<tr>
<td>Functional Ack Handled by B2B</td>
<td>If set to true, then B2B autogenerates the functional acknowledgment (FA) message for inbound EDI and HL7 messages. Inbound FA messages are consumed when this option is true. When this option is set to false, B2B does not autogenerate the FA document. The back-end application (middleware) must generate the FA and provide it to B2B as an outbound message. When option is set to false, inbound FA documents are passed back to the back-end application. If the document does not require an FA (as indicated by the agreement-level setting), then this option is ignored. The default value for this property is true. See Section B.1, “Properties To Set in Fusion Middleware Control,” for more information. When Functional Ack Handled by B2B is set to false, then Notify Inbound Functional Acks must be set to false also for the inbound FA to be sent to the back-end application. If Notify Inbound Functional Acks is set to true (while Functional Ack Handled by B2B is set to false), then the incoming 997 (FA doc) generates only a notification and the 997 document itself is not sent back to the back-end application.</td>
</tr>
<tr>
<td>Functional Ack Internal Properties</td>
<td>Generates the internal properties structure in the functional acknowledgment XML for EDI transactions. A document type 997 (for X12) or CONTRL (for EDIFACT) must exist. The default value is false, which means that the functional acknowledgment uses the original message-internal properties. If true, then the FA message autogenerated by B2B contains interchange/group envelope information from the original message.</td>
</tr>
<tr>
<td>Notify Inbound Receipt Acks</td>
<td>If set to true, B2B sends an acknowledgment notification to the application when an exchange acknowledgment is received.</td>
</tr>
<tr>
<td>Notify Inbound Functional Acks</td>
<td>If set to true, B2B sends an acknowledgment notification to the application when a functional acknowledgment is received. When Functional Ack Handled by B2B is set to false, then Notify Inbound Functional Acks must be set to false also for the inbound FA to be sent to the back-end application. If Notify Inbound Functional Acks is set to true (while Functional Ack Handled by B2B is set to false), then the incoming 997 (FA doc) generates only a notification and the 997 document itself is not sent back to the back-end application.</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>-</td>
</tr>
<tr>
<td>Default Trading Partner</td>
<td>Defaults to this trading partner if trading partner agreement identification fails. Used for HL7 documents.</td>
</tr>
<tr>
<td>Ignore Correlation</td>
<td>When an acknowledgment is received from a trading partner, it is correlated to the actual business message of the sender. If the correlation fails, an exception is generated and the acknowledgment processing stops. To ignore the correlation and process the acknowledgment, set this property to true.</td>
</tr>
<tr>
<td>Log Payload</td>
<td>If true, B2B logs the payload in a diagnostic log (also depends on log level setting). Error messages are logged by default. Payload logging is useful for diagnostic purposes, but may be undesirable for security reasons. The default value is false.</td>
</tr>
<tr>
<td>Reconnect on Error</td>
<td>If set to true, the AQ adapter retries the enqueue operation when the initial enqueue fails. This parameter is not available in this release.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HTTP Header Delimiter</td>
<td>A delimiter to separate the HTTP headers provided in the Additional Transport Headers field for HTTP delivery channel configuration.</td>
</tr>
<tr>
<td>Treat Reply to Message as Request</td>
<td>Used in ebMS to indicate that the conversation message is to be considered as a request message.</td>
</tr>
<tr>
<td>Generic Message Type</td>
<td>If this property is enabled (set to true), B2B finds the agreement for the specific message type first, and then the generic message type. The default value is false.</td>
</tr>
<tr>
<td>Miscellaneous (continued)</td>
<td></td>
</tr>
<tr>
<td>Outbound Dispatcher Count</td>
<td>The number of dispatchers used for handling the outbound messages. Used in message sequencing for MLLP. The default value is 0.</td>
</tr>
<tr>
<td>Inbound Dispatcher Count</td>
<td>The number of dispatchers used for handling the inbound messages. Used in message sequencing for MLLP. The default value is 0.</td>
</tr>
<tr>
<td>Auto Stack Handler</td>
<td>Used in stacking for MLLP. If true, the stack handler processes stacked messages in automatic mode. The default value is false.</td>
</tr>
<tr>
<td>Auto Stack Handler Interval</td>
<td>Used in stacking for MLLP. Enter comma-separated values for the time interval in seconds for the stack handler to process the stacked messages. The default value is 1.</td>
</tr>
<tr>
<td>Partial Batch Commit Size</td>
<td>Used while debatching an EDI message that has large transaction sets. Instead of using a single commit, which is prone to errors in case of failure, this feature enables specifying a partial batch commit size. For example, if Partial Batch Commit Size=10, then, for an envelope with more than 10 transactions, B2B calls a separate commit for every 10 transaction sets.</td>
</tr>
<tr>
<td>Exception Queue</td>
<td>Select a JMS internal delivery channel for the host trading partner to use as the exception queue. A null default value for this parameter means that exceptions are sent to the JMS queue (B2B_IN_QUEUE) if Use JMS Queue as default is set to true or to the AQ queue (IP_IN_QUEUE) if Use JMS Queue as default is set to false. AQ queues are not supported for use as custom exception queues.</td>
</tr>
<tr>
<td>Non Purgeable</td>
<td>The nonpurgeable parameters—Use JMS Queue as default and Callout Directory—retain their values even after a metadata repository purge is invoked.</td>
</tr>
</tbody>
</table>
| Use JMS Queue as default | If this option is set to true, then B2B starts to poll on the JMS queue, B2B_OUT_QUEUE for outbound messages, and delivers all inbound messages to B2B_IN_QUEUE. Polling on IP_OUT_QUEUE is stopped.  
If this option is set to false (the default), then B2B starts to poll on the AQ queue, IP_OUT_QUEUE for outbound messages, and delivers all inbound messages to IP_IN_QUEUE. Polling on B2B_OUT_QUEUE is stopped. When a non-Oracle database is used and therefore no AQ queues are available, the JMS queues are used instead. This option is set.  
If you select a queue from Exception Queue, then exception messages are sent to that configured queue. The default queues continue to be valid for other messages, depending on the setting for Use JMS Queue as default.  
If the value of Use JMS Queue as default is set to true before purging the metadata repository, then after a purge, the value continues to be true and does not revert back to the default value, false. |
| Callout Directory     | Specify a directory for the callout JAR file location if you do not use the default callout. The callout directory path cannot end with / or \. The default file location, /MyCalloutDir, is retained after purging the metadata repository. |
| SMTP Host             | Specify the host name of the SMTP server in the enterprise to send the negative MDN to the trading partner for an AS1 exchange.              |
| Performance           |                                                                                                                                           |
| Large Payload Size    | Specify a large payload size, in bytes. The default value is 2,000,000 (2MG).                                                               |
To set configuration parameters:

1. Click the Administration link.
2. Click the Configuration tab.
3. Provide values for the configuration parameters, as described in Table 15–1.
4. Click Save.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large Payload Directory</td>
<td>The default directory is /tmp. For Windows-based systems, change the directory to an appropriate directory, such as C: \ temp.</td>
</tr>
<tr>
<td>UI</td>
<td>-</td>
</tr>
<tr>
<td>Show Payload</td>
<td>Enables the payload to be displayed in reports accessible from the Reports tab. If set to true, the database is automatically searched with the default search parameters and the results are displayed.</td>
</tr>
<tr>
<td>Enable Auto Search</td>
<td>Enables automatic searching in reports accessible from the Reports tab. The default value is true. If set to false, a blank result table is displayed on the report pages until the Search button is clicked.</td>
</tr>
<tr>
<td>Payload Display Size</td>
<td>The default value is 1,048,576 KB. This parameter (in bytes) is used to display the payload only if its size is less than the value configured in the interface.</td>
</tr>
</tbody>
</table>
Part IV
Reports and Metrics

This part contains the following chapters:

- Chapter 16, "Creating Reports"
- Chapter 17, "Using B2B Metrics"
Oracle B2B reports provide real-time status on the run-time behavior of deployed data.

This chapter contains the following topics:

- Section 16.1, "Introduction to Reports"
- Section 16.2, "Creating Business Message Reports"
- Section 16.3, "Creating Wire Message Reports"
- Section 16.4, "Creating Application Message Reports"
- Section 16.5, "Creating Error Reports"
- Section 16.6, "Creating Conversation Reports"

### 16.1 Introduction to Reports

Use the Reports link to search on data in the run-time repository. The Saved Search function is not available.

The following message types are available for searching:

- Business messages—See Section 16.2, "Creating Business Message Reports"
- Wire messages—See Section 16.3, "Creating Wire Message Reports"
- Application messages—See Section 16.4, "Creating Application Message Reports"
- Error messages—See Section 16.5, "Creating Error Reports"
- Conversation messages—See Section 16.6, "Creating Conversation Reports"

---

**Note:** In a cluster environment, if system time stamps are not synchronized for all nodes in the cluster, then you may see message time stamps that look incorrect, but are not. For example, given an unsynchronized, multinode cluster, if an outbound message is received on one node, but the reply is sent from another node, it is possible for a report to show message receipt at 4 a.m., but an acknowledgment sent at 3:55 a.m.

---

### 16.1.1 The Monitor User Role

For individuals such as business analysts who create and analyze message reports, Oracle B2B provides a monitor user role that an administrator can assign to trading partner users. This role provides a user with access to only the functionality of the Reports tab of Oracle B2B. A user with the Monitor role cannot see or access the other
parts of the interface or see data for other trading partners. See Section 5.3, "Adding Trading Partner Users," for how to assign the Monitor role.

16.1.2 Purging Messages

From the Business Message tab, use the Purge button to purge one or more messages that display after you search the instance data.

16.1.3 Resubmitting Messages from Oracle B2B

If errors that occur when sending an inbound or outbound message are internal to Oracle B2B, then you can correct the problem and resend the message. For example, if B2B attempts to send a message to an endpoint that is not configured correctly, or if the agreement is not configured correctly, correct the error and use Resubmit for application messages or wire messages.

Resubmitting an application message, for an outbound message, replays the message from the time of receipt of the message and goes through agreement lookup, message translation (for EDI) and then finally the delivery is attempted. An application message resubmit is helpful when the agreement settings or document configuration is not as required and the message must be restructured with updated settings.

Resubmitting an application message, for an inbound message, attempts to deliver the message again to the back-end application. Resubmitting is useful when the back-end application is down and the delivery must be retried.

Resubmitting a wire message, for an outbound message, tries to redeliver only the previously processed message. There is no repackaging or other message transformation. This is helpful when the problem was with the delivery endpoint (for example, the partner's server is down and unable to receive the message).

Resubmitting a wire message, for an inbound message, replays the message from the time of receipt from the trading partner. The exchange and document are re-identified and an agreement lookup is done. The processed message is then delivered to the back-end. This is useful when the agreement or document setting are not correct and the message must be translated and validated again.

Note: If you resubmit an inbound AS2 synchronous wire message, the MDN is generated, but it is not returned to the sender in synchronous mode. This is because the sender is not the one who is initiating the originating message. In this scenario, the MDN message state is in the MSG_COMPLETE state.

16.2 Creating Business Message Reports

Business message status reports identify business message instance details for a document protocol. These details include the sending and receiving trading partners, the agreement name, the business action, the business message ID, the status, the exchange protocol and document protocol, and message details.

Figure 16–1 shows a business message report.
Figure 16–1  Business Message Report

To create a business message report:
1. Click Reports, and then Business Message.
2. Provide search parameters.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match</td>
<td>Select All or Any.</td>
</tr>
<tr>
<td>Sender</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of a trading partner name.</td>
</tr>
<tr>
<td>Receiver</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of a trading partner name.</td>
</tr>
<tr>
<td>Agreement</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of a trading partner agreement name.</td>
</tr>
<tr>
<td>Send Time Stamp</td>
<td>Select from Less Than, Greater Than, Greater Than Equals, Equals, or Less Than Equals. Provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the Select Date and Time icon.</td>
</tr>
<tr>
<td>Receive Time Stamp</td>
<td>Select from Less Than, Greater Than, Greater Than Equals, Equals, or Less Than Equals. Provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the Select Date and Time icon.</td>
</tr>
</tbody>
</table>
Creating Business Message Reports

3. To add more search fields, click Advanced and select from Add Fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of a message state: MSG_COMPLETE, MSG_ERROR, MSG_WAIT_TRANSMIT, MSG_WAIT_FA, MSG_WAIT_BATCH</td>
</tr>
<tr>
<td>Message ID</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of a message ID.</td>
</tr>
</tbody>
</table>

Field Description

- **Document Protocol Name**: Enter Custom, EDI_EDIFACT, EDI_X2, HL7, OAG, PositionalFlatFile, RosettaNet, or UCCNet. (Equals is the only operator.)
- **Document Type**: Select from a previously created document type, for example, 850 for EDI X12. (Equals is the only operator.)
- **Document Protocol Version**: Select from a previously created document protocol version. (Equals is the only operator.)
- **Document Definition**: Select from a previously created document definition. (Equals is the only operator.)

Use the document search parameters as follows: Select a document protocol name first to populate the list of document protocol versions; next select a document protocol version to populate the list of document types; and then select a document type to populate the list of document definitions.

4. Click Search.

View the results, as shown in Figure 16–1.

5. In the Details column of the Results area, click the icon to see report details.

Figure 16–2 shows the business message details.
16.3 Creating Wire Message Reports

Wire messages are the native format of data sent from trading partners. Wire messages can contain several sections, such as payloads, attachments, or trailers. Wire message status reports identify details about wire message instances, such as the transport protocol name, the transport protocol revision, and the protocol message identification and its state. The reports enable you to go from a business message to its corresponding wire message and from a wire message to its corresponding business messages.

Figure 16–3 shows a wire message report.
Creating Wire Message Reports

To create a wire message report:

1. Click Reports, and then Wire Message.

2. Provide search parameters.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of the URL.</td>
</tr>
<tr>
<td>Transport Protocol</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of the transport protocol.</td>
</tr>
<tr>
<td>State</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of a message state: MSG_COMPLETE, MSG_ERROR, MSG_WAIT_TRANSMIT, MSG_WAIT_FA, MSG_WAIT_BATCH</td>
</tr>
<tr>
<td>Created Date</td>
<td>Select from Less Than, Greater Than, Greater Than Equals, Equals, or Less Than Equals. Provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the Select Date and Time icon.</td>
</tr>
<tr>
<td>Message ID</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of a message ID.</td>
</tr>
</tbody>
</table>

3. To add more search fields, click Advanced and select from Add Fields.
4. Click Search.

View the results, as shown in Figure 16–3.

5. In the Details column of the Results area, click the icon to see report details.

Figure 16–4 shows wire message details.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Protocol Name</td>
<td>Select from Custom, EDI_EDIFACT, EDI_X2, HL7, OAG, PositionalFlatFile, RosettaNet, or UCCNet. (Equals is the only operator.)</td>
</tr>
<tr>
<td>Document Type</td>
<td>Select from a previously created document type, for example, 850 for EDI X12. (Equals is the only operator.)</td>
</tr>
<tr>
<td>Document Protocol Version</td>
<td>Select from a previously created document protocol version. (Equals is the only operator.)</td>
</tr>
<tr>
<td>Document Definition</td>
<td>Select from a previously created document definition. (Equals is the only operator.)</td>
</tr>
</tbody>
</table>

16.4 Creating Application Message Reports

This report provides information related to the SOA Composite—the name, version, and so on, if a back-end composite application sent or received the message.
Figure 16–5 shows an application message report.

**Figure 16–5 Application Message Report**

To create an application message report:

1. Click Reports, and then Application Message.
2. Provide search parameters.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match</td>
<td>Select All or Any.</td>
</tr>
<tr>
<td>Created Date</td>
<td>Select from Less Than, Greater Than, Greater Than Equals, Equals, or Less Than Equals. Provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the Select Date and Time icon.</td>
</tr>
<tr>
<td>Document Protocol Name</td>
<td>Select from Custom, EDI_EDIFACT, EDI_X2, HL7, OAG, PositionalFlatFile, RosettaNet, or UCCNet. (Equals is the only operator.)</td>
</tr>
<tr>
<td>Document Protocol Version</td>
<td>Select from a previously created document protocol version. (Equals is the only operator.)</td>
</tr>
<tr>
<td>Document Type</td>
<td>Select from a previously created document type, for example, 850 for EDI X12. (Equals is the only operator.)</td>
</tr>
<tr>
<td>Document Definition</td>
<td>Select from a previously created document definition. (Equals is the only operator.)</td>
</tr>
<tr>
<td>State</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of a message state: MSG_COMPLETE, MSG_ERROR, MSG_WAIT_TRANSMIT, MSG_WAIT_FA, MSG_WAIT_BATCH</td>
</tr>
</tbody>
</table>
3. To add more search fields, click **Advanced** and select from **Add Fields**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composite Name</td>
<td>Select from <em>Starts With, Equals, Contains, or Ends With</em>. Provide all or part of the SOA composite application name.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Name</td>
<td>Provide the name of the application.</td>
</tr>
<tr>
<td>Composite Version</td>
<td>Provide the version of the SOA composite application in Oracle JDeveloper.</td>
</tr>
<tr>
<td>ECID</td>
<td>Select from <em>Starts With, Equals, Contains, or Ends With</em>. Provide an instance ID.</td>
</tr>
<tr>
<td>Sender ID Type</td>
<td>Provide the sender’s identifier type, such as Name, DUNS, or MLLP ID.</td>
</tr>
<tr>
<td>Service Name</td>
<td>Provide the name of the B2B service binding component.</td>
</tr>
<tr>
<td>Receiver ID Type</td>
<td>Provide the receiver’s identifier type, such as Name, DUNS, or MLLP ID.</td>
</tr>
<tr>
<td>Receiver Value</td>
<td>Provide the value of the receiver’s identifier type. For example, if DUNS is the Receiver ID Type, provide the DUNS number.</td>
</tr>
<tr>
<td>Sender Value</td>
<td>Provide the value of the sender’s identifier type. For example, if Name is the Sender ID Type, provide the trading partner name as set in the identifier type in the trading partner’s profile.</td>
</tr>
<tr>
<td>Reference Name</td>
<td>Provide the name of the B2B reference binding component.</td>
</tr>
<tr>
<td>Fabric CompositeDn</td>
<td>Select from <em>Starts With, Equals, Contains, or Ends With</em>. Provide all or part of the composite name.</td>
</tr>
</tbody>
</table>

4. Click **Search**.

   View the results, as shown in **Figure 16–5**.

5. In the **Details** column of the **Results** area, click the icon to see report details.

   **Figure 16–6** shows application message details.
16.5 Creating Error Reports

Error status reports provide error message details. These details include the error code, error text, business message identification, message date, and message details.

Figure 16–7 shows an error report.
Creating Error Reports

1. Click Reports, and then Error.
2. Provide search parameters.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match</td>
<td>Select All or Any.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of an error code.</td>
</tr>
<tr>
<td>Error Level</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of an error level</td>
</tr>
<tr>
<td>Error Severity</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of an error severity.</td>
</tr>
<tr>
<td>Error Text</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of the error text.</td>
</tr>
<tr>
<td>Error Description</td>
<td>Select from Starts With, Equals, Contains, or Ends With. Provide all or part of the error description.</td>
</tr>
<tr>
<td>Send Time Stamp</td>
<td>Select from Less Than, Greater Than, Greater Than Equals, Equals, or Less Than Equals. Provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the Select Date and Time icon.</td>
</tr>
</tbody>
</table>

3. To add more search fields, click Advanced and select from Add Fields.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Definition</td>
<td>Select from a previously created document definition. (Equals is the only operator.)</td>
</tr>
</tbody>
</table>
Creating Conversation Reports

4. Click Search.

View the results, as shown in Figure 16–7.

5. In the Details column of the Results area, click the icon to see report details.

Figure 16–8 shows error report details.

Figure 16–8 Error Reports

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Type</td>
<td>Select from a previously created document type, for example, 850 for EDI X12. <em>(Equals is the only operator.)</em></td>
</tr>
<tr>
<td>Document Protocol Version</td>
<td>Select from a previously created document protocol version. <em>(Equals is the only operator.)</em></td>
</tr>
<tr>
<td>Document Protocol Name</td>
<td>Select from Custom, EDI_EDIFACT, EDI_X2, HL7, OAG, PositionalFlatFile, RosettaNet, or UCCNet. <em>(Equals is the only operator.)</em></td>
</tr>
</tbody>
</table>

16.6 Creating Conversation Reports

A conversation message results when the correlation XPath is set in a document definition to correlate messages. A correlation message also shows messages that are correlated automatically. For example, an AS2 message and its acknowledgment (MDN) are automatically correlated as part of a conversation. In RosettaNet, request
and response messages are also correlated, in addition to the acknowledgments sent and received. These related messages are displayed on the Conversation tab.

Figure 16–9 shows a conversation report.

**Figure 16–9 Conversation Report**

To create a conversation report:
1. Click Reports, and then Conversation.
2. Provide search parameters.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match</td>
<td>Select All or Any.</td>
</tr>
<tr>
<td>Send Time Stamp</td>
<td>Select from Less Than, Greater Than, Greater Than Equals, Equals, or Less Than Equals. Provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the Select Date and Time icon.</td>
</tr>
<tr>
<td>Collaboration Name</td>
<td>Applies to ebMS and RosettaNet documents and is available from header information.</td>
</tr>
<tr>
<td>Collaboration ID</td>
<td>Applies to ebMS and RosettaNet documents and is available from header information.</td>
</tr>
</tbody>
</table>

No additional fields can be added using the Advanced search button.

3. Click Search.

View the results, as shown in Figure 16–9.

4. In the Details column of the Results area, click the icon to see report details.

Figure 16–10 shows conversation report details.
Figure 16–10  Conversation Report Details
Oracle B2B metrics provide system-level and partner-level status on B2B run-time data. This includes status on messages and errors, message counts, active document types and trading partners, and error messages.

This chapter contains the following topics:

- Section 17.1, "Introduction to B2B Metrics"
- Section 17.2, "B2B System Metrics"
- Section 17.3, "B2B Partner Metrics"

17.1 Introduction to B2B Metrics

Use the Metrics tab to view current run-time data in the repository. The Metrics tab reflects changes that occur in the run-time repository (for example, purging the run-time instance data).

Metrics data shown in the Messages and Errors chart and the Message Count chart, shown in Figure 17–1, display data for the last 10 hours or the last 20 hours.

![The Messages and Errors Chart and Message Count Chart](image)

The metrics tables show all data from the time the first message was received. Current data is available by using the Refresh button. In contrast, changes are not immediately reflected in Oracle Enterprise Manager Fusion Middleware Control, which is based on dynamic monitoring service (DMS) metrics collected from the Weblogic managed server node. Enterprise Manager also shows limited information (the top 5 partners, the top 5 documents) and the data is available only from the last restart of the server. See Oracle Fusion Middleware Administrator’s Guide for Oracle SOA Suite for more information.

Most fields in the active document types, active trading partners, and errors tables can be sorted in ascending or descending order, as shown in Figure 17–2.
This is useful to identify the largest average message size or to group all the responding partner error messages, for example. You can resize columns to see any text that may be obscured. For error text, place the mouse over the text to see the entire message. The business message IDs in the Errors area link to business message details, as shown in Figure 17–3.

17.2 B2B System Metrics

Figure 17–4 shows system metrics summary data.
**Table 17–1** describes the information on the **System** metrics tab.

**Table 17–1  B2B System Metrics**

<table>
<thead>
<tr>
<th>Area/Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary</strong></td>
<td>Active partners are partners for which at least one agreement has been deployed. Active agreements are agreements that have been deployed and are in the active state. Active document types are document types that have been included in deployed and active agreements.</td>
</tr>
<tr>
<td><strong>Messages and Errors</strong></td>
<td>Processed messages = Completed messages + Errored messages Details of the errored messages are listed under <strong>Errors</strong>.</td>
</tr>
<tr>
<td><strong>Message Count</strong></td>
<td>Active messages are shown in this trend of inbound and outbound message quantity over time.</td>
</tr>
<tr>
<td><strong>Active Document Types</strong></td>
<td>Active document types are document types that have been included in active agreements. Details of the errors are listed under <strong>Errors</strong>. Messages processed include completed plus errored messages, that is, active messages.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Name of the document definition</td>
</tr>
<tr>
<td><strong>No. of Messages Processed</strong></td>
<td>Shows the number of document messages exchanged between the host and trading partners. <strong>Outbound</strong> indicates messages sent from the host to the trading partner and <strong>Inbound</strong> indicates messages sent from the trading partner to the host.</td>
</tr>
<tr>
<td><strong>Average Processing Time</strong></td>
<td>Shows the average document processing time, in milliseconds, for exchanged messages. <strong>Outbound</strong> indicates messages sent from the host to the trading partner and <strong>Inbound</strong> indicates messages sent from the trading partner to the host.</td>
</tr>
<tr>
<td><strong>Average Message Size (kb)</strong></td>
<td>Shows the average document size, in kilobytes, for outbound and inbound messages.</td>
</tr>
<tr>
<td><strong>Errors</strong></td>
<td>Shows the document error count.</td>
</tr>
</tbody>
</table>
17.3 B2B Partner Metrics

Figure 17–5 shows metrics summary data for a selected trading partner.

Table 17–2 describes the information on the Partners metrics tab. The data displayed is specific to the trading partner selected under Active Trading Partners.
### Table 17–2  B2B Partner Metrics

<table>
<thead>
<tr>
<th>Area/Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Messages and Errors**             | Processed messages = Completed messages + Errored messages  
Details of the errored messages are listed under **Errors**.                                                                                      |
| **Message Count**                   | Active messages are shown in this trend of inbound and outbound message quantity over time.                                               |
| **Summary**                         | The number of messages processed, the average processing time, the average message size, and the number of errors are summarized for the selected trading partner. |
| **No. of Messages Processed**       | Shows the number of messages sent by (From column) and received by (To column) the specified trading partner.                                  |
| **Average Processing Time (millsec)** | Shows the average document processing time, in milliseconds, for the specified trading partner.                                               |
| **Average Message Size (kb)**       | Shows the average document size, in kilobytes, for the specified trading partner.                                                           |
| **Errors**                          | Shows the document error count.                                                                                                               |
| **Active Document Types**           | Active document types are document types that have been included in active agreements. Details of the errors are listed under **Errors**. Messages processed include completed plus errored messages, that is, active messages. |
| **Name**                            | Name of the document definition                                                                                                               |
| **No. of Messages Processed**       | Shows the number of document messages exchanged between the host and trading partners. **Outbound** indicates messages sent from the host to the trading partner and **Inbound** indicates messages sent from the trading partner to the host. |
| **Average Processing Time (millsec)** | Shows the average document processing time, in milliseconds, for exchanged messages. **Outbound** indicates messages sent from the host to the trading partner and **Inbound** indicates messages sent from the trading partner to the host. |
| **Average Message Size (kb)**       | Shows the average document size, in kilobytes, for outbound and inbound messages.                                                            |
| **Errors**                          | Shows the document error count.                                                                                                               |
| **Errors**                          | Error message text is available from the Java resource bundle. The business message IDs link to business message details.                  |
Part V

Scripts and Utilities

This part describes how to do various tasks using scripts and utilities that are provided in Oracle B2B.

This part contains the following chapters:

- Chapter 18, "B2B Command-Line Tools"
- Chapter 19, "Utilities for Enqueuing and Dequeuing"
B2B Command-Line Tools

B2B command-line tools are available for a number of tasks.

---

**Note:** Command-line tools are for administrator use only. No security or permission checks are performed to prevent the logged-in user from purging, importing, or exporting data.

---

This chapter contains the following topics:

- Section 18.1, "Prerequisites for Running the Command-line Tools"
- Section 18.2, "Purging Data"
- Section 18.3, "Importing Data"
- Section 18.4, "Exporting Data"
- Section 18.5, "CPP/CPA Templates"
- Section 18.6, "CPP/CPA Import"
- Section 18.7, "CPP/CPA Export"
- Section 18.8, "Errors During Import"

### 18.1 Prerequisites for Running the Command-line Tools

Do the following before using the command-line tools:

1. Set `ORACLE_HOME` to your Oracle Fusion Middleware installation directory and then set the following environment variables:

   ```
   ANT_HOME = $ORACLE_HOME/.../modules/org.apache.ant_1.7.0
   JAVA_HOME = $ORACLE_HOME/.../jdk160_11
   ```

2. Create `jndi.properties`.

   ```
   cd $ORACLE_HOME/bin
   ant -f ant-b2b-util.xml b2bcreate-prop
   ```

3. Edit the `jndi.properties` file to include the `weblogic` password.
18.2 Purging Data

Note: Before purging data, exporting or archiving data is recommended. No security or permission checks are performed to prevent the logged-in user from purging data.

The following utility purges both design-time and run-time data and resets the environment to the installation time.

ant -f ant-b2b-util.xml b2bpurge

Table 18–1 lists the options for this command-line utility.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Domain</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Specifies purging design-time or run-time data.</td>
<td>DT, RT</td>
<td>No</td>
</tr>
<tr>
<td>msgState</td>
<td>Deletes messages with the specified message state. Used for run-time data.</td>
<td>MSG_COMPLETE, MSG_ERROR, MSG_WAIT_TRANSMIT, MSG_WAIT_FA, MSG_WAIT_BATCH</td>
<td>No. If msgstate is present, then start and end must be used.</td>
</tr>
<tr>
<td>start</td>
<td>Deletes messages that are created on or after the specified date. Used for run-time data.</td>
<td>dd-MMM-yyyy</td>
<td>No</td>
</tr>
<tr>
<td>end</td>
<td>Deletes messages that are created on or before the specified date. Used for run-time data.</td>
<td>dd-MMM-yyyy</td>
<td>No</td>
</tr>
<tr>
<td>purgecontrolnumber</td>
<td>Deletes control numbers. Used for run-time data.</td>
<td>true, false (default)</td>
<td>No</td>
</tr>
</tbody>
</table>

Example 18–1 Removes Design-Time Data
ant -f ant-b2b-util.xml b2bpurge -Dmode=DT

Example 18–2 Purges Run-Time Data
ant -f ant-b2b-util.xml b2bpurge -Dmode=RT

Example 18–3 Purges Run-Time Data, Including Control Numbers
ant -f ant-b2b-util.xml b2bpurge -Dmode=RT -Dpurgecontrolnumber=true

Example 18–4 Purges Messages with the Specified State Between the Specified Dates
ant -f ant-b2b-util.xml b2bpurge -Dmode=RT -Dstart=01-FEB-2009 -Dend=10-FEB-2009 -Dmsgstate=MSG_COMPLETE
18.3 Importing Data

**Note:** No security or permission checks are performed to prevent the logged-in user from importing data.

The following utility imports a configuration ZIP file to the repository. Basic validation is performed, but it is not a complete validation as with deployment validation. No data is overwritten unless you use the `overwrite` option.

```
ant -f ant-b2b-util.xml b2bimport -Dlocalfile=true -Dexportfile="/tmp/export.zip"
```

Table 18–2 lists the options for this command-line utility.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Domain</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>exportfile</td>
<td>Location of the export (ZIP) file</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>overwrite</td>
<td>Overwrites the existing business elements. For example, an existing delivery channel with the same trading partner name as a delivery channel in the import file is replaced if this option is set to true.</td>
<td>true (default)</td>
<td>No</td>
</tr>
<tr>
<td>localfile</td>
<td>If the export file location exists on the server, then set this option to true to improve performance. The export file must be on the server on which B2B is running.</td>
<td>true (default)</td>
<td>No</td>
</tr>
</tbody>
</table>

18.4 Exporting Data

**Note:** No security or permission checks are performed to prevent the logged-in user from exporting data.

The following utility exports the entire repository (without policy details) if no other options are specified.

```
ant -f ant-b2b-util.xml b2bexport
```

Table 18–3 lists the options for this command-line utility.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Domain</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>exportfile</td>
<td>Location of the ZIP file where the exported data is stored</td>
<td>/tmp/export.zip (default)</td>
<td>No</td>
</tr>
<tr>
<td>tpname</td>
<td>The trading partner name to be exported</td>
<td>Name of the trading partner</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 18–3  (Cont.) Options for ant -f ant-b2b-util.xml b2bexport

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Domain</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpanames</td>
<td>One or more agreement names to be exported. If one agreement is exported, then the ZIP file contains the folder /soa/b2b. If multiple agreements are exported, then the ZIP file contains an individual ZIP file for each of the agreements.</td>
<td>Agreement names must be separated by a comma</td>
<td>No</td>
</tr>
<tr>
<td>active</td>
<td>Exports agreements that have been deployed and are in active state.</td>
<td>true/false (default)</td>
<td>No</td>
</tr>
<tr>
<td>policies</td>
<td>Set to true to export the entire repository with user and role details, which is needed for the policy store. A warning is displayed to remind you to export the policy store also.</td>
<td>true/false (default)</td>
<td>No</td>
</tr>
</tbody>
</table>

See Section 10.2, "What Is Copied When You Import or Export from the Import/Export Tab," for more information.

Example 18–5  Exports the Trading Partner Acme to /tmp/Acme.zip

ant -f ant-b2b-util.xml b2bexport -Dtppname="Acme" -Dexportfile="/tmp/Acme.zip"

Example 18–6  Exports an Agreement from Design-Time with Listening Channel Details to /tmp/acmeGc.zip

ant -f ant-b2b-util.xml b2bexport -Dtppnames="Acme_GC_Agreement1" -Dexportfile="/tmp/acmeGc.zip"

Listening channels are deactivated while exporting and must be reactivated after you import data.

Example 18–7  Exports Multiple Deployed and Active Agreements to /tmp/export.zip

ant -f ant-b2b-util.xml b2bexport -Dtppnames="Acme_GC_Agreement1, GC_Acme_Agreement1" -Dactive=true

No listening channels are exported.

18.5 CPP/CPA Templates

The following utility creates a cpp_cpa.properties template file, which is used in the propfile option.

ant -f ant-b2b-util.xml b2bcreate-cpaprop

Table 18–4 lists the options for this command-line utility.
Example 18–8  Creates a Property File Template That Is Used in the propfile Option
ant -f ant-b2b-util.xml b2bcprop

18.5.1 Properties of cpp_cpa.properties

The following properties can be configured as part of the cpp_cpa.properties file:

- CPA Import Properties
- CPA Export Properties
- Common Properties

18.5.1.1 CPA Import Properties

The CPA import properties are as follows:

- oracle.tip.b2b.ebms.BPSSDocument (Optional Property)
  This property holds the absolute path for the BPSS document, which is used to get
  the BPSS document details to be imported into the Oracle B2B repository. If the
  property does not exist, then the values are imported from the CPA document.
  Multiple BPSS documents are separated by ; (semi-colon).

- oracle.tip.b2b.ebms.CPADocument (Required Property)
  This property is used to get the absolute path of the CPA document to be imported
  into the Oracle B2B repository.

- oracle.tip.b2b.ebms.xsdLocation (Optional Property)
  This property is used to specify the absolute path of the schema file location. This
  schema file is used for document validation. It is used only when a BPSS
  document is specified.

- oracle.tip.b2b.ebms.internalDeliveryChannel.protocol (Optional Property)
  The default internal delivery channel is an AQ queue. If you want to add a specific
  internal delivery channel (JMS/FTP/FILE/SFTP), then this property is used in
  Oracle B2B configuration. Specify all the required properties with respect to the
  specific transport protocol. Then use the specific channel to send messages to
  back-end applications.

18.5.1.2 CPA Export Properties

The CPA export properties are as follows:

- oracle.tip.b2b.ebms.OutputFolder (Required Property)
  This property is used to place the generated CPP/CPA files in the specified
  location.

- oracle.tip.b2b.ebms.Host (Required Property)
  This property is used to set the host trading partner.

- oracle.tip.b2b.ebms.HostEndPoint (Required Property)
  This property is used to set the host endpoint while generating the CPP/CPA
  export.
oracle.tip.b2b.ebms.HostCertificateAlias (Optional Property)
For secure message transfer, this property is used to get the host certificate details to the CPP/CPA export.

oracle.tip.b2b.ebms.TPCertificateAlias (Optional Property)
For secure message transfer, this property is used to get the trading partner certificate details to the CPP/CPA export.

oracle.tip.b2b.ebms.BPSSExport (Optional Property)
This Boolean property is used to generate the BPSS document.

### 18.5.1.3 Common Properties

The common properties are as follows:

oracle.tip.b2b.ebms.LogDirectory (Required Property)
This property is used to store the log files.

oracle.tip.b2b.ebms.LogLevel (Required Property)
This property is used to specify the mode of the logs, such as DEBUG, INFO, or ERROR.

oracle.tip.b2b.ebms.LogType (Required Property)
This property is used to specify whether the log file is stored as text or XML.

### 18.6 CPP/CPA Import

The following utility converts an ebXML standard `cpa.xml` file to an Oracle B2B metadata file, which must then be imported into Oracle B2B.

```
ant -f ant-b2b-util.xml b2bcpaimport
```

**Table 18–5** lists the options for this command-line utility.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Domain</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>propfile</td>
<td>Property file that stores configuration details for b2bcpaimport and b2bcpaexport</td>
<td>-</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Example 18–9** Converts CPA-Formatted XML to an Oracle B2B ZIP File

```
ant -f ant-b2b-util.xml b2bcpaimport -Dpropfile="/tmp/cpp_cpa.properties"
```

### 18.7 CPP/CPA Export

The following utility converts an Oracle B2B metadata file (data exported from Oracle B2B) to an ebXML standard `cpa.xml` file (a CPA-ready configuration).

```
ant -f ant-b2b-util.xml b2bcpaexport
```

**Table 18–6** lists the options for this command-line utility.
Table 18–6 Options for ant -f ant-b2b-util.xml b2bcpaimport

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Domain</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>propfile</td>
<td>Property file that stores configuration details for b2bcpaimport and b2bcpaexport</td>
<td>-</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Example 18–10 Converts an Oracle B2B ZIP File to a CPA-Formatted XML File

ant -f ant-b2b-util.xml b2bcpaexport -Dpropfile="/tmp/cpp_cpa.properties"

18.8 Errors During Import

If you get the following broken pipe error, use Oracle WebLogic Server Administration Console to increase Maximum Message Size to 20000000.

```
java
Exception in thread 'main' java.lang.Exception: java.rmi.UnmarshalException: Broken pipe;
nested exception is:
java.net.SocketException: Broken pipe

Caused by: java.rmi.UnmarshalException: Broken pipe; nested exception is:
java.net.SocketException: Broken pipe
```
Oracle B2B provides utilities to test and verify your installation and configuration before connecting to the host (back-end) applications. Use the utilities to learn how to send and receive business messages to and from Oracle B2B through the default AQ queue interface or the JMS queue interface. Other AQ internal delivery channels can be handled in the same way. See the B2B samples for examples of how to implement these utilities (See Section 1.7, "Oracle B2B Samples and Cookbooks," for information about the samples.)

This chapter contains the following topics:
- Section 19.1, "AQ Enqueue and Dequeue Utilities"
- Section 19.2, "JMS Enqueue and Dequeue Utilities"
- Section 19.3, "Using the attachmentsDescriptor.xsd"

19.1 AQ Enqueue and Dequeue Utilities

You can enqueue to and dequeue from an AQ queue using Java. IPEnqueue and IPDequeue must be executed in the Oracle B2B environment.

19.1.1 AQ Enqueue

Table 19–1 lists the Java AQ enqueue utility properties.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queue</td>
<td>The outbound AQ queue name. If unspecified, the Java enqueue utility uses the default outbound queue IP_OUT_QUEUE.</td>
</tr>
<tr>
<td>replyToMsgID</td>
<td>The message ID to which the sending message is replying, typically used for the response message type.</td>
</tr>
<tr>
<td>from</td>
<td>Trading partner that sends the message</td>
</tr>
<tr>
<td>to</td>
<td>Trading partner that receives the message</td>
</tr>
<tr>
<td>doctypeName</td>
<td>Document type name for the message</td>
</tr>
<tr>
<td>doctypeRevision</td>
<td>Document protocol revision for the message</td>
</tr>
<tr>
<td>payload</td>
<td>Payload file name</td>
</tr>
<tr>
<td>attachment</td>
<td>Attachment file name</td>
</tr>
<tr>
<td>url</td>
<td>The database URL format is jdbc:oracle:thin:@host:port:sid</td>
</tr>
</tbody>
</table>

Utilities for Enqueuing and Dequeuing
### Example: ipenqueue.properties

```properties
queue = 
url = jdbc:oracle:thin:@host:1521:sid
user = user1
password = password
replyToMsgID =
from = Acme
to = GlobalChips
doctypeName = 850
doctypeRevision = 4010
payload = Acme_850.xml
attachment =
```

#### Note:
In Windows ja_JP locale instances, the VARCHAR/String values are not enqueued correctly to the queue. The INT and CLOB values are enqueued correctly. This causes some fields, such as the from and to fields, to be null when the IPEnqueue utility is used to enqueue a file. As a workaround, in ja_JP locales, orai18n.jar should be added to the classpath while using oracle.tip.b2b.data.IPEnqueue.

### 19.1.2 AQ Dequeue

To dequeue messages, use the IPDequeue utility.

**Table 19–2** lists the Java AQ dequeue utility properties.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queue</td>
<td>The inbound AQ queue name. If unspecified, the Java dequeue utility uses the default inbound queue IP_IN_QUEUE.</td>
</tr>
<tr>
<td>count</td>
<td>The number of messages to dequeue. If unspecified, only one message is dequeued.</td>
</tr>
<tr>
<td>output</td>
<td>Output file name</td>
</tr>
<tr>
<td>url</td>
<td>The database URL format is jdbc:oracle:thin:@host:port:sid</td>
</tr>
<tr>
<td>user</td>
<td>The database user</td>
</tr>
</tbody>
</table>
19.2 JMS Enqueue and Dequeue Utilities

You can enqueue to and dequeue from a JMS destination (queue or topic) using utilities. If a user name and password are not provided, the local JNDI is used, including in a clustered environment, provided that the destinations are distributed. Oracle B2B does not support javax.jms.ObjectMessage.

19.2.1 JMS Enqueue

Use the JMS enqueue utility to send a message to a JMS destination (queue or topic). This utility expects a property file to be provided as a command-line argument where it reads the details to be sent.

Table 19–3 lists the properties that can be configured in the file.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination</td>
<td>JNDI name of queue or topic to send message to</td>
</tr>
<tr>
<td>cf</td>
<td>JNDI name of connection factory to use</td>
</tr>
<tr>
<td>factory</td>
<td>Factory provider class</td>
</tr>
<tr>
<td>isTopic</td>
<td>Indicator for topic (optional)</td>
</tr>
<tr>
<td>url</td>
<td>The JNDI URL format is url=t3://host_name:port_number/</td>
</tr>
<tr>
<td>user</td>
<td>The application server administrator userID.</td>
</tr>
<tr>
<td>password</td>
<td>The application server administrator password</td>
</tr>
<tr>
<td>from</td>
<td>From party</td>
</tr>
<tr>
<td>to</td>
<td>To party</td>
</tr>
<tr>
<td>eventName</td>
<td>Action name</td>
</tr>
<tr>
<td>doctypeName</td>
<td>Document type name</td>
</tr>
<tr>
<td>doctypeRevision</td>
<td>Document type revision</td>
</tr>
<tr>
<td>payload</td>
<td>Payload file path</td>
</tr>
<tr>
<td>attachment</td>
<td>Attachment file path</td>
</tr>
<tr>
<td>msgID</td>
<td>Message ID (optional). B2B generates its own message ID if it is not provided as part of an enqueue.</td>
</tr>
</tbody>
</table>
Example 19–1 shows the sample jms_enqueue.properties file.

**Example 19–1 Sample jms_enqueue.properties File**

```properties
####### Destination Details #######
destination = jms/b2b/B2B_IN_QUEUE
cf = jms/b2b/B2BQueueConnectionFactory

####### Server and Factory Details #######
factory=weblogic.jndi.WLInitialContextFactory
url=t3://host_name:port_number/
#user=<uncomment and provide you username>
#password=<uncomment and provide you password if required>

####### Payload Details #######
from=Acme
to=GlobalChips
(eventName=SampleEvent
doctypeName=Custom
doctypeRevision=1.0
payload=/scratch/work/GlobalChips_1234.dat
```

See the sample documentation for how to run these utilities.

### 19.2.2 Enqueue—Using a JMS JCA Adapter or Custom Utilities

The properties used by the AQ and JMS utilities are translated internally before the message is sent to the destination. Ensure that the properties in Table 19–4 are set as part of the javax.jms.Message delivered to the destination that B2B listens on.

**Table 19–4 How AQ/JMS Properties Are Translated for Custom Utilities**

<table>
<thead>
<tr>
<th>AQ/JMS Utilities</th>
<th>Translated Value—For Custom Utilities</th>
<th>JMS Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>FROM_PARTY</td>
<td>Sent as a string type message property</td>
</tr>
<tr>
<td>to</td>
<td>TO_PARTY</td>
<td>Sent as a string type message property</td>
</tr>
<tr>
<td>doctypeName</td>
<td>DOCTYPE_NAME</td>
<td>Sent as a string type message property</td>
</tr>
<tr>
<td>doctypeRevision</td>
<td>DOCTYPE_REVISION</td>
<td>Sent as a string type message property</td>
</tr>
<tr>
<td>eventName</td>
<td>ACTION_NAME</td>
<td>Sent as a string type message property</td>
</tr>
<tr>
<td>msgID</td>
<td>MSG_ID</td>
<td>Sent as a string type message property</td>
</tr>
<tr>
<td>replyToMsgID</td>
<td>INREPLYTO_MSG_ID</td>
<td>Sent as a string type message property</td>
</tr>
</tbody>
</table>
19.2.3 JMS Dequeue

This utility receives messages from the destination. The `count` property can be specified to control the number of messages to be picked up from the destination. Retrieved messages are written to the file `JMSDequeue.txt` at the current path (where you run the utility).

See the samples documentation for how to run these utilities.

Example 19–2 shows the sample JMS dequeue properties file.

**Example 19–2 Sample jms_dequeue.properties File**

```properties
#### Destination Details####
destination =.jms/b2b/B2B_IN_QUEUE
cf =.jms/b2b/B2BQueueConnectionFactory
count=1

#### Server and Factory Details ####
factory=weblogic.jndi.WLInitialContextFactory
url=t3://host_name:port_number/
#user=<uncomment and provide your username>
#password=<uncomment and provide your password if required>
```

19.3 Using the attachmentsDescriptor.xsd

Use the `attachmentsDescriptor.xsd` file for sending attachments.

Example 19–3 shows a sample attachment XML file.

**Example 19–3 Sample Attachment XML File**

```xml
<?xml version='1.0' encoding='ISO-8859-1'?>
<!--Sample XML file generated by XMLSpy v2005 sp1 U (http://www.xmlspy.com)-->
<Attachments xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="AttachmentsDescriptor.xsd" version="1.0" boundary="boundary---">
<AttachmentPart>
  <Location>file:///home/user_dir/data.xml</Location>
  <Content-Type>
    <Top-Level-Type>text</Top-Level-Type>
    <Sub-Type>plain</Sub-Type>
    <Parameter Value="charset" Name="us-ascii"/>
  </Content-Type>
  <!--Content-Transfer-Encoding>BASE64</Content-Transfer-Encoding-->
  <Content-ID/>
  <Content-Description/>
</AttachmentPart>
</Attachments>
```
This part contains the following appendixes:

- Appendix A, "Performance Tuning and Large Payloads"
- Appendix B, "Setting B2B Configuration Properties in Fusion Middleware Control"
- Appendix C, "Back-End Applications Interface"
- Appendix D, "Exception Handling"
This appendix contains the following topics:

- Section A.1, "Settings for Performance Tuning"
  - Section A.1.1, "Memory Arguments"
  - Section A.1.2, "Heap Size Settings"
  - Section A.1.3, "MDS Cache Size"
  - Section A.1.4, "Number of Threads"
  - Section A.1.5, "Stuck Thread Max Time"
  - Section A.1.6, "Tablespace Size"
  - Section A.1.7, "JTA Settings"
  - Section A.1.8, "Configuring High-Volume Systems"
  - Section A.1.9, "Settings for Batching a Large Number of Documents"
  - Section A.1.10, "Streaming Inbound and Outbound Attachments"

- Section A.2, "Handling Large Payloads"
  - Section A.2.1, "Introduction to Large Payload Support"
  - Section A.2.2, "Large Payloads and 32-Bit Windows PCs"
  - Section A.2.3, "Settings for a Large Dataset Scenario"

### A.1 Settings for Performance Tuning

To improve performance, set memory arguments appropriately based on your requirements and system. Code clean-up, multithreading, and table indexing are major contributors to maximizing the use of available resources. Java performance tuning also helps in sharing the resources among the various processes based on the usage and need.

When using the large payload settings, the internal delivery channel must be the default channel or a JMS queue.

Changes to B2B configuration properties typically require a server restart. See the following for more information:

- Appendix B, "Setting B2B Configuration Properties in Fusion Middleware Control"
- *Oracle Fusion Middleware Administrator’s Guide for Oracle SOA Suite*
The syntax in various examples in this section reflect generic UNIX format.

The following settings improved Oracle B2B performance based on 2 GB of RAM on a 32-bit computer and 200 MB of B2B configuration data. When working in a Windows operating system with large payloads, a 64-bit server is recommended.

### A.1.1 Memory Arguments

Memory arguments are captured in `DOMAIN_HOME/bin/setSOADomainEnv.sh`. Memory tuning applies to Oracle JRocket or SUN JVM, as shown in Example A–1 and Example A–2.

**Example A–1  Settings for Oracle JRocket**

```bash
export JAVA_VENDOR Oracle
DEFAULT_MEM_ARGS="-Xms1024m -Xmx1024m"
```

**Example A–2  Settings for Sun JVM**

```bash
export JAVA_VENDOR Sun
DEFAULT_MEM_ARGS="-Xms1024m -Xmx1024m"
if ["$JAVA_VENDOR"!= "Oracle" ];then
  DEFAULT_MEM_ARGS="$DEFAULT_MEM_ARGS -XX:CompileThreshold=100000 -XX:PermSize=256m -XX:MaxPermSize=256m"
fi
```

-Xms and -Xmx can be increased up to 2 GB based on memory availability.

### A.1.2 Heap Size Settings

Verify the heap size settings in the `setSOADomain.sh` script (see DEFAULT_MEM_ARGS) before starting any of the following servers in the WebLogic domain:

- The SOA managed server
- The WebLogic Admin Server

Using precise heap settings when starting the servers is necessary for B2B to process large payloads.

### A.1.3 MDS Cache Size

To set the Metadata Service (MDS) instance cache size, use Oracle Enterprise Manager Fusion Middleware Control to set `b2b.mdsCache` to a value such as 200000. See Appendix B.1, "Properties To Set in Fusion Middleware Control," for more information.

### A.1.4 Number of Threads

Changing the value of `b2b.inboundThreadCount` and `b2b.outboundThreadCount` can improve Oracle B2B message processing. The recommended value depends on your system. For a 2 GB computer, a setting of 3 to 5 is recommended. The `b2b.inboundThreadSleepTime` and `b2b.outboundThreadSleepTime` properties put a thread to sleep after message processing. A setting between 10 and 1000 (milliseconds) is recommended.

### A.1.5 Stuck Thread Max Time

Changing the value of **Stuck Thread Max Time** can improve Oracle B2B message processing if a thread is stuck. This is the maximum amount of time that the server
checks the number of seconds that a thread must be continually working before the server considers the thread stuck.

Only if you see a stuck thread exception should you change the Stuck Thread Max Time setting in Oracle WebLogic Server Administration Console. Increasing this number can degrade performance.

Navigate to Environment > Servers > soa_server_name > Configuration > Tuning. Set Stuck Thread Max Time, shown in Figure A–1, to a maximum of 1200. (The default value is 600 seconds.)

Figure A–1 Changing Stuck Thread Max Time
### A.1.6 Tablespace Size

If you store more than an a 150 MG configuration, extend or add a data file to increase tablespace size, as shown in Example A–3.

**Example A–3 Increasing Tablespace Size**

```
ALTER TABLESPACE sh_mds add DATAFILE 'sh_mds01.DBF' SIZE 100M autoextend on next 10M maxsize unlimited;
ALTER TABLESPACE sh_ias_temp add TEMPFILE 'sh_ias_temp01.DBF' SIZE 100M autoextend on next 10M maxsize unlimited;
```

### A.1.7 JTA Settings

On slower Windows computers (2 to 4 GB, 32-bit), the JTA timeout must be increased for Oracle B2B. Use the Oracle WebLogic Server Administration Console to increase the JTA transaction timeout to a higher number, depending on your environment. In some situations, the suggested setting is an increase to 90 seconds, and if required, to higher values.

### A.1.8 Configuring High-Volume Systems

To improve performance on systems handling a large number of messages per second, the following steps are recommended:

1. Create a separate table space for the `B2B_DATA_STORAGE` table so that the LOB data can be stored separately.
2. Increase the block size for the tablespace for storing LOB data to reduce insert contention.

### A.1.9 Settings for Batching a Large Number of Documents

The following are recommended to batch more 1000 documents, where the payload size is greater than 30 KB.

- Install the SOA server on a 64-bit computer.
- Set the batch commit size to a value greater than 0, such as 100.
- Use the Oracle WebLogic Server Administration Console to increase the JTA transaction timeout to a higher value.
- In `setSOADomainEnv.sh`, change the heap size setting from `-Xms1024m` to `-Xmx2048m`.

### A.1.10 Streaming Inbound and Outbound Attachments

To stream attachments from the SOA WS binding layer, add the following properties in `composite.xml` for services and references:

```
streamIncomingAttachments="true" streamOutgoingAttachments="true"
```

**Example A–4** shows a sample.

**Example A–4  Properties for Streaming Inbound and Outbound Attachments**

```
<binding.ws
port="http://services.otn.com#wsdl.endpoint(MIMEService/MIMEService)"
xmlns:ns="http://xmlns.oracle.com/sca/1.0"
streamIncomingAttachments="true" streamOutgoingAttachments="true"/>
```
A.2 Handling Large Payloads

Oracle B2B can handle large payloads through the SOA Infrastructure and JMS internal queues.

A.2.1 Introduction to Large Payload Support

**Inbound Setup**

Figure A–2 shows the properties to set for inbound cases. Go to Administration > Configuration.

*Figure A–2  Large Payload Size*

If a composite is deployed to handle the large payload, this is the only configuration needed. If B2B is not delivering the payload to a composite, set Use JMS Queue as default to true, as shown in Figure A–3. Go to Administration > Configuration.

*Figure A–3  Use JMS Queue*

With Use JMS Queue as default set to true, the payload is delivered to B2B_IN_QUEUE, a JMS-based queue.

**Outbound Setup**

Figure A–4 shows the properties to set for the outbound case.

*Figure A–4  Large Payload Directory*

**Notes**

1. If you are doing large payload testing, set Log Payload on the Administration > Configuration tab to false.
2. If you are doing large payload testing, set **Show Payload** on the **Administration > Configuration** tab to false to avoid listing the payload in reports.

3. If an enqueue script is used when working with large payloads, add
   ```
   eventName=LARGE_PAYLOAD=true
   ```

4. Increase the maximum heap size to use `-Xmx2048m`.

5. Increase the database tablespace size for soadatasource to have autoextend on and increase the tablespace file size maximum limit.
   ```
   alter database datafile '/scratch/$user/auto_work/db230/.oradata/db230/SH_soainfra.dbf' autoextend on next 10M maxsize 4096M
   ```

6. Set the transaction timeout in Oracle WebLogic Administration Server:
   - Weblogic Console Services -> JTA Timeout Seconds=720 seconds
   - Weblogic Console Services -> JDBC->DataSources->SOADatasource - increase XA timeout to 120-180 seconds

7. If Oracle B2B is used alone (without the SOA Infrastructure), the JTA timeout can be set in `b2b.jtaTimeout` by using Oracle Enterprise Manager Fusion Middleware Control. See *Oracle Fusion Middleware Administrator’s Guide for Oracle SOA Suite* for more information.

8. For an outbound SOA composite, always select the **Use file streaming** option for the File Adapter, as shown in **Figure A–5**.

---

**Figure A–5  The File Adapter Use File Streaming Option**
A.2.2 Large Payloads and 32-Bit Windows PCs

On a 32 bit Windows computer, the payload size limit is 50 MB. This is because the heap size cannot be set to more than 1536m due to Windows-specific limits. Java VM throws an out-of-memory exception.

A.2.3 Settings for a Large Dataset Scenario

The following suggested settings are based on a dataset with approximately 2,500 trading partners, an export ZIP file that is approximately 253 MB in size, and assumes a 6 GB computer. Using these settings can considerably reduce data upload time when using the Upgrade Assistant.

1. Use Oracle WebLogic Server Administration Console to increase the
   - JTA transaction timeout from 30 to 350
   - Maximum message size from the default size to 20000000

2. Add indices for better performance. Using Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - Production, with the Partitioning, OLAP, Data Mining and Real Application Testing options, do the following:

   SQL> create index idx_mds_attr on
   rcl_mds.MDS_ATTRIBUTES("ATT_VALUE","ATT_LOCALNAME");
   Index created.

   SQL> create index idx_mds_path on
   rcl_mds.MDS_PATHS("PATH_CONTENTID","PATH_PARTITION_ID");
   Index created.

   SQL> commit;

3. Start the managed server with the following updated memory setting:

   DEFAULT_MEM_ARGS="-Xms1024m -Xmx2048m"

4. Change ORACLE_HOME/bin/UA default memory from the default 256 to 2048. The default is

   $JAVA_HOME/bin/java ${JAVAMODE} -Xmx256m -classpath ${CLASSPATH}
   -Dua.home=$base_dir -Dice.pilots.html4.ignoreNonGenericFonts=true
   -Dsun.lang.ClassLoader.allowArraySyntax=true
   -Doracle.installer.oui_loc=$OUI_HOME oracle.ias.upgrade.UpgradeDriver
   $ARGUMENTS

   Change the default to

   $JAVA_HOME/bin/java ${JAVAMODE} -Xmx2048m -classpath ${CLASSPATH}
   -Dua.home=$base_dir -Dice.pilots.html4.ignoreNonGenericFonts=true
   -Dsun.lang.ClassLoader.allowArraySyntax=true
   -Doracle.installer.oui_loc=$OUI_HOME oracle.ias.upgrade.UpgradeDriver
   $ARGUMENTS

5. Change the value of Stuck Thread Max Time from 600 to 2000.
Setting B2B Configuration Properties in Fusion Middleware Control

Use Oracle Enterprise Manager Fusion Middleware Control to set B2B configuration properties for properties that are not set on the Configuration tab of the Oracle B2B interface (see Chapter 15, "Configuring B2B System Parameters"). B2B properties can also be set by using the configmbeanutil utility.

This appendix contains the following topics:

- Section B.1, "Properties To Set in Fusion Middleware Control"
- Section B.2, "Using the configmbeanutil Utility"

**B.1 Properties To Set in Fusion Middleware Control**

The following properties can be set in Oracle Enterprise Manager Fusion Middleware Control. See Oracle Fusion Middleware Administrator’s Guide for Oracle SOA Suite for how to set the properties.

**Note:** Restarting the SOA Server is required for changes to B2B properties.

- To turn off validation during deployment, set this property to false:
  
  `b2b.deploy.validation`

  This is useful when deploying a large number of agreements where you are certain that the data is valid.

- To set the Metadata Service (MDS) instance cache size, set this property:
  
  `b2b.mdsCache cache_size`

  A ratio of 5:1 is recommended for the xmx-to-mdsCache values. For example, if the xmx size is 1024, maintain mdsCache at 200 MB.

- To set the number of threads and thread sleep time to improve message processing, set the following:
  
  `b2b.inboundThreadCount`
  `b2b.inboundThreadSleepTime`
  `b2b.outboundThreadCount`
  `b2b.outboundThreadSleepTime`
  `b2b.defaultThreadCount`
  `b2b.defaultThreadSleepTime`
The recommended values for `b2b.inboundThreadCount` and `b2b.outboundThreadCount` depend on your system. For a 2 GB computer, a setting of 3 to 5 is recommended.

The `b2b.inboundThreadSleepTime` and `b2b.outboundThreadSleepTime` properties put a thread to sleep after message processing. A setting between 10 and 1000 (milliseconds) is recommended.

- To set up File, FTP, or Email transports in an HA environment, specify a unique name for each instance by using
  
  `b2b.HAInstanceName unique_instance_name`

  If you use `#ServerName#` for the value, B2B retrieves the WebLogic Server name as the `HAInstanceName`.

- To swap the interchange and group IDs in the generated FA from the sender to the receiver and vice versa, set this property to true:
  
  `b2b.FAInternalProperties`

  The test usage indicator parameter is also mapped; that is, an inbound message set as a test message generates an FA with the flag set to test message. Setting `b2b.FAInternalProperties` to true, by default, does not map ISA segment elements (1-4).

- To specify how a functional acknowledgment is handled, set this property to true or false:
  
  `b2b.FAHandledByB2B`

  When the `b2b.FAHandledByB2B` property is set to false, then for an inbound EDI message, B2B does not generate a functional acknowledgment (FA).

  If the FA is marked as expected in an agreement, then the message is placed into the MSG_WAIT_FA state and the back-end application is expected to generate the FA and push it to B2B as an outbound message back to the partner.

  The following limitations apply when generating the FA from the back-end application:
- The FA is correlated with the original message based on the ReferToMsgID value set in the enqueue properties. The FA is correlated based on control numbers also.

- If the FA indicates that there was an error in the received message, the status of the correlated message is not updated to indicate an error. The correlated message is updated to MSG_COMPLETE.

These limitations are not present when the FA is generated by B2B (that is, when b2b.FAHandledByB2B is true).

- To specify how errors are flagged, set this parameter to true or false:

  b2b.outboundOneErrorAllError

For outbound messages, b2b.outboundOneErrorAllError can be set to true or false. The default is false. If the parameter is set to true, then for outbound messages, even with a single message in error, all outbound batch messages are set to the error state. None of the batched messages are sent to the trading partner. When using the b2b.outboundOneErrorAllError parameter, inbound messages behave as if b2b.outboundOneErrorAllError is set to false; that is, if an error occurs during an inbound message process, then only that message is flagged with the error and other messages are passed. There is no option to flag every message as failed.

- To turn on payload obfuscation, set this property to true:

  b2b.payloadObfuscation

See Section 1.4.1, "Payload Obfuscation," for more information.

- To ignore interchange validation errors for EDI messages, use this property:

  b2b.edi.ignoreValidation

If you add or update this flag, a server restart is not needed. However, if you remove the flag, the server must be restarted for the change to take effect.

- To check for duplicate messages, use this property:

  b2b.checkDuplicate

If this property is set to false, a check for a duplicate of the incoming business message is not performed. By default, Oracle B2B checks for duplicate messages based on the business message ID of the incoming message.

**B.2 Using the configmbeanutil Utility**

You can also use the configmbeanutil utility to set properties.

**To use the configmbeanutil utility:**

```
Note: To access the Fusion Middleware directory, you must provide an mbean property file (mbeanutil.properties) that contains host, port, user, and password information. For example,

host=myfmw.com
port=7001
user=weblogic
password=mypwd
```
1. Set the `MW_HOME` environment variable to point to the Fusion Middleware installation directory. For example,

   ```
   setenv MW_HOME /scratch/$user/fmwhome
   ```

2. Set the `JAVA_HOME` environment variable. For example,

   ```
   setenv JAVA_HOME `${MW_HOME}/jdk160_14_R27.6.4-18`
   ```

3. Add the Java bin directory to the `PATH` environment variable. For example,

   ```
   setenv PATH `${JAVA_HOME}/bin:${PATH}`
   ```

Example B–1, Example B–2, Example B–3, and Example B–4 show uses for this utility.

**Example B–1  To Print All Properties to the Console**

```java
java -cp $MW_HOME/AS11gR1SOA/soa/modules/oracle.soa.b2b_11.1.1/b2b.jar:$MW_HOME/wlserver_10.3/server/lib/wljmxclient.jar:$MW_HOME/modules/glassfish.jaxb_1.2.0.0_2-1-7.jar oracle.tip.b2b.utility.ConfigMBeanUtility
```

**Example B–2  To Add a Property**

```java
java -cp $MW_HOME/AS11gR1SOA/soa/modules/oracle.soa.b2b_11.1.1/b2b.jar:$MW_HOME/wlserver_10.3/server/lib/wljmxclient.jar:$MW_HOME/modules/glassfish.jaxb_1.2.0.0_2-1-7.jar oracle.tip.b2b.utility.ConfigMBeanUtility add b2b.test cool ok
```

**Example B–3  To Update a Property**

```java
java -cp $MW_HOME/AS11gR1SOA/soa/modules/oracle.soa.b2b_11.1.1/b2b.jar:$MW_HOME/wlserver_10.3/server/lib/wljmxclient.jar:$MW_HOME/modules/glassfish.jaxb_1.2.0.0_2-1-7.jar oracle.tip.b2b.utility.ConfigMBeanUtility update b2b.test thru
```

**Example B–4  To Remove a Property**

```java
java -cp $MW_HOME/AS11gR1SOA/soa/modules/oracle.soa.b2b_11.1.1/b2b.jar:$MW_HOME/wlserver_10.3/server/lib/wljmxclient.jar:$MW_HOME/modules/glassfish.jaxb_1.2.0.0_2-1-7.jar oracle.tip.b2b.utility.ConfigMBeanUtility remove b2b.test
```
This appendix contains the following topics:

- Section C.1, "Mapping B2B IP_MESSAGE_TYPE to SCA Normalized Message Properties"
- Section C.2, "Normalized Message Properties"

### C.1 Mapping B2B IP_MESSAGE_TYPE to SCA Normalized Message Properties

Table C–1 maps the B2B IP_MESSAGE_TYPE to SCA normalized message properties.

<table>
<thead>
<tr>
<th>AQ (IP_MESSAGE_TYPE)</th>
<th>SCA</th>
<th>JMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG_ID</td>
<td>b2b.messageId</td>
<td>MSG_ID</td>
</tr>
<tr>
<td>INREPLYTO_MSG_ID</td>
<td>b2b.replyToMessageId</td>
<td>INREPLYTO_MSG_ID</td>
</tr>
<tr>
<td>FROM_PARTY</td>
<td>b2b.fromTradingPartnerId</td>
<td>FROM_PARTY</td>
</tr>
<tr>
<td>-</td>
<td>b2b.fromTradingPartnerIdType</td>
<td>-</td>
</tr>
<tr>
<td>TO_PARTY</td>
<td>b2b.toTradingPartnerId</td>
<td>TO_PARTY</td>
</tr>
<tr>
<td>-</td>
<td>b2b.toTradingPartnerIdType</td>
<td>-</td>
</tr>
<tr>
<td>ACTION_NAME</td>
<td>-</td>
<td>ACTION_NAME</td>
</tr>
<tr>
<td>DOCTYPE_NAME</td>
<td>b2b.documentTypeName</td>
<td>DOCTYPE_NAME</td>
</tr>
<tr>
<td>DOCTYPE_REVISION</td>
<td>b2b.documentProtocolVersion</td>
<td>DOCTYPE_REVISION</td>
</tr>
<tr>
<td>-</td>
<td>b2b.documentProtocolName</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>b2b.documentDefinitionName</td>
<td>-</td>
</tr>
<tr>
<td>MSG_TYPE</td>
<td>b2b.messageType</td>
<td>MSG_TYPE</td>
</tr>
<tr>
<td>-</td>
<td>b2b.conversationId</td>
<td>-</td>
</tr>
<tr>
<td>PAYLOAD</td>
<td>body</td>
<td>-</td>
</tr>
<tr>
<td>ATTACHMENT</td>
<td>attachments</td>
<td>-</td>
</tr>
</tbody>
</table>

### C.2 Normalized Message Properties

Header manipulation and propagation are key business integration messaging requirements. Like other SOA components such as Oracle BPEL Process Manager, Oracle Mediator, and Oracle JCA, Oracle B2B relies on header support to solve
Normalized Message Properties

integration needs. For example, you can preserve a file name from the source directory to the target directory by propagating it through message headers.

Normalized messages have two parts, properties and payload. Typically, properties are name-value pairs of scalar types. To fit the existing complex headers into properties, properties are flattened into scalar types.

Manipulating headers in design time is simplified by using predetermined complex properties. In B2B, you can manipulate headers with reserved key words. However, some properties are dynamically generated based on your input. These definitions are not predetermined and hence cannot be accounted for in the list of predetermined property definitions. You cannot design header manipulation of the dynamic properties before they are defined. To address this limitation, you must generate all the necessary services (composite entry points) and references. This restriction applies to services that are expected to generate dynamic properties. After dynamic properties are generated, they are stored for each composite, and can be manipulated in the composite editor.

Figure C–1 shows the Properties tab of an Invoke activity, part of a BPEL process that includes a B2B binding component. Enter values and specify the input or output type for B2B properties on this dialog.

Figure C–1 Invoke Activity Showing B2B Normalized Message Properties

Table C–2 lists the predetermined properties of a normalized message for Oracle B2B.
### Table C–2 Properties for Oracle B2B

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Propagable (Yes/No)</th>
<th>Direction (Inbound/Outbound)</th>
<th>Data Type</th>
<th>Range of Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b2b.conversationId</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The ID used to relate the message to the message response</td>
</tr>
<tr>
<td>b2b.documentDefinitionName</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The document definition, for example, 850_def for an EDI X2 document</td>
</tr>
<tr>
<td>b2b.documentProtocolName</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The document protocol, for example, X12 for an EDI X12 document</td>
</tr>
<tr>
<td>b2b.documentProtocolVersion</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The document version, for example, 4010 for an EDI X12 document</td>
</tr>
<tr>
<td>b2b.documentTypeName</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The document type, for example, 850 for an EDI X12 document</td>
</tr>
<tr>
<td>b2b.fromTradingPartnerId</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The trading partner identifier of the sender, for example, the name, such as Acme, or a DUNS number</td>
</tr>
<tr>
<td>b2b.fromTradingPartnerIdType</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The trading partner identifier type for the sender, for example, Name or DUNS</td>
</tr>
<tr>
<td>b2b.messageId</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>A unique message ID, not directly related to ECID. (ECID information is stored in the B2B AppMessage table.)</td>
</tr>
<tr>
<td>b2b.messageType</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>Message type values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ Request = 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ Response = 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>■ Functional Ack = 9</td>
</tr>
<tr>
<td>b2b.replyToMessageId</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The message ID to which the sending message is replying</td>
</tr>
<tr>
<td>b2b.toTradingPartnerId</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The trading partner identifier of the receiver, for example, the name, such as Acme, or a DUNS number.</td>
</tr>
<tr>
<td>b2b.toTradingPartnerIdType</td>
<td>No</td>
<td>Both</td>
<td>String</td>
<td>-</td>
<td>The trading partner identifier type for the receiver, for example, Name or DUNS. If no value is found, the Name type is assumed.</td>
</tr>
</tbody>
</table>
Oracle B2B handles exceptions for inbound and outbound messages. This appendix describes the exception handling, error messages, and structures for Oracle B2B.

This appendix contains the following topics:

- Section D.1, "Inbound Messages"
- Section D.2, "Outbound Messages"
- Section D.3, "Using a JMS Queue for Error Message Delivery"
- Section D.4, "Using a Custom Exception Queue for Error Message Delivery"
- Section D.5, "Inbound Exception Handling Scenarios"
- Section D.6, "Exception Payload Definition"

D.1 Inbound Messages

This section describes the following inbound message types:

- Request or Response Messages
- Acknowledgment Messages
- Exception Messages

D.1.1 Request or Response Messages

For an incoming request, response, or functional acknowledgment message that results in an exception, the following actions occur when you use the default error handling settings:

- An exception message is sent to the application.
  
  The exception message is enqueued to IP_IN_QUEUE and has the recipient name b2berroruser. The enqueued exception is based on ipException.xsd and contains information such as the error message (errorText has a short description and errorDescription has a longer description) and the error code.

- An exception message is sent to the trading partner, if mandated by the exchange specification.

  The exception message is sent back to the trading partner only if there is enough information to identify the outgoing trading partner agreement. For this purpose, the flag B2BHeader.sendException is used. The flag is set to true when enough information is extracted from the incoming message to send the exception message to the trading partner.
Oracle B2B catches exceptions thrown by exchange or document layers. If the B2Bheader.sendException flag is set to true, the outgoing trading partner agreement is processed and an exception message is sent to the trading partner.

D.1.1.1 Inbound ebMS, AS1, and AS2 Messages
If the following types of failure occur while an incoming message is processing, then the receiving trading partner sends a negative acknowledgment to the sender.

- Decryption fails
- Verification fails
- Agreement is not found
- Document identification fails
- Document validation fails (and so on)

The negative acknowledgment message has the reference for the original (request) message details to correlate at the sender side.

D.1.2 Acknowledgment Messages
For an incoming acknowledgment message that results in an exception, the following actions occur when you use the default error handling settings:

- An exception message is sent to the application.
  
  The exception message is enqueued to IP_IN_QUEUE and has the recipient name b2berroruser. The enqueued exception is based on ipException.xsd and contains information such as error text and error code.

- No exception message is sent back to the trading partner.

D.1.3 Exception Messages
For an incoming exception message, the following actions occur when you use the default error handling settings:

- The original message is updated so that it is in an errored state. The incoming exception is processed and delivered to the application normally.

- If the incoming exception message itself results in an exception, an exception message is sent to the application.
  
  The exception message is enqueued to IP_IN_QUEUE and has the recipient name b2berroruser. The enqueued exception is based on ipException.xsd and contains information such as error text and error code. No exception message is sent back to the trading partner in this case.

Exceptions can be delivered to default queues (B2B_IN_QUEUE or IP_IN_QUEUE) or custom JMS queues configured for exception messages. See Section D.4, "Using a Custom Exception Queue for Error Message Delivery," for more information.

D.2 Outbound Messages
If an exception occurs while an outbound message is being sent (for example, if the trading partner identification fails), then an exception message is sent to the application. When you use the default error handling settings, the exception message
is enqueued to IP_IN_QUEUE and has the recipient name b2berroruser. The enqueued exception is based on ipException.xsd and contains information such as error text and error code.

If an exception occurs during Oracle B2B startup, then an exception message is enqueued to IP_IN_QUEUE and has the recipient name b2berroruser. The enqueued exception is based on ipException.xsd and contains information such as error text and error code. The correlation ID is not populated in this case.

Note the following:

- When the exception message is sent back to the application, the document type is Exception instead of the original message document type.
- When the exception message is sent back to the application, inReplyToMessageId is populated with the correlation ID value.
- For inbound exception handling, a business message is always created and populated with the available information. It also points to the corresponding wire message. The wire message is updated so that it is in an errored state. For the outbound direction, only the business message is updated, because the wire message does not exist. However, if a transmission failure occurs, then the wire message table does have an entry.
- The error reports are updated to show only business messages; a business message is always created in the inbound and outbound directions.

D.3 Using a JMS Queue for Error Message Delivery

The default settings, as described in Section D.1 and Section D.2, use an AQ queue, IP_IN_QUEUE, as the exception queue. You can configure B2B to use a JMS queue by setting the Use JMS Queue as default parameter to true on the Configuration tab. The JMS queue, B2B_IN_QUEUE, becomes the default exception queue unless you have configured a custom JMS exception queue and selected it as the value for the Exception Queue parameter (see Section D.4, "Using a Custom Exception Queue for Error Message Delivery.") In general, B2B sends inbound messages to B2B_IN_QUEUE and polls on B2B_OUT_QUEUE for outbound messages.

Because JMS queues cannot use b2berroruser as the consumer, a JMS message property is used to filter exception messages for error handling. Specifically, when the MSG_TYPE value equals 3 (MSG_TYPE=’3’), all exception messages are received by the JMS receiver. (For successful messages, MSG_TYPE=’1’. ) All JMS message properties are of type string.

See Table 15–1, "Configuration Settings" for more information on the Use JMS Queue as default parameter.

D.4 Using a Custom Exception Queue for Error Message Delivery

You can create custom JMS exception queues by configuring JMS internal delivery channels (JMS queues or topics) for the host trading partner on the Partners > Channels tab, as shown in Figure D–1.
Then select the queue from the Exception Queue parameter on the Configuration tab. The Exception Queue dropdown lists all JMS internal delivery channels from the host trading partner.

A null default value for this parameter means that the JMS queue, B2B_IN_QUEUE, is the exception queue if Use JMS Queue as default is set to true, and that the AQ queue, IP_IN_QUEUE, is the exception queue if Use JMS Queue as default is set to false.

If B2B fails to deliver an exception message to the selected custom exception queue, then the exception message is sent to the default internal delivery channel.

See Table 15–1, "Configuration Settings" for more information on the Exception Queue parameter.

### D.5 Inbound Exception Handling Scenarios

Table D–1 describes inbound exception handling scenarios.
### Inbound Exception Handling Scenarios

<table>
<thead>
<tr>
<th>If an exception occurs because...</th>
<th>Then Oracle B2B does...</th>
</tr>
</thead>
</table>
| The identification of the exchange fails or the exchange is not supported | - Notifies the middleware  
- Updates the wire message as in an errored state  
- Creates a business message in an errored state for the wire message  
- Sends a transport error message to the trading partner if the sendException flag is set in the exchange layer |
| Message unpacking fails | - Notifies the middleware  
- Updates the wire message as in an errored state  
- Creates a business message in an errored state for the wire message |
| Incoming message decoding fails | - Notifies the middleware  
- Updates the wire message as in an errored state  
- Creates a business message in an errored state for the wire message  
- Sends an exception message to the trading partner, if the sendException flag is set in the exchange layer |
| The message is duplicated | - Notifies the middleware  
- Updates the wire message as a duplicated message error  
- Creates a business message as a duplicated message error for the wire message |
| Document identification fails | - Notifies the middleware  
- Updates the wire message as in an errored state  
- Creates a business message in an errored state for the wire message  
- Sends an exception message to the trading partner, if the sendException flag is set in the exchange layer |
| Incoming trading partner agreement processing fails | - Notifies the middleware  
- Updates the wire message as in an errored state  
- Creates a business message in an errored state for the wire message  
- Sends an exception message to the trading partner, if the sendException flag is set in the exchange layer |
| Incoming document processing fails | - Notifies the middleware  
- Updates the wire message as in an errored state  
- Creates a business message in an errored state for the wire message  
- Sends an exception message to the trading partner, if the sendException flag is set in the exchange layer |

Note the following:

- The exception is sent back to the trading partner only for RosettaNet exchanges. For other exchanges, a failure is reported as mandated in the respective specifications. For example, for an ebMS exchange, an acknowledgment is sent along with the error list that is defined. For an AS2 exchange, the acknowledgment is sent indicating an error, without exception details.
An exception is sent back to the trading partner for all message types except acknowledgments.

D.6 Exception Payload Definition

Example D–1 shows the definition for the exception payload, ipException.xsd.

Example D–1 Exception Payload Definition

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns="http://integration.oracle.com/B2B/Exception"
           targetNamespace="http://integration.oracle.com/B2B/Exception">

  <xs:element name="Exception">
    <!--xs:complexType name='Exception'-->
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="correlationId"/>
        <xs:element ref="b2bMessageId"/>
        <xs:element ref="errorCode"/>
        <xs:element ref="errorText"/>
        <xs:element ref="errorDescription"/>
        <xs:element ref="errorSeverity"/>
        <xs:element ref="errorDetails" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="correlationId" type="xs:string"/>
  <xs:element name="b2bMessageId" type="xs:string"/>
  <xs:element name="errorCode" type="xs:string"/>
  <xs:element name="errorText" type="xs:string"/>
  <xs:element name="errorDescription" type="xs:string"/>
  <xs:element name="errorSeverity" type="xs:string"/>
  <xs:element name="errorDetails">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="parameter" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="parameter">
    <xs:complexType>
      <xs:attribute name="name" type="xs:string" use="required" />
      <xs:attribute name="value" type="xs:string" use="required" />
    </xs:complexType>
  </xs:element>

</xs:schema>
```
A
Administration page, 2-5
administrator role, 5-7
agreements
  active, 8-4
    exporting, 8-4
  creating, 6-2
  definition, 6-1
deleting, 6-7
deploying, 2-13, 6-5
deployment states, 8-1
functional acknowledgements, 6-4
redeploying, 6-6
trading partners, 2-12
translation, 6-4
validation, 6-4
Analyzer, 3-11
Applicability Statement 1 (AS1)
  channel, 5-13
  identifier, 9-2
Applicability Statement 2 (AS2)
  exchange protocol, 5-13
  identifier, 9-2
application message reports, 16-7
AS2
  See Applicability Statement 2 (AS2) exchange protocol
Auto Create Agreement, 5-34

B
B2B binding components
  in SOA composite, 2-15
B2B Configuration Wizard, 2-17
B2B.IP_MESSAGE_TYPE, C-1
B2B reference, 2-15
B2B service, 2-15
B2B transaction, 2-9
b2b.checkDuplicate, B-3
b2bcpaexport command line tool, 18-6
b2bcpaimport command line tool, 18-6
b2bcreate-cpaprop command line tool, 18-4
b2b.defaultThreadCount property, B-1
b2b.defaultThreadSleepTime property, B-1
b2b.deploy.validation property, B-1
b2b.edi.ignoreValidation property, B-3
b2bexport command line tool, 18-3
b2b.FAHandledByB2B property, B-2
b2b.FAInternalProperties property, B-2
b2b.HAInstanceName property, B-2
B2BHeader.sendException flag, D-1, D-2
b2bimport command line tool, 18-3
b2b.inboundThreadCount property, B-1
b2b.inboundThreadSleepTime property, B-1
b2b.mdsCache property, B-1
b2b.outboundOneErrorAllError property, B-3
b2b.outboundThreadCount property, B-1
b2b.outboundThreadSleepTime property, B-1
b2b.payloadObfuscation property, B-3
b2bpurge command line tool, 18-2
b2b.r1ps1 property, 1-6
B2BUser, 10-4
batched messages
  creating, 11-3
  managing, 11-4
  scheduling, 11-3
  searching for agreements, 11-2
  setting up, 11-1
business message reports, creating, 16-2
business-to-business e-commerce
  comparison to traditional transactions, 1-2
description, 1-1

C
Callout Directory parameter, 12-6
callouts
  code example, 12-8
  creating, 12-4
  creating a library JAR file, 12-4
  defining callout JAR file library location, 12-6
  definition, 12-1
differently formatted XML messages, 12-2
  implementation class, 12-5
  implementing, 12-8
  including in an agreement, 12-7
  library name, 12-6
  parameter attributes, 12-6
  predefined XSLTCalloutImpl class file, 12-5
  transport, 12-3
channel attributes
configuring, 5-21
CLIENT-CERT authentication method, 1-5
cloning trading partners, 5-2
command line tools
  b2bcaexport, 18-6
  b2bcaimport, 18-6
  b2bcreate-cpaprop, 18-4
  b2bexport, 18-3
  b2bimport, 18-3
  b2bpurge, 18-2
  introduction, 18-1
  prerequisites for running, 18-1
configuration parameters, setting, 15-1
control number information, purging, 13-1
conversation report, creating, 16-12
CPP/CPA
  exporting, 18-6
  importing, 18-6
  templates, creating, 18-4
custom contact information types, creating, 9-4
custom document protocols
  configuring XPath expressions, 7-4
  definition, 7-2
document definition parameters, 7-3
document type parameters, 7-2
document version parameters, 7-2
custom headers
  extracting, 5-32
custom identifier types, creating, 9-1
custom trading partner parameter types, creating, 9-4

D
Data Generator, 3-9
deployed agreement states
  active, 8-2
  changing, 8-4
  inactive, 8-2
  purged, 8-2
  retired, 8-2
deployed agreements
  managing, 8-2
  metrics, 2-8
  searching, 8-2
dequeuing
  See Java dequeue command
design metadata, 13-1
design-time repository
  description, 10-1
  exported file, 10-5
  exporting, 10-2
  importing, 10-2
  what is imported or exported, 10-4
document definitions
  adding, 5-11
  changing after importing metadata, 7-31
  creating, 2-11, 4-3
  definition, 4-1
  deleting, 4-7

EDI EDIFACT, 7-7
  example, 4-3
  trading partner agreements, 6-3
document guidelines
  available types, 3-1
  example, 3-3
document protocols
  custom, 7-2
  definition, 7-1
  EDI, 3-1
  EDI EDIFACT, 7-7
  EDI X12, 4-2, 7-11
  HL7, 1-2, 1-10, 3-1, 7-15
  introduction, 4-1
  managing, 2-5
  NCPDP Telecom, 3-1
  ParserSchema, 3-1
  Partner Interface Process (PIP), 7-22
  Positional Flat File Document Protocol, 7-20
  proprietary transactions, 7-2
  RosettaNet, 1-2, 1-3, 1-10, 3-1, 7-21
  supported, 1-2
  UCCnet, 1-2, 1-3, 7-25
  Document Routing IDs, 7-31
document types
  restricting access, 1-6
  DUNS, 9-2
dynamic endpoints, 5-31

E
edi MS Identifier, 9-4
ebMS-1.0, 1-3, 5-13, 5-17
ebMS-2.0, 5-13, 5-17
ediXML messaging service (ebMS) parameters, 7-2
EDI
  document guidelines, 3-1
  Group ID, 9-2
  Group ID Qualifier, 9-2
  Interchange ID, 9-3
  Interchange ID Qualifier, 9-3
  Interchange Internal ID, 9-3
  Interchange Internal Sub ID, 9-3
EDI EDIFACT
  definition, 7-7
  document definition parameters, 7-10
  document definitions, 7-7
document protocols, 7-7
document type parameters, 7-9
document version parameters, 7-7
Oracle B2B support, 1-3
samples, 1-10
traditional transaction comparison, 1-2
UNA, 7-8
EDI X12
  definition, 7-11
  document definition parameters, 7-14
document hierarchy, 4-2
document type parameters, 7-13
document version parameters, 7-12
Oracle B2B support, 1-3
samples, 1-10
traditional transaction comparison, 1-2
enqueue, JMS JCA adapter, 19-4
enqueuing
See Java enqueue command
enqueuing and dequeuing, 19-1
error message reports, creating, 16-10
errorDescription, D-1
errorText, D-1
exception handling
description, D-1
inbound exception handling scenarios, D-4
inbound messages, D-1
acknowledgement messages, D-2
document identification failure, D-5
document processing failure, D-5
ebMS, AS1, and AS2 failures, D-2
exception messages, D-2
exchange identification failure, D-5
exchange not supported, D-5
message decoding failure, D-5
message duplicated, D-5
message unpacking failure, D-5
request or response messages, D-1
scenarios, D-4
trading partner agreement failure, D-5
outbound messages, D-2
payload definition, D-6
Exception Queue parameter, 15-4, D-3
exchange protocol parameters configuring, 5-25
exporting
active agreement, 8-4
CPP/CPA files, 18-6
data, 18-3
description, 10-1
design-time repository to file, 10-1
restrictions, 8-4, 10-1
types of export supported, 10-1

F
FusionOrderDemo_R1PS1.zip, 2-25

G
Generic AQ-1.0, 1-3, 5-13
Generic Email-1.0, 1-3, 5-14
Generic File-1.0, 1-3, 5-13
Generic FTP-1.0, 1-3, 5-14
Generic HTTP-1.0, 1-3, 5-14
Generic Identifier, 9-3
Generic JMS-1.0, 1-3, 5-14
Generic SFTP-1.0, 1-3, 5-14
generic support for TCP, 5-30

H
Health Level 7 (HL7)
Batch Application ID, 9-3
Batch Application Universal ID, 9-3
Batch Application Universal ID Type, 9-3
Batch Facility ID, 9-3
Batch Facility Universal ID, 9-3
Batch Facility Universal ID Type, 9-3
definition, 7-15
document definition parameters, 7-19
document guidelines, 3-1
document type parameters, 7-18
document version parameters, 7-16
File Application ID, 9-3
File Application Universal ID, 9-3
File Application Universal ID Type, 9-3
File Facility ID, 9-3
File Facility Universal ID, 9-3
File Facility Universal ID Type, 9-3
immediate acknowledgment, 7-20
Message Application ID, 9-3
Message Application Universal ID, 9-3
Message Application Universal ID Type, 9-4
Message Facility ID, 9-4
Message Facility Universal ID, 9-4
Message Facility Universal ID Type, 9-4
negative acknowledgment messages, 7-20
Oracle B2B support, 1-3
samples, 1-10
traditional transaction comparison, 1-2
host administrator role, 1-4
host monitor role, 1-4
host trading partner, 5-1
agreements, 6-3
key store information, 5-6

I
Identity Store, creating a new user, 5-7
import errors, 18-7
importing
CPP/CPA files, 18-6
data, 18-3
description, 10-1
design-time repository from file, 10-1
restrictions, 8-4, 10-1
instance data, 2-7, 13-1
instance metadata, 13-1
ipException.xsd file, D-3
enqueued exception, D-1, D-2, D-3
eception payload definition, D-6

J
Java dequeue command, 19-1
Java enqueue command, 19-1
JMS
dequeue properties, 19-5
enqueue and dequeue, 19-3
enqueue properties, 19-3
JCA adapter, 19-4
JMS adapter properties, 2-23
JMS integration type, 2-23
large payloads
  32-bit Windows, A-7
description, A-5
inbound setup, A-5
outbound setup, A-5
listening channel
  adding, 14-1
  adding channel attributes, 14-10
  adding details, 14-4
  adding exchange protocol parameters, 14-11
  adding transport protocol parameters, 14-10
configuring, 14-9
definition, 14-1
protocols, 14-2
logging in, user interface tool, 2-2
logPayload property, A-5

message exchange protocol
  ebMS-1.0, 1-3
ebMS-2.0, 1-3
Generic AQ-1.0, 1-3
Generic Email-1.0, 1-3
Generic File-1.0, 1-3
Generic FTP-1.0, 1-3
Generic HTTP-1.0, 1-3
Generic JMS-1.0, 1-3
Generic SFTP-1.0, 1-3
MLLP-1.0, 1-3
RosettaNet-01.10, 1-3
RosettaNet-V02.00, 1-3
message sequencing, 5-32
messages, purging, 16-2
Metadata Service (MDS)
  managing metadata, 1-4
  repository, 1-4
metrics
  introduction, 17-1
  summary data, 17-3
Metrics page, 2-8
MIME, 1-2
Minimum Lower Layer Protocol (MLLP)
  delivery channels, 5-29
  description, 5-13
  ID, 9-4
  message exchange protocol, 1-3
Monitor role, 16-1
monitor role, 5-7

NCPDP Telecom
  document guidelines, 3-1
normalized message properties, C-1
  mapping IP_MESSAGE_TYPE to, C-1

Oracle B2B
  adding a channel, 5-15
  adding a user, 5-9
  Administration page, 2-5
  as a binding component, 1-8
  available channels, 5-13
  channel attributes, 5-21
  command line tools, 18-1
  configuration parameters, 15-1
  document protocols, 4-1
  exception handling, D-1
  getting started, 2-1
  in a SOA implementation, 1-8
  in JDeveloper environment, 2-14
  introduction, 1-1
logging in, 2-2
metadata, 1-3
metrics, 17-1
Metrics page, 2-8
partner metrics, 17-4
Partners page, 2-6
port information, 2-3
process flow, 3-1
Reports page, 2-7
resubmitting messages, 16-2
samples, 1-10
security features, 1-4
supported protocols, 1-2
trading partners, 5-1
transport protocols, 14-3
user interface, 2-5
user interface, accessing, 2-2
Oracle B2B Document Editor, 2-1, 2-10
Analyzer, 3-11
Data Generator, 3-9
document definitions, 4-3
document guidelines, 3-1
installation, 3-3
introduction, 3-1
Oracle BPEL Process Manager, 1-9
Oracle Business Activity Monitoring (BAM), 1-9
Oracle Enterprise Manager Fusion Middleware Control, 1-11
Oracle JDeveloper, 2-1, 2-14
Oracle Mediator, 1-9
Oracle Platform Security Services, 1-4
Oracle SOA Suite overview, 1-8
Oracle WebLogic Server Administration Console
  administering Oracle B2B, 1-10
  Security Realms, 5-7

ParameterValue, 10-4
ParserSchema, document guidelines, 3-1
Partner Interface Process (PIP)
  document definition parameters, 7-24
document type parameters, 7-23
document version parameters, 7-23
introduction, 7-22
partner metrics, 17-4
Partners page, 2-6
payload obfuscation, 1-5
performance tuning, 1-10
description, A-1
heap size settings, A-2
JTA settings, A-4
large payloads, A-5
MDS cache size, A-2
memory arguments, A-2
number of threads, A-2, B-2
Stuck Thread Max Time, A-2
tablespace, A-4
positional flat files
document definition parameters, 7-20
document type parameters, 7-20
document version parameters, 7-20
introduction, 7-20
predefined identifier types, 9-2
process flow, overview, 2-9
properties
setting in Enterprise Manager, B-1
setting on the Configuration tab, 15-1
proprietary transactions, 7-2
protocols
Custom, 7-2
document, 7-1
EDI EDIFACT, 7-7
EDI X12, 7-11
HL7, 7-15
positional flat file, 7-20
RosettaNet, 7-21
UCCnet, 7-25
purging
data, 18-2
design metadata, 13-1
messages, 16-2
R
reference, 1-8
remote administrator role, 1-4
remote monitor role, 1-4
remote trading partners, 5-3
agreements, 6-3
definition, 5-1
reports
business message status, 16-2
conversation message status, 16-12
description, 16-1
error message status, 16-10
monitor user role, 16-1
Reports page, 2-7
restricting access to document types, 1-6
resubmitting messages, 16-2
roles
administrator role, 5-7
host administrator, 1-4
host monitor, 1-4
monitor role, 5-7
remote administrator, 1-4
remote monitor, 1-4
RosettaNet
definition, 7-21
dictionaries, 7-22
document guidelines, 3-1
document type definition (DTD), 7-21
Message Guideline specification, 7-25
Oracle B2B support, 1-3
Partner Interface Process (PIP), 7-22
samples, 1-10
traditional transaction comparison, 1-2
validation, 7-25
RosettaNet-01.10, 1-3, 5-13, 5-17
RosettaNet-V02.00, 1-3, 5-13, 5-17
run-time
data, 2-7
metadata, 13-1
S
sample SOA implementation, 1-8
SCA normalized message properties, C-1
Secure Socket Layer, 1-4
security
configuring, 5-27
features, 1-4
Security Realms, 5-7
SegmentCount macro, 3-9
service, 1-8
service components
introduction, 2-17
setSOADomain.sh script, A-2
SMIME, 1-2
SOA composite application, 1-8
SOA Server, 1-10
SOAP, 1-2
supported protocols, 1-2
system parameters, setting, 15-1
T
trading partner agreements
creating, 2-12, 6-1, 6-2
definition, 6-1
deleting, 6-7
deploying, 6-5
functional acknowledgements, 6-4
host trading partner, 6-3
redeploying, 6-6
remote trading partner, 6-3
translation, 6-4
validation, 6-4
trading partners, 2-6, 2-12
adding users, 5-7
cloning, 5-2
configuring, 5-1
contact information, 5-5
creating a profile, 5-2
creating custom parameter types, 9-4
custom identifier types, 9-1
definition, 5-1
host, 5-1
host key store information, 5-6
identifying types and values, 5-4
overview, 5-1
parameters and values, 5-5
remote, 5-1, 5-3
transport callouts, 5-32, 12-3
transport protocol parameters
configuring, 5-16
transport protocols
available in Oracle B2B, 14-3
using, 14-3

U

UCCnet
definition, 7-25
document definition parameters, 7-26
document type parameters, 7-26
document version parameters, 7-26
Oracle B2B support, 1-3
traditional transaction comparison, 1-2
UNA, 7-8
Unicode support, 2-16
Use JMS Queue as default parameter, 15-4, D-3
user interface
accessing, 2-2
main page, 2-3
users
accessing user interface tool, 2-2
admin user name, 2-2

X

XMLDSig, 1-2
XMLEncrypt, 1-2
XMLSchema, document guidelines, 3-1
XSLTCalloutImpl class file, 12-5