Oracle® Fusion Middleware

User's Guide for Oracle B2B

11*g* Release 1 (11.1.1)

E10229-01

May 2009

ORACLE®

Oracle Fusion Middleware User's Guide for Oracle B2B, 11*g* Release 1 (11.1.1)

E10229-01

Primary Author: D. K. Bradshaw

Contributing Author: Mark Kennedy

Contributor: Oracle B2B development, product management, and quality assurance teams

Contents

Preface ix

Audience ix
Documentation Accessibility ix
Related Documents x
Conventions x

Part I Introduction to Oracle B2B

1 Introduction to Oracle B2B

Oracle B2B and Business-to-Business E-Commerce 1-1
Protocols Supported in Oracle B2B 1-2
Features in Preview Mode 1-4
Oracle B2B Metadata 1-5
Security Features of Oracle B2B 1-5
How Does Oracle B2B Fit into a SOA Implementation? 1-6
Sending a Purchase Order: An Example of a SOA Implementation 1-6
Oracle B2B Samples 1-8
Administering Oracle B2B in the Oracle Fusion Middleware Environment 1-8

2 Getting Started with Oracle B2B

What You Need to Get Started with Oracle B2B 2-1
Logging in to Oracle B2B 2-2
Finding Port Information 2-3
Enabling the weblogic User for Logging in to Oracle B2B 2-4
Using the Oracle B2B Interface 2-4
Administration 2-5
Partners 2-5
Reports 2-6
Metrics 2-7
Creating a B2B Transaction: An Overview of the Process Flow 2-8
Using Oracle B2B in the Oracle JDeveloper Environment 2-13
What You May Need To Know About Using Oracle B2B 2-14
Enabling Debug Mode at Run Time 2-14
Logging Out: SSO Logout Configuration for Oracle Access Manager 2-14

Part II Oracle B2B Process Flow

3 Creating Guideline Files

Introduction to Oracle B2B Document Editor 3-1
Installing Oracle B2B Document Editor 3-3
Creating Guideline Files: EDIFACT D98 Example 3-3

4 Creating Document Definitions

Introduction to Document Protocols 4-1
Creating Document Definitions 4-3
Deleting a Document Definition 4-7

5 Configuring Trading Partners

Introduction to Trading Partners 5-1
Creating Trading Partner Profiles 5-2
Adding Trading Partner Users 5-7
Adding Document Definitions 5-9
Configuring Channels 5-10
About MLLP 5-20
Adding Channel Details 5-21
Using the Auto Create Agreement Feature 5-29
Using Identifiers for Trading Partner Lookup 5-30

6 Creating and Deploying Trading Partner Agreements

Introduction to Agreements 6-1
Creating an Agreement 6-2
Deploying an Agreement 6-5
Redeploying an Agreement 6-6
Deleting and Exporting Agreements 6-6

Part III Oracle B2B Administration

7 Using Document Protocols

Using the Custom Document Protocol 7-2
How to Configure the XPath Expression for a Custom XML Document 7-5
Option 1: Specify the XPath and the Matching Value 7-5
Option 2: Check for the Existence of a Node 7-6
Option 3: Check the Value of an Attribute 7-7
Using the EDI EDIFACT Document Protocol 7-7
Using the EDI X12 Document Protocol 7-12
Using the HL7 Document Protocol 7-16
Using the OAG Document Protocol 7-20
Using the Positional Flat File Document Protocol 7-22
Using the RosettaNet Document Protocol 7-24
PIPs 7-24

RosettaNet Dictionaries 7-28
RosettaNet Validation 7-28
Using the UCCnet Document Protocol 7-29
Summary of Document Protocol Parameter Types 7-31
Changing Document Definitions 7-32
Changing Document Definitions After Deploying an Agreement 7-33
Changing Document Definitions After Importing Metadata 7-33
Using Document Routing IDs 7-33

8 Managing Deployments

Introduction to Agreement Deployment States 8-1
Managing Deployed Agreements 8-1
Searching for Deployed Agreements 8-2
Changing the Deployment State 8-3
Exporting an Active Agreement 8-3

9 Creating Types

Creating Custom Identifier Types 9-1
Creating Custom Contact Information Types 9-4
Creating Custom Trading Partner Parameter Types 9-4

10 Importing and Exporting Data

Importing and Exporting the Design-Time Repository 10-1
What Is Copied When You Import or Export from the Import/Export Tab 10-3
About the Exported File 10-4

11 Batching EDI Messages

Setting Up a Batch 11-1
Managing Batched Messages 11-4

12 Managing Callouts

Introduction to Callouts 12-1
Creating a Callout Library JAR File 12-3
Creating a Callout 12-3
Including a Callout in an Agreement 12-5
Implementing a Callout 12-6

13 Purging Data

Purging Design Metadata and Instance Data 13-1

14 Configuring Listening Channels

Adding a Listening Channel and Protocol 14-1
Using Transport Protocols 14-3
Adding Listening Channel Details 14-4

Configuring a Listening Channel 14-7

15 Configuring B2B System Parameters

Setting Configuration Parameters 15-1

Part IV Reports and Metrics

16 Creating Reports

Introduction to Reports 16-1
The Monitor User Role 16-1
Purging Messages 16-2
Resubmitting Messages from Oracle B2B 16-2
Creating Business Message Reports 16-2
Creating Wire Message Reports 16-5
Creating Application Message Reports 16-7
Creating Error Reports 16-10
Creating Conversation Reports 16-12

17 Using B2B Metrics

Introduction to B2B Metrics 17-1
B2B System Metrics 17-2
B2B Partner Metrics 17-3

Part V Scripts and Utilities

18 B2B Command Line Tools

Prerequisites for Running the Command Line Tools 18-1
Purging Data 18-2
Importing Data 18-2
Exporting Data 18-3
Deploying Agreements 18-4
Validating B2B Metadata 18-4
CPP/CPA Templates 18-5
CPP/CPA Import 18-5
CPP/CPA Export 18-6
Errors During Import 18-6

19 Scripts for Archiving and Restoring Data

Introduction to Archiving and Restoring B2B Business Messages 19-1
Archiving B2B Business Messages 19-1
Restoring B2B Business Messages 19-2

20 Utilities for Enqueuing and Dequeuing

AQ Enqueue and Dequeue Utilities 20-1
AQ Enqueue 20-1

AQ Dequeue ... 20-2
JMS Enqueue and Dequeue Utilities ... 20-3
JMS Enqueue ... 20-3
Enqueue—Using a JMS JCA Adapter or Custom Utilities ... 20-4
JMS Dequeue ... 20-4

Part VI Appendixes

A Performance Tuning and Large Payloads

Settings for Performance Tuning ... A-1
Memory Arguments ... A-1
Heap Size Settings ... A-2
MDS Cache Size ... A-2
Number of Threads ... A-2
Stuck Thread Max Time ... A-3
Tablespace ... A-4
JTA Settings ... A-5
Handling Large Payloads ... A-5
Introduction to Large Payload Support ... A-5
Large Payloads and 32-Bit Windows PCs ... A-7

B Properties of b2b-config.xml

Turning off Validation During Deployment ... B-1
MDS Cache Size ... B-1
Number of Threads ... B-1
Setting Up File, FTP, or Email in an HA Environment ... B-1
Setting Internal Properties for a Functional Acknowledgment ... B-2
Setting b2b.FAHandleByB2B for EDI EDIFACT and EDI X12 ... B-2
Setting the b2b.outboundOneErrorAllError Parameter ... B-2

C Back-End Applications Interface

Mapping B2B IP_MESSAGE_TYPE to SCA Normalized Message Properties ... C-1

D Exception Handling

Inbound Messages ... D-1
Request or Response Messages ... D-1
Acknowledgment Messages ... D-2
Exception Messages ... D-2
Failures with Inbound ebMS, AS1, and AS2 Messages ... D-2
Outbound Messages ... D-2
Inbound Exception Handling Scenarios ... D-3
Exception Payload Definition ... D-4

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 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:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

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"

1

Introduction to 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. The Oracle SOA Suite platform, of which Oracle B2B is a binding component, enables the implementation of e-commerce business processes. 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:

- Oracle B2B and Business-to-Business E-Commerce
- Protocols Supported in Oracle B2B
- Features in Preview Mode
- Oracle B2B Metadata
- Security Features of Oracle B2B
- How Does Oracle B2B Fit into a SOA Implementation?
- Sending a Purchase Order: An Example of a SOA Implementation
- Oracle B2B Samples
- Administering Oracle B2B in the Oracle Fusion Middleware Environment

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.

Table 1–1 Comparing Traditional and E-Commerce Transactions: An Example

	Traditional Shipping Transaction	E-Commerce Transaction
What is the item to be shipped, that is, the transaction item?	Cell phone	Electronic document Document protocols: Custom, EDI EDIFACT, EDI X12, HL7, OAG, positional flat file, RosettaNet, UCCnet, and more
How is the item packaged?	Box, bubble wrap	Packaging protocols: MIME, SMIME, SOAP, XMLDSig, XMLEncrypt
How is the item sent and received?	Truck, ship, airplane	Transport protocols: HTTP, File, FTP, SFTP (SSH FTP), TCP/IP, SMTP, MLLP
Who is the carrier?	DHL, FedEx, UPS, USPS	Message exchange protocols: RNIF, AS1, AS2, ebMS
What carrier services are required?	Required? ■ Signed receipt ■ Overnight/next day ■ Delivery attempts	Required? ■ Nonrepudiation ■ Time to acknowledge/respond ■ Retry counts

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.

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. Protocols marked with an asterisk (*) are in preview mode in this release.

Table 1–2 Protocols Supported in Oracle B2B

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

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.

Features in Preview Mode

The following document protocols are in preview mode for this release:

- OAG

  See "Using the OAG Document Protocol" on page 7-20 for more information about this protocol.

- Positional flat files, including SAP iDocs

  See "Using the Positional Flat File Document Protocol" on page 7-22 for more information about this protocol.

- UCCnet

  See "Using the UCCnet Document Protocol" on page 7-29 for more information about this protocol.

- non-XML Custom

  See "Using the Custom Document Protocol" on page 7-2 for more information about this protocol.

- NCPDP Telecom
- EDIEL

The following message exchange protocols are in preview mode for this release:

- AS1

  See Chapter 5, "Configuring Trading Partners," and Chapter 14, "Configuring Listening Channels," for more information about AS1.

Command-line tools for the following operations are in preview mode for this release:

- CPP/CPA Templates

  See "CPP/CPA Templates" on page 18-5 for more information.

- CPP/CPA Import

  See "CPP/CPA Import" on page 18-5 for more information.

- CPP/CPA Export

  See "CPP/CPA Export" on page 18-6 for more information.

The following B2B integration type is in preview mode for this release:

- JMS

  See "Using Oracle B2B in the Oracle JDeveloper Environment" on page 2-13 for more information, and see the Help for the B2B Configuration Wizard in Oracle JDeveloper.

Scripts for archiving and restoring data are in preview mode for this release. See Chapter 19, "Scripts for Archiving and Restoring Data," for more information.

Note: Use preview mode features with caution, particularly in production environments.

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 managed through Metadata Services (MDS), available in Oracle Fusion Middleware. See *Oracle Fusion Middleware Administrator's Guide* for more information about MDS.

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. When logged in as the default administrator user, you 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.

See "Adding Trading Partner Users" on page 5-7 for instructions on how to assign roles.

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 (CSF) 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 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*

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.

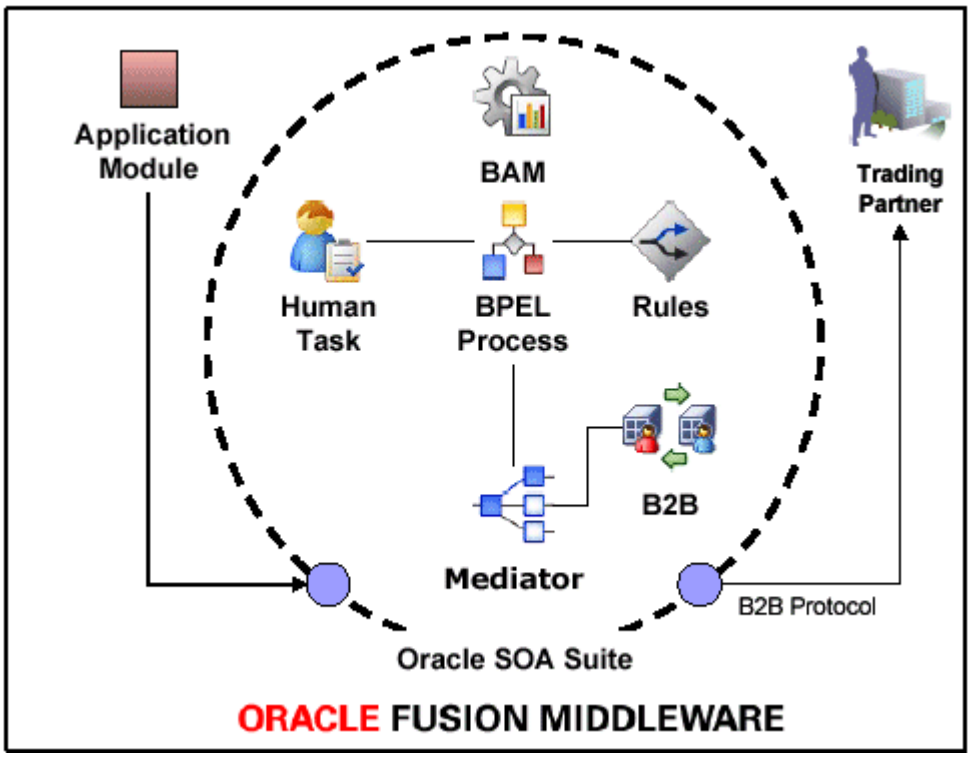
In addition to messages, Oracle B2B can also send attachments and large payloads in a SOA implementation. See Appendix A, "Performance Tuning and Large Payloads," for information about handling large payloads.

> **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 11*g* to communicate with Oracle E-Business Suite. This can be achieved by using the Oracle Application Adapter available in Oracle SOA Suite.

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 1–1.

Figure 1–1 An Outbound Purchase Order in a SOA Composite Application



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:

- "Using Oracle B2B in the Oracle JDeveloper Environment" on page 2-13 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

Oracle B2B Samples

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:

- Custom
- 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/`

See "What You Need to Get Started with Oracle B2B" on page 2-1 for the components required to use the samples.

Administering Oracle B2B in the Oracle Fusion Middleware Environment

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

- SOA Server—Set memory arguments to optimize B2B performance.

  See "Memory Arguments" on page A-1 for more information.

- b2b-config.xml file—Set the cache size for the Metadata Service instance. Also set the number of threads to improve B2B message processing.

  See "MDS Cache Size" on page A-2 and "Number of Threads" on page A-2 for more information.

- Oracle WebLogic Server Administration Console—Administer settings for performance tuning.

  See "Stuck Thread Max Time" on page A-3 for more information.

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

  See "Administering B2B" in *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite*.

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

- Reports link

  See Chapter 16, "Creating Reports."
- Metrics link

  See Chapter 17, "Using B2B Metrics."
- Administration > Configuration tab

  See Chapter 15, "Configuring B2B System Parameters."

2

Getting Started with Oracle B2B

Oracle B2B provides a Web-based interface for creating B2B transactions.

This chapter contains the following topics:

- What You Need to Get Started with Oracle B2B
- Logging in to Oracle B2B
- Using the Oracle B2B Interface
- Creating a B2B Transaction: An Overview of the Process Flow
- Using Oracle B2B in the Oracle JDeveloper Environment
- What You May Need To Know About Using Oracle B2B

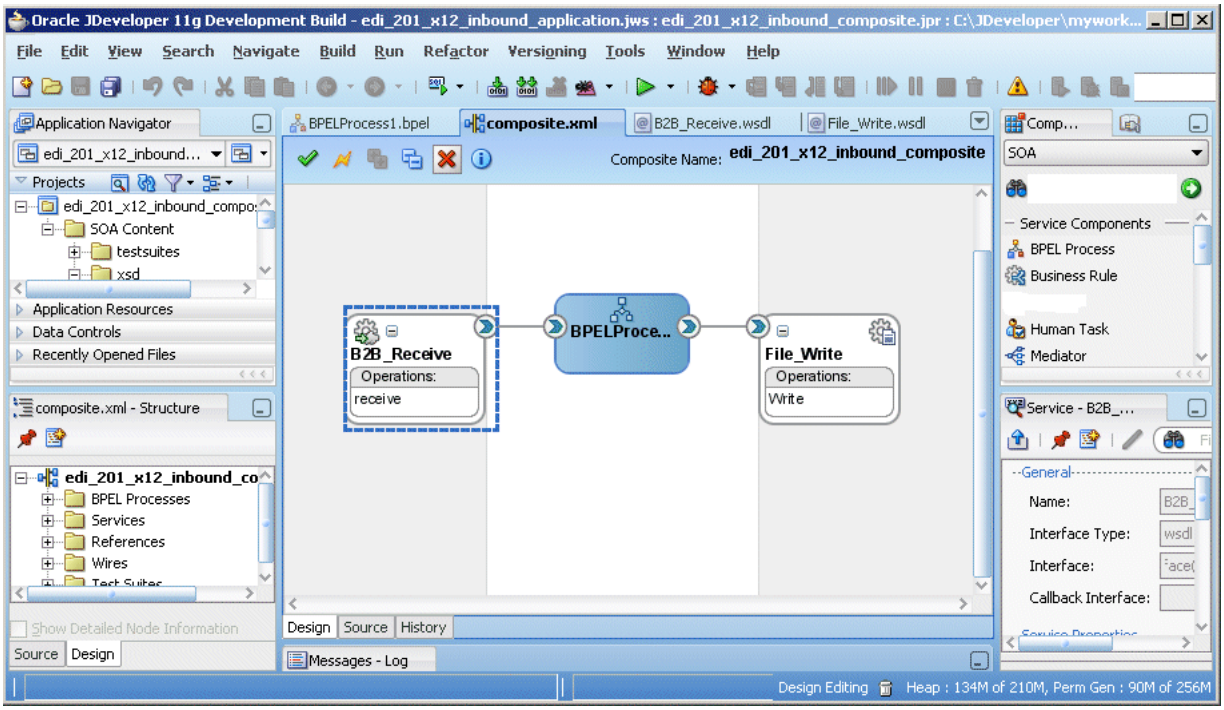
What You Need to Get Started with Oracle B2B

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.

Figure 2–1 Oracle JDeveloper: A SOA Composite Application with a B2B Binding Component



See the following for more information:

- *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite* for information on installing Oracle B2B as part of Oracle SOA Suite
- Chapter 3, "Creating Guideline Files"
- Oracle B2B Document Editor **Help** menu
- Oracle JDeveloper **Help** menu

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/b2b`

   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 "Finding Port Information" on page 2-3 for more information.

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

`http://hostname:port/b2b/ssologin`

---

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

For This Field...	Do...
Username	Enter the default administrator user name.
Password	Use the administrator password from your Oracle Fusion Middleware 11*g* installation.

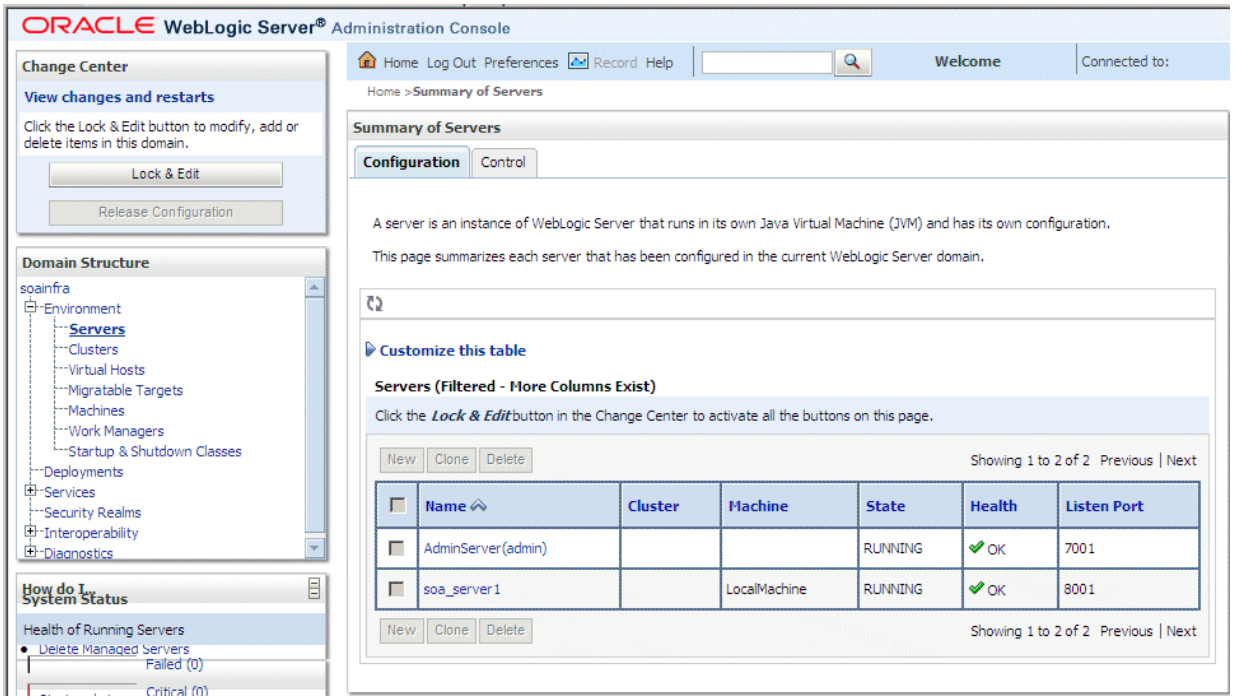
3. Click **Login**.

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, expand **Environment** and click **Servers**.



3. Note the **Listen Port** column for your server.

- From *FMW_HOME*/user_projects/domains/*your_domain_name*/config/config.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>
  <listen-address/>
</server>
```

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
```

Using the Oracle B2B Interface

B2B activities are grouped as follows:

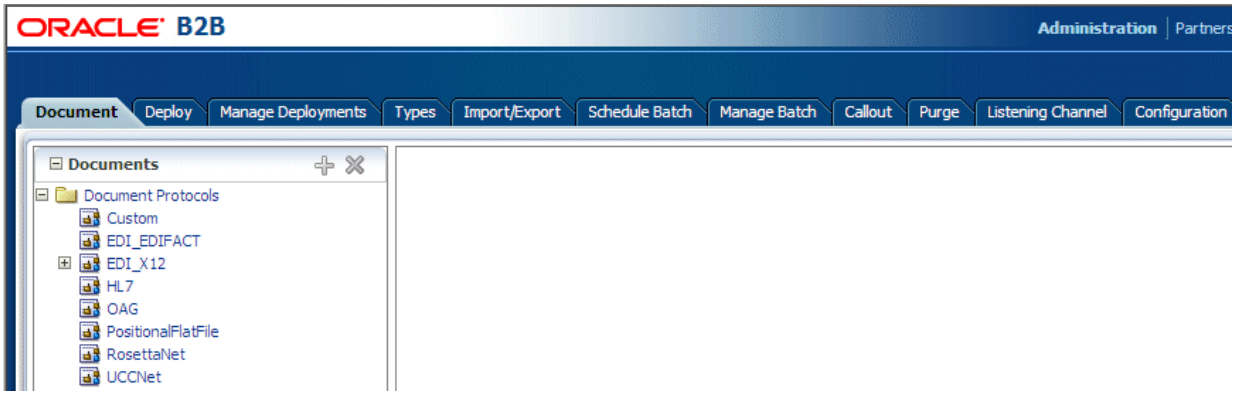
- Administration
- Partners
- Reports
- Metrics

Administration

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

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

Figure 2–2 Administration Activities

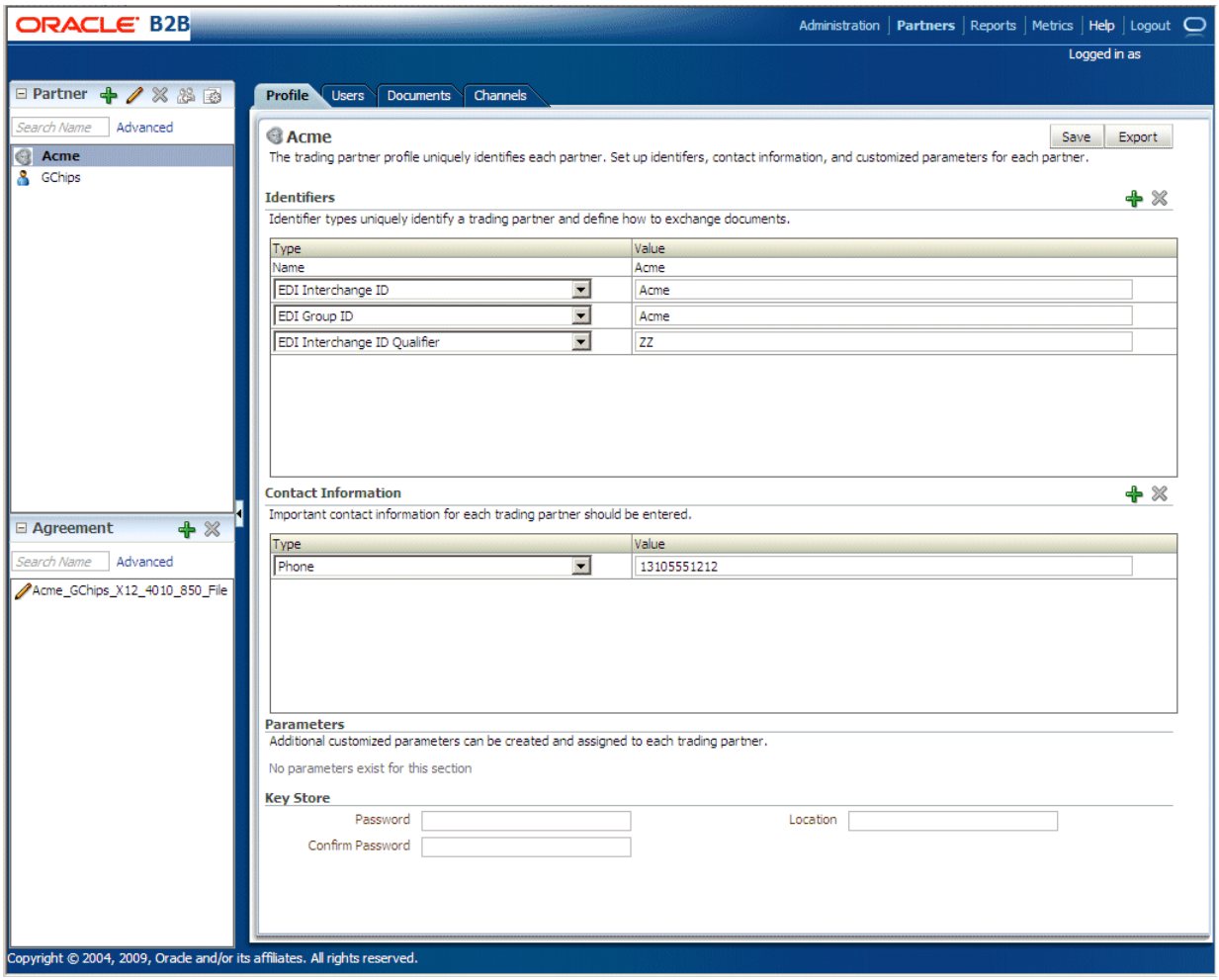


Partners

Use the tabs of the **Partners** page, shown in Figure 2–3, 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.

See Part II, "Oracle B2B Process Flow" for more information.

Figure 2–3 Partner Activities

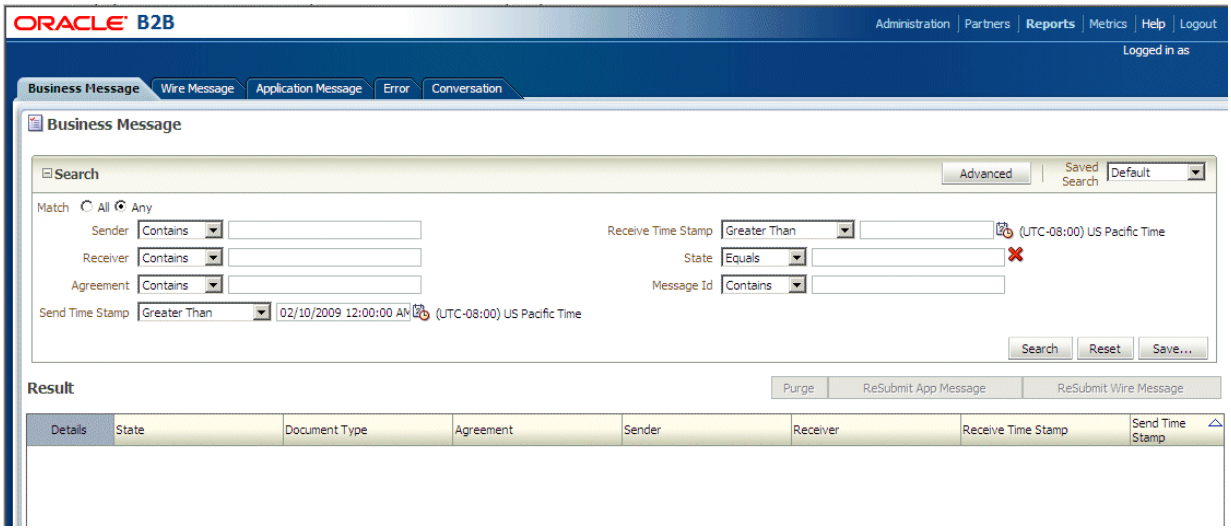


Reports

Use the tabs of the **Reports** page, shown in Figure 2–4, to create and view reports about the instance (run-time) data.

See Chapter 16, "Creating Reports," for more information.

Figure 2–4 Reports

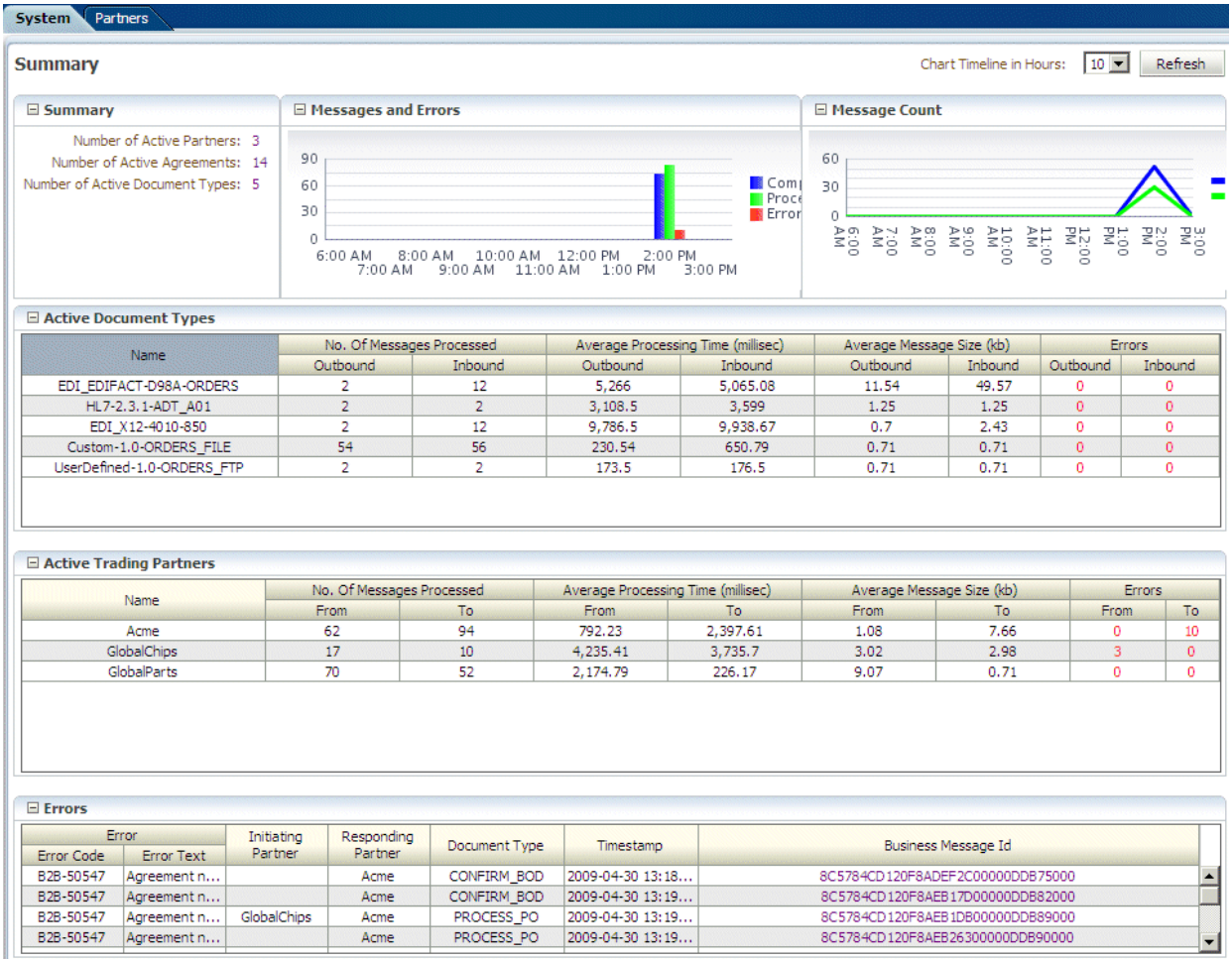


Metrics

Use the tabs of the **Metrics** page, as shown in Figure 2–5, 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.

See Chapter 17, "Using B2B Metrics," for more information.

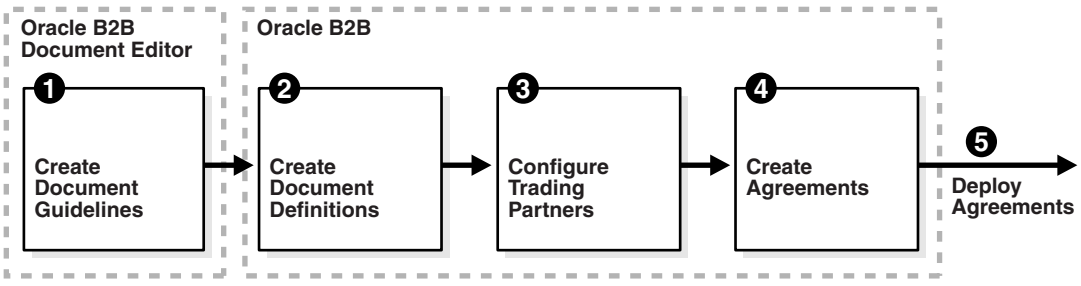
Figure 2–5 Metrics



Creating a B2B Transaction: An Overview of the Process Flow

Figure 2–6 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.

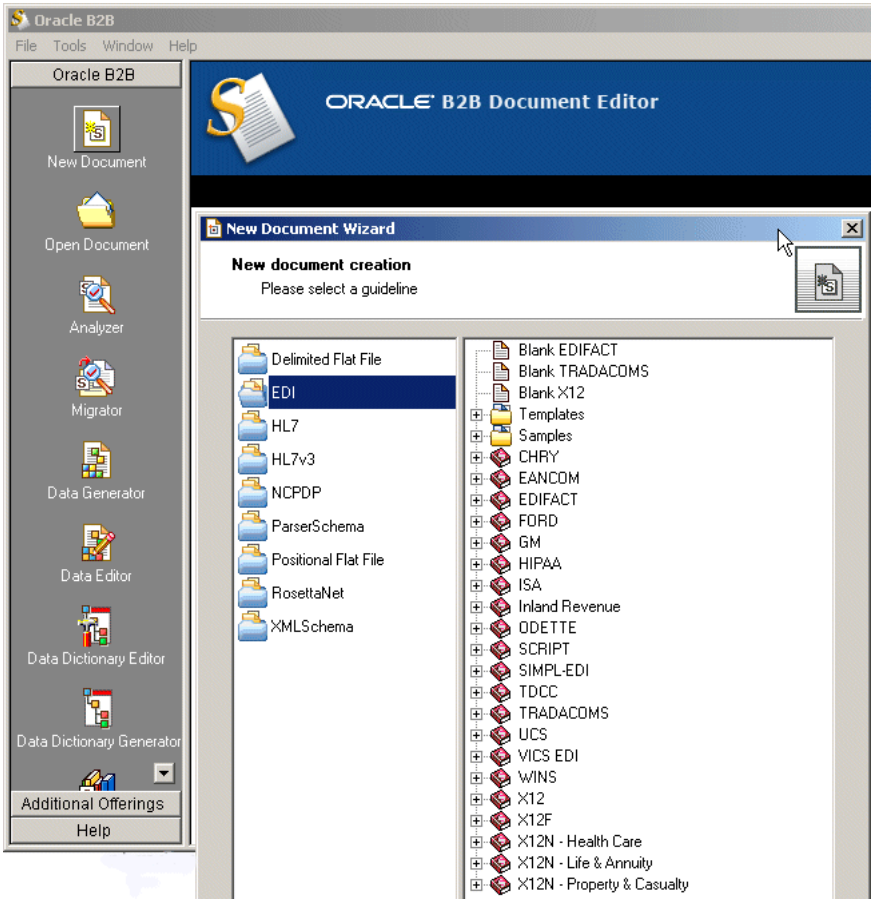
Figure 2–6 Oracle B2B Process Flow



Step 1: Create guideline files (ECS and optional XSD files) in Oracle B2B Document Editor

Using Oracle B2B Document Editor, shown in Figure 2–7, 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.

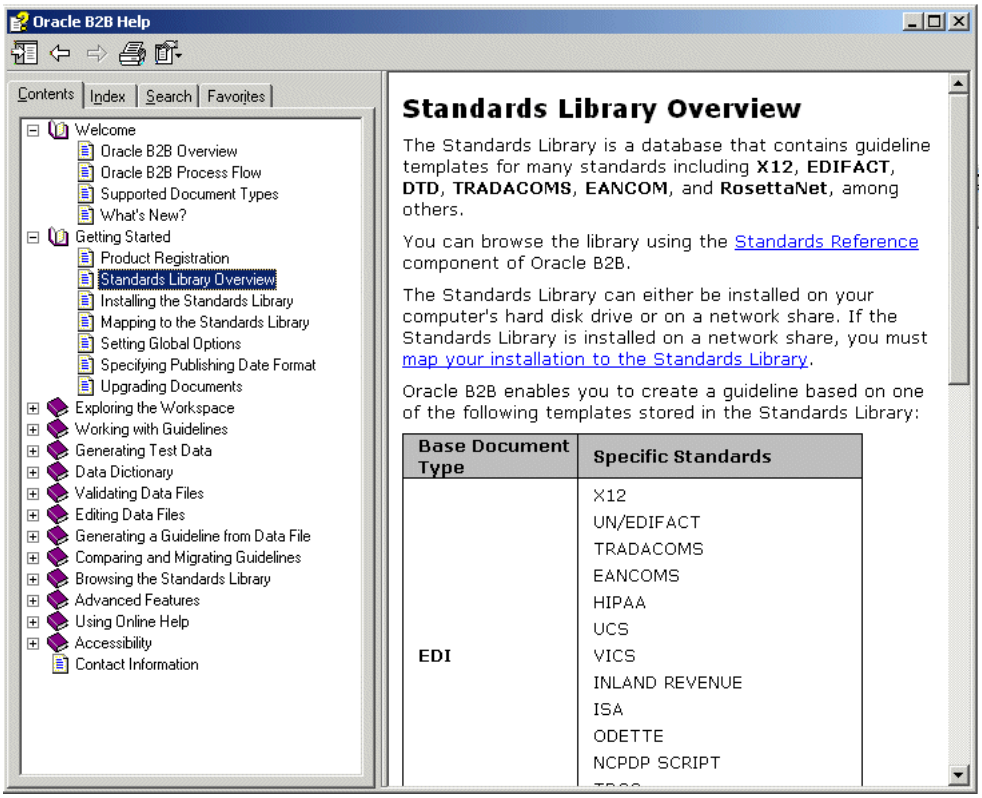
Figure 2–7 Creating a Document in Oracle B2B Document Editor



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

- Chapter 3, "Creating Guideline Files."
- The **Help** menu of Oracle B2B Document Editor, as shown in Figure 2–8.

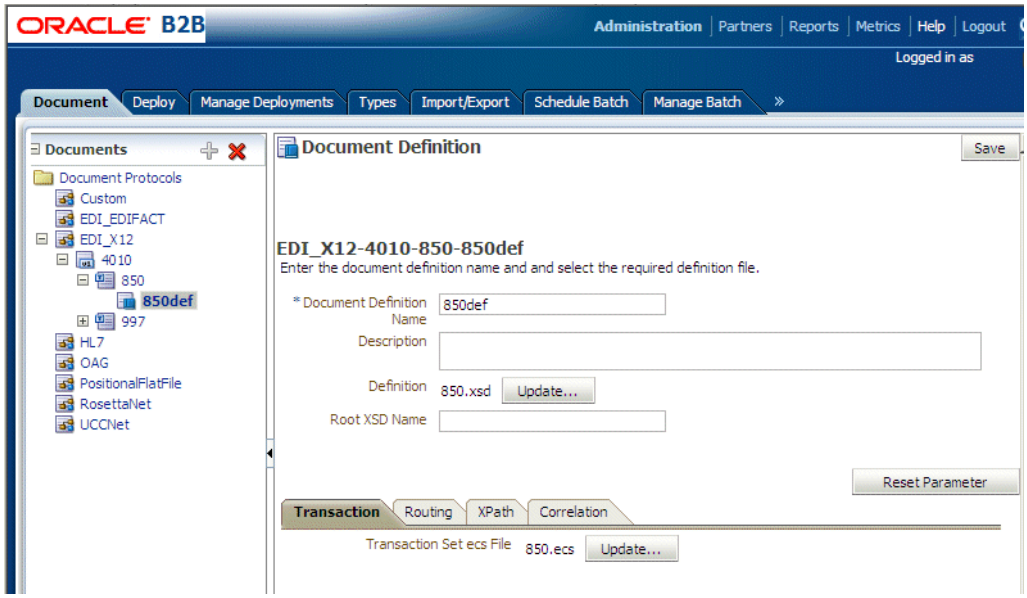
Figure 2–8 Oracle B2B Document Editor Help



Step 2: Create document definitions

Using the **Administration > Document** tab of Oracle B2B, shown in Figure 2–9, 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.)

Figure 2–9 Creating a Document Definition



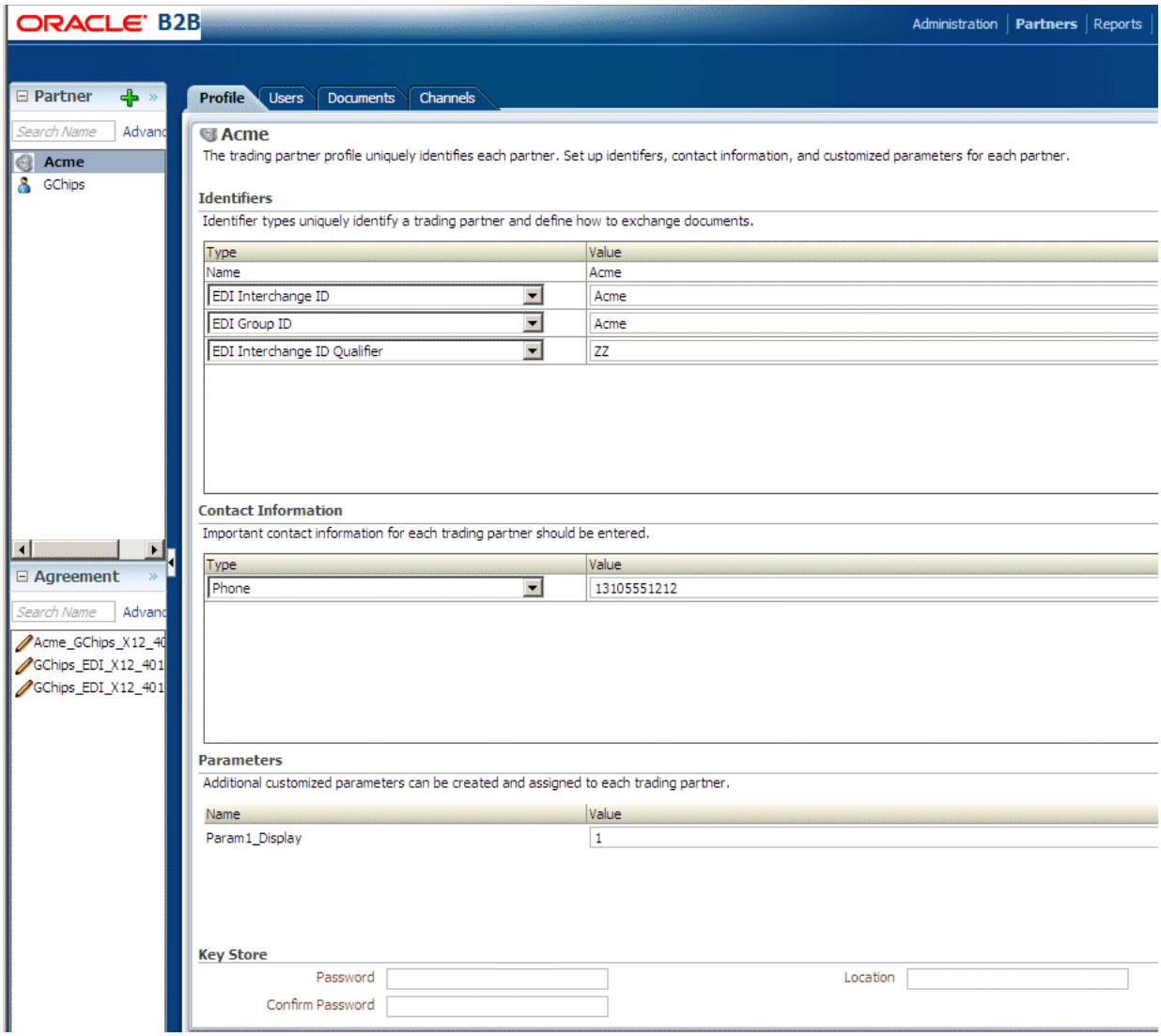
After selecting the ECS and optional XSD files you created in Step 1, you have created the document definition.

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 2–10, 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 2–10 Configuring Trading Partners

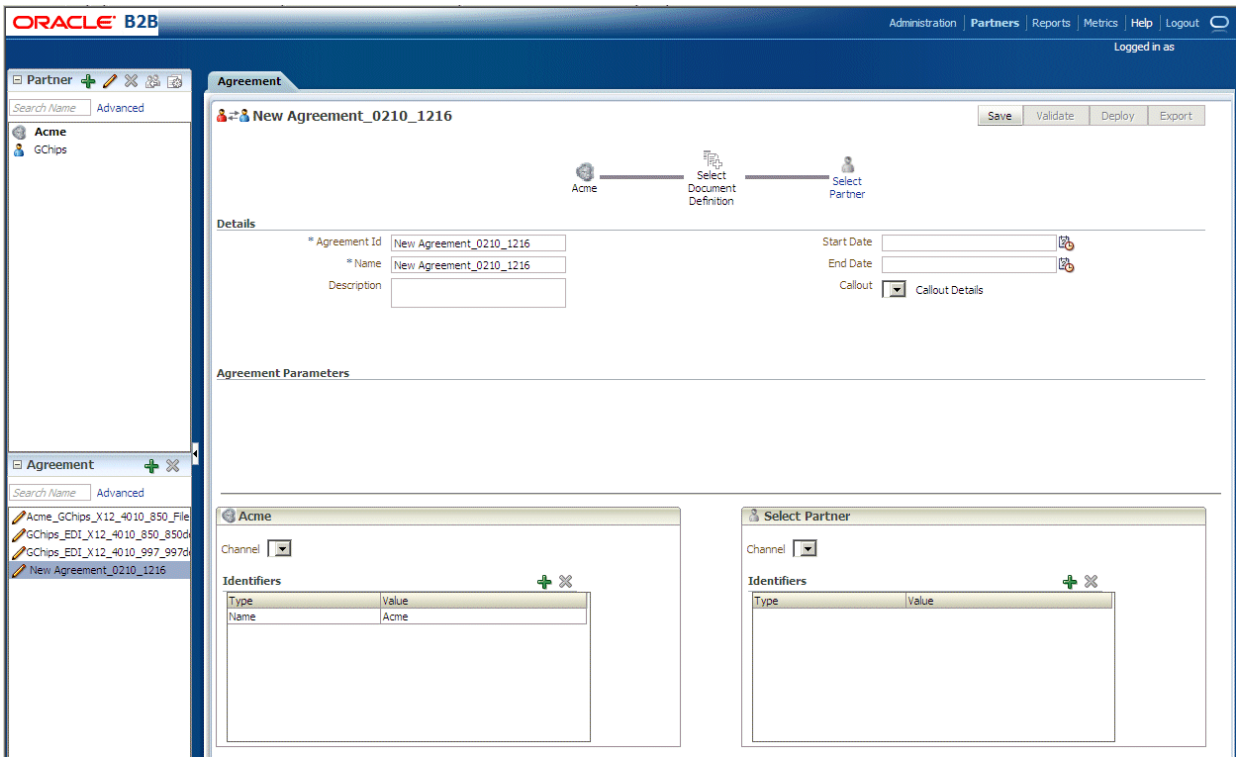


For more information, see Chapter 5, "Configuring Trading Partners."

Step 4: Create agreements

Using the **Partners > Agreement** tab of Oracle B2B, shown in Figure 2–11, create an agreement that specifies the trading partners involved and associates the document definitions, channels, and identifiers with the agreement.

Figure 2–11 Creating a Trading Partner Agreement

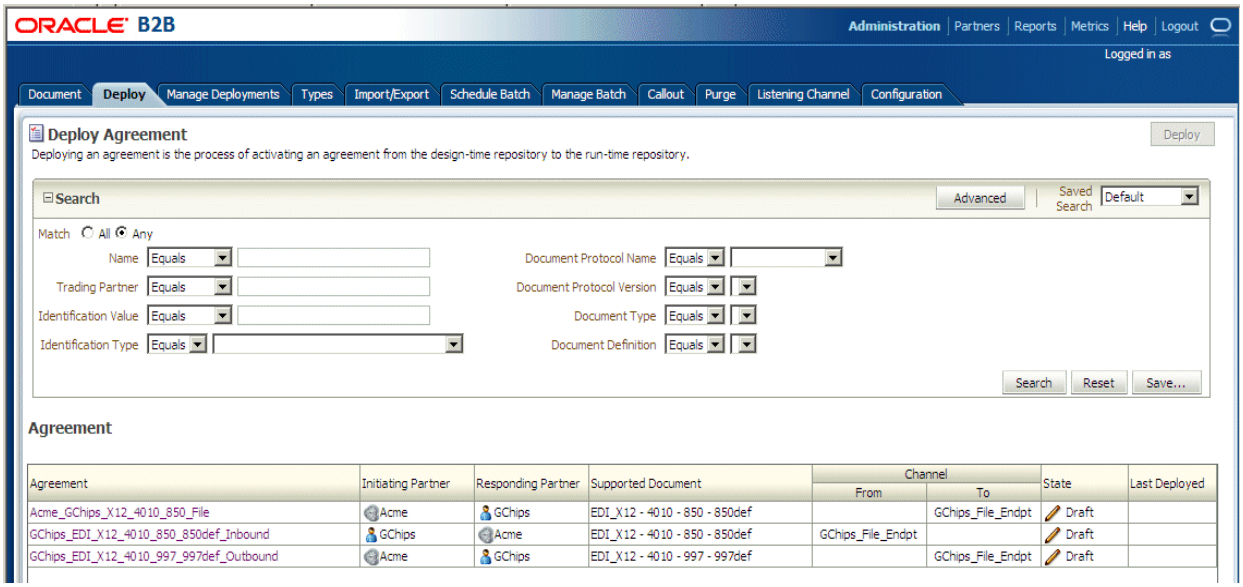


For more information, see Chapter 6, "Creating and Deploying Trading Partner Agreements."

Step 5: Deploy agreements

Using the **Administration > Deploy** tab of Oracle B2B, shown in Figure 2–12, search for and deploy agreements.

Figure 2–12 Searching for and Deploying Agreements

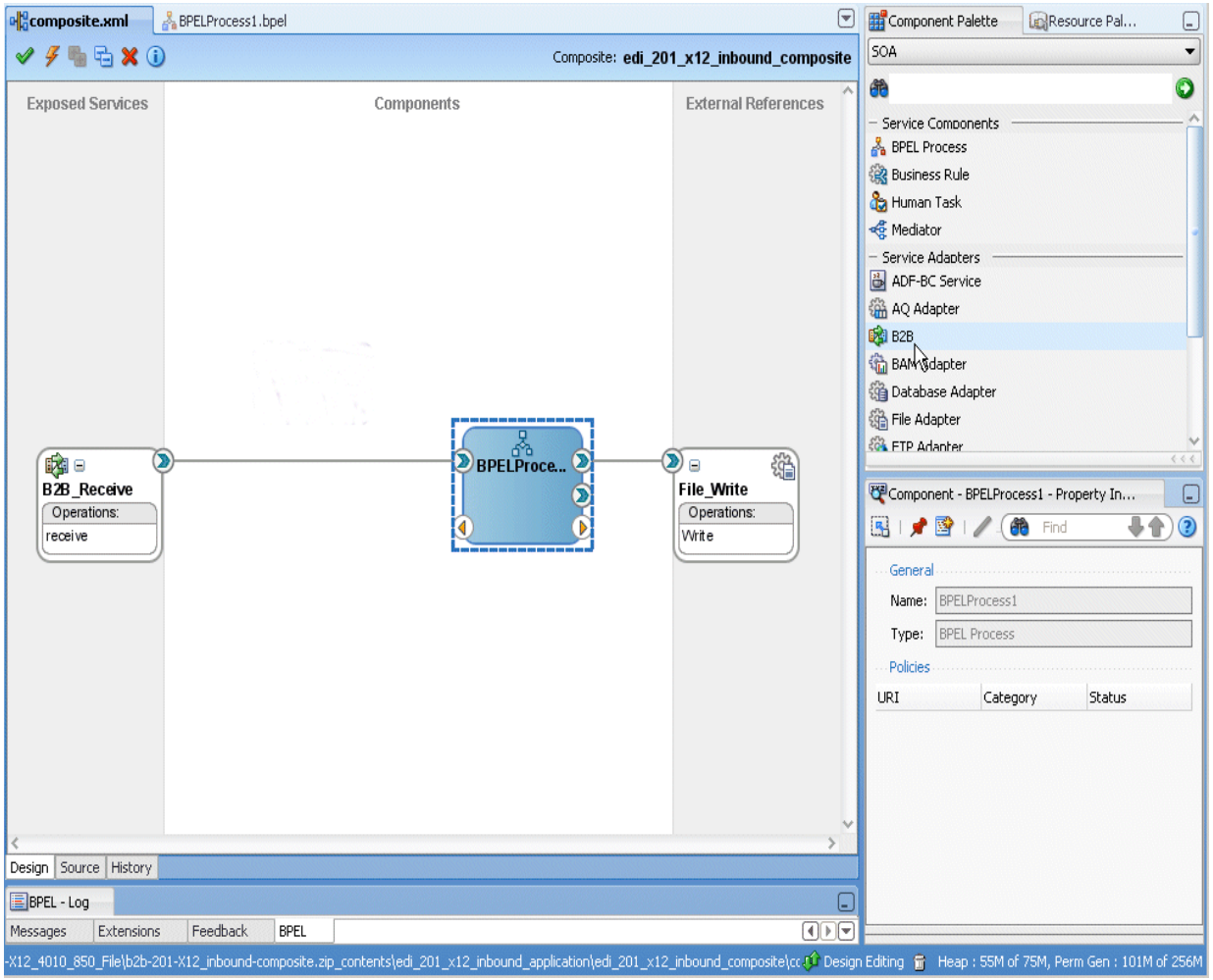


Using Oracle B2B in the Oracle JDeveloper Environment

To include Oracle B2B in an SOA composite application, use the B2B Configuration Wizard as follows:

To access the B2B binding component:

1. In Oracle JDeveloper, open the composite for which you want to add a B2B binding component.
2. From the **SOA** > **Service Adapters** panel of the **Component Palette**, drag **B2B** to one of the following:
   - To the **Exposed Services** area when 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.
   - The **External References** area when B2B is used as a *reference* (outbound) to send messages from the SOA composite application to partners.



3. Follow the steps in the B2B Configuration Wizard of Oracle JDeveloper.

   Select a document definition that was previously created in Oracle B2B. Or, you can launch Oracle B2B from the wizard to create a document definition.

Note: The JMS option on the B2B Integration Type page of the B2B Configuration Wizard is in preview mode. If you want to use this option to communicate with Oracle B2B through JMS queues, start Oracle JDeveloper in preview mode as follows:

```
ORACLE_JDEV_HOME\jdeveloper\jdeveloper -J"-DPREVIEW_MODE=true"
```

What You May Need To Know About Using Oracle B2B

This section contains topics to help with troubleshooting.

Enabling Debug Mode at Run Time

Use Oracle Enterprise Manager 11*g* 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*

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.

See *Oracle Fusion Middleware Security Guide*for information about Oracle Single Sign-On and Oracle Access Manager.

Part II

Oracle B2B Process Flow

This part describes the Oracle B2B process flow.

This part contains the following chapters:

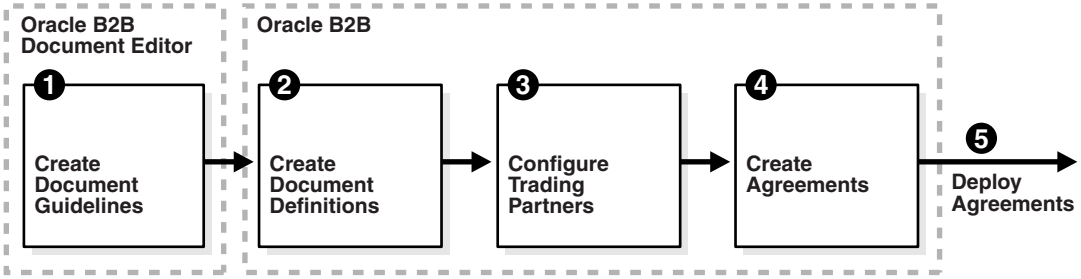
- Chapter 3, "Creating Guideline Files"
- Chapter 4, "Creating Document Definitions"
- Chapter 5, "Configuring Trading Partners"
- Chapter 6, "Creating and Deploying Trading Partner Agreements"

3

Creating Guideline Files

The first step in the Oracle B2B process flow, shown in Figure 3–1, is to create 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:

- Introduction to Oracle B2B Document Editor
- Installing Oracle B2B Document Editor
- Creating Guideline Files: EDIFACT D98 Example

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

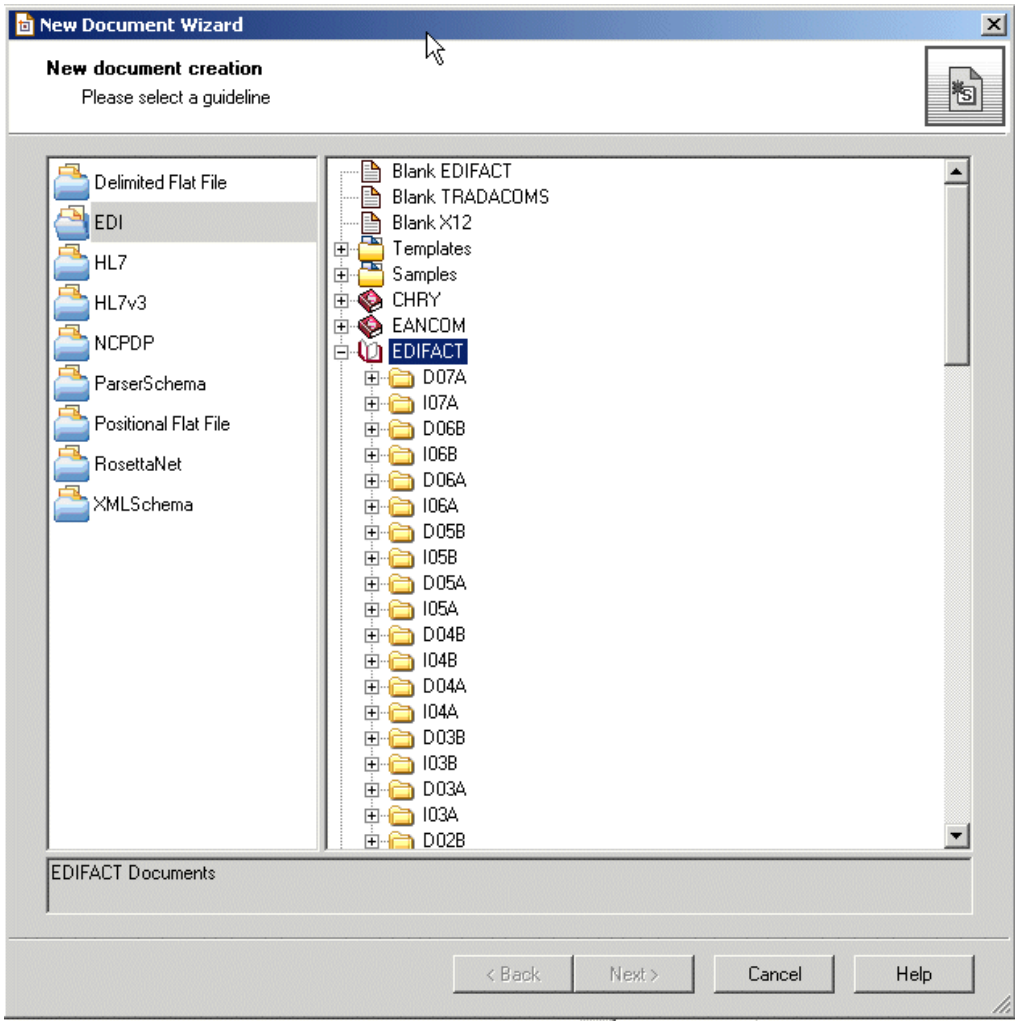
Introduction to Oracle B2B Document Editor

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.

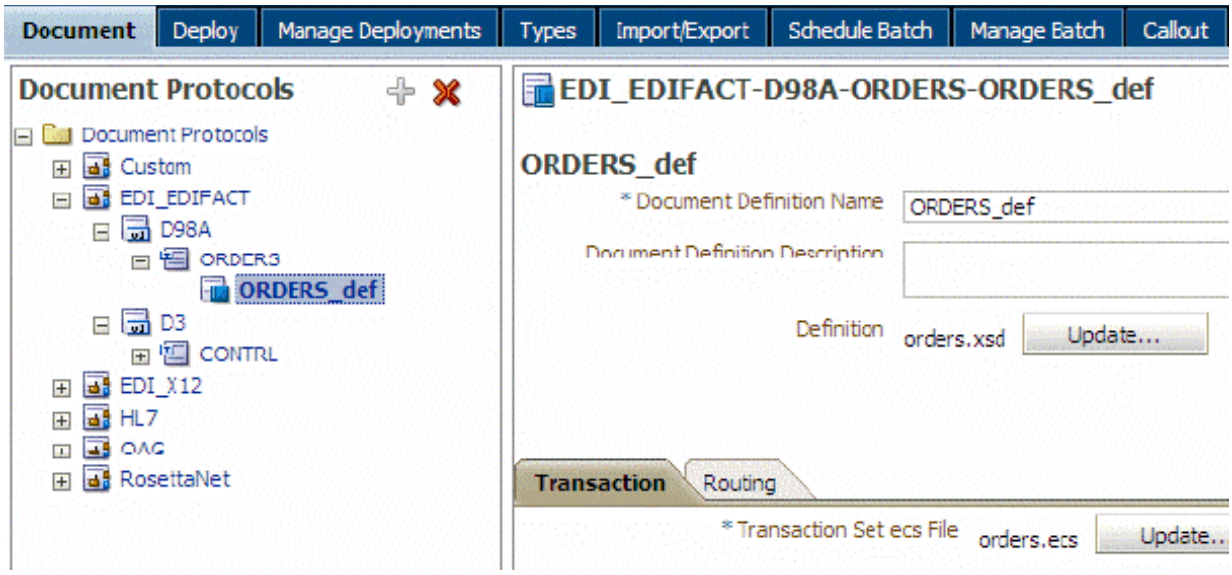
Figure 3–2 Document Guidelines Available in Oracle B2B Document Editor



In addition to using the RosettaNet document guide lines 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 "Creating Document Definitions" on page 4-3 for more information about this step.

Figure 3–3 Importing XSD and ECS File Created in Oracle B2B Document Editor



In Figure 3–3, `orders.xsd` and `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.

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.

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:

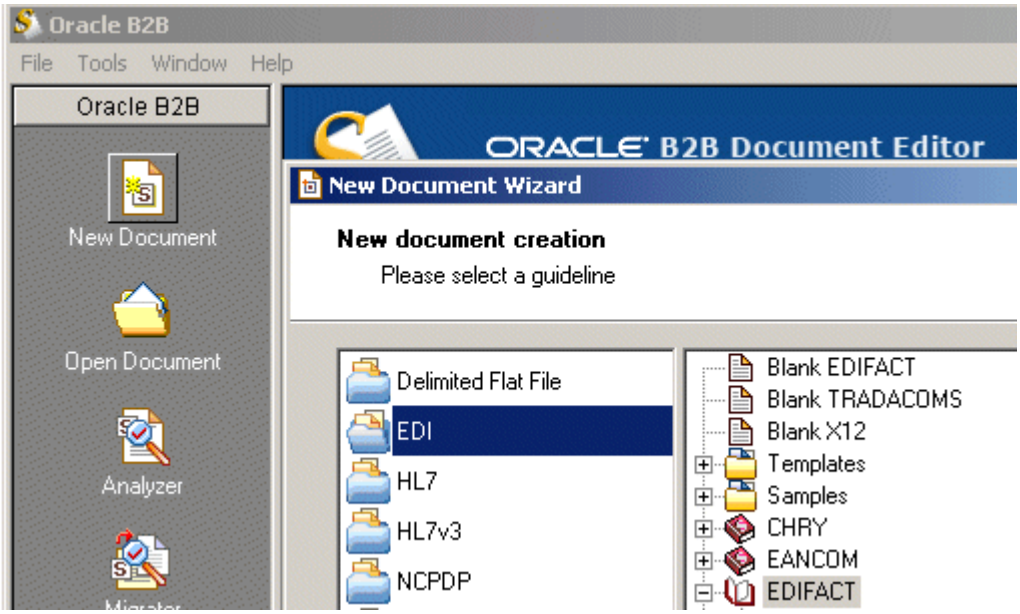
- Task 1, "Create the ECS File"
- Task 2, "Create the XSD File"
- Task 3, "Generate Data Using the ECS File"
- Task 4, "Analyze the Data"

[1] When using Microsoft Vista, do not install Oracle B2B Document Editor in the program folder, for which admin privilege is needed.

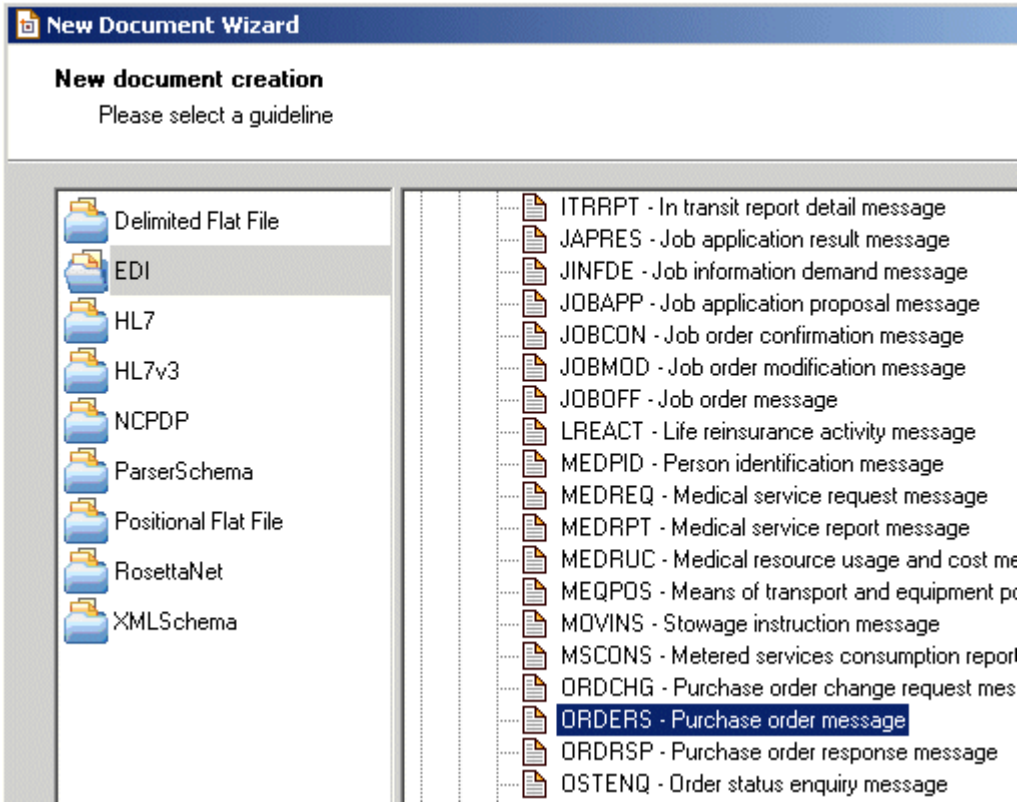
Task 1 Create the ECS File

Using an existing EDIFACT guideline (standard) as a template, create a purchase order guideline file called **orders.ecs**.

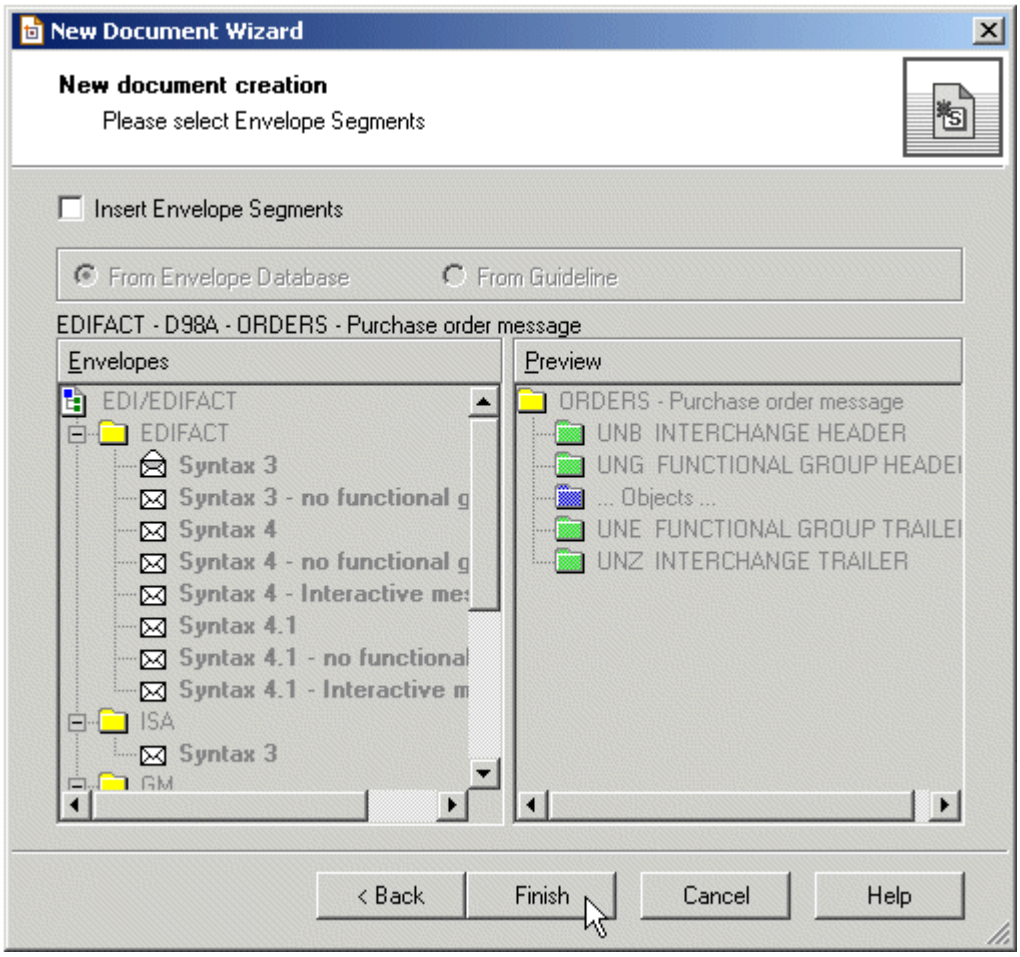
1. Open Oracle B2B Document Editor.
2. Click **New Document** and then **EDI**.



3. Expand **EDIFACT** and **D98A**.
4. Select **ORDERS - Purchase order message** and click **Next**.



5. Ensure that **Insert Envelope Segments** is *not* selected and click **Finish**.

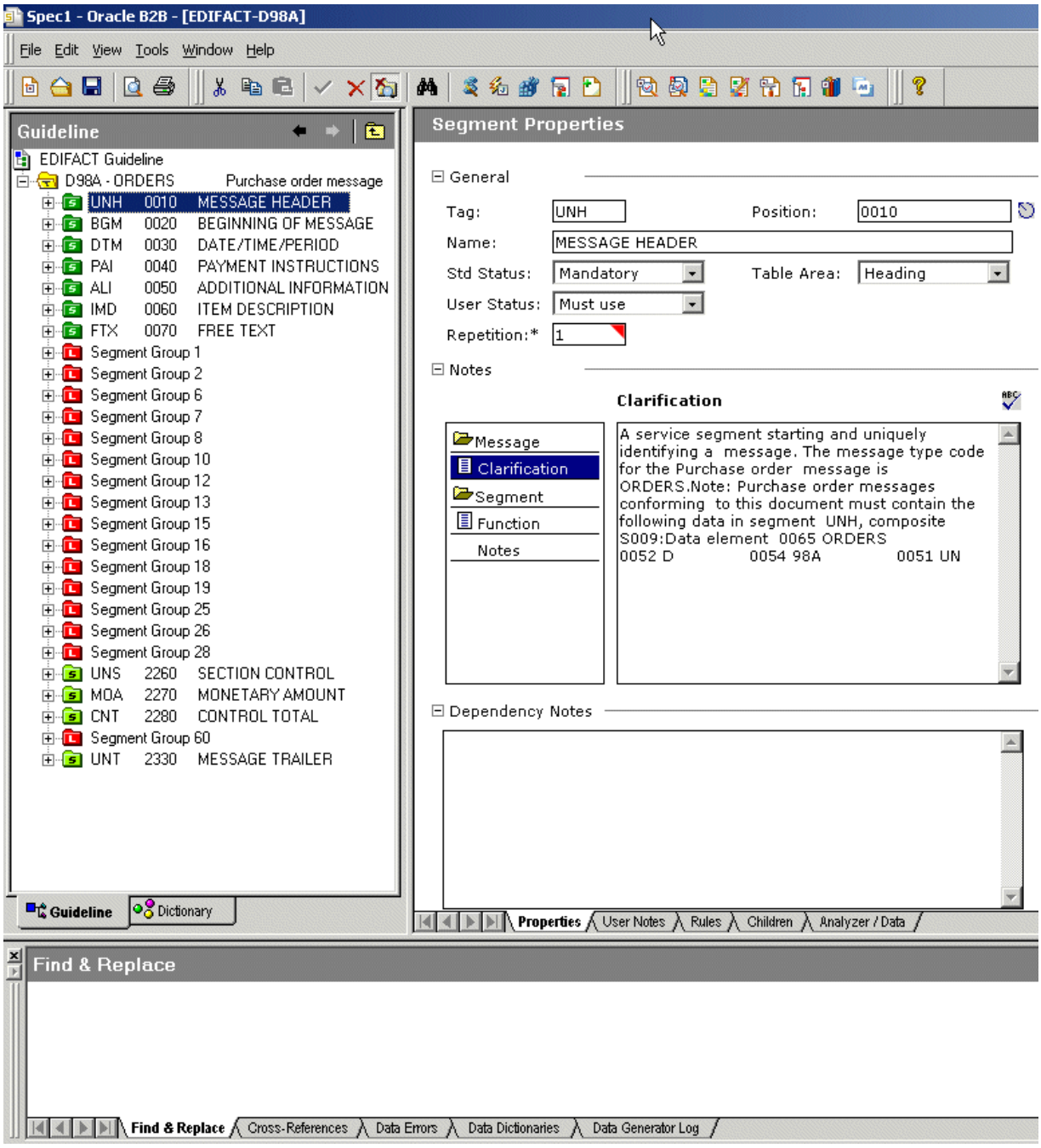


Oracle B2B Document Editor is preseeded with all versions of the interchange (envelope). Oracle B2B handles the envelop based on the settings.

Select this option only if you require a variation from the standard (for example, if you want to use a nonstandard qualifier for the partner identification code qualifier in the interchange sender or recipient, then add a required value in the codelist).

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

   No edits are needed for this scenario.



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**.

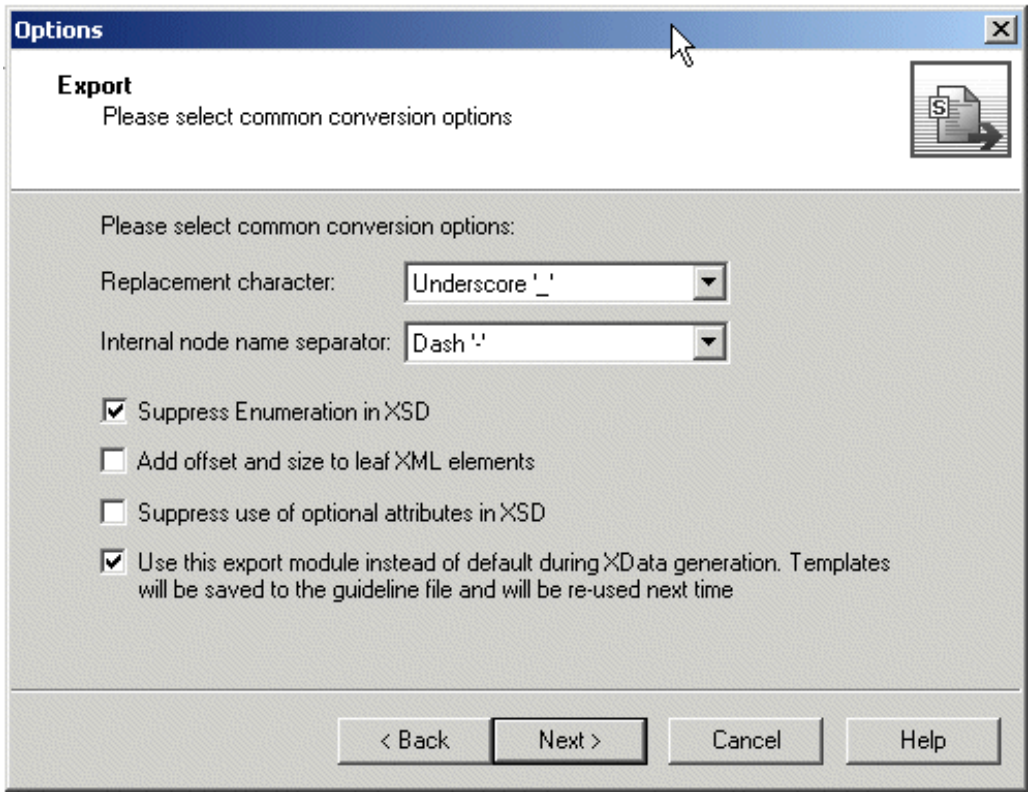
2. Select `orders.ecs` and click **Open**.
3. From **File**, select **Export**.
4. In the Export Wizard, select **Oracle B2B 2.0** from the list of export types and click **Next**.

   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.)

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, do the following and click **Next**.



- 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.

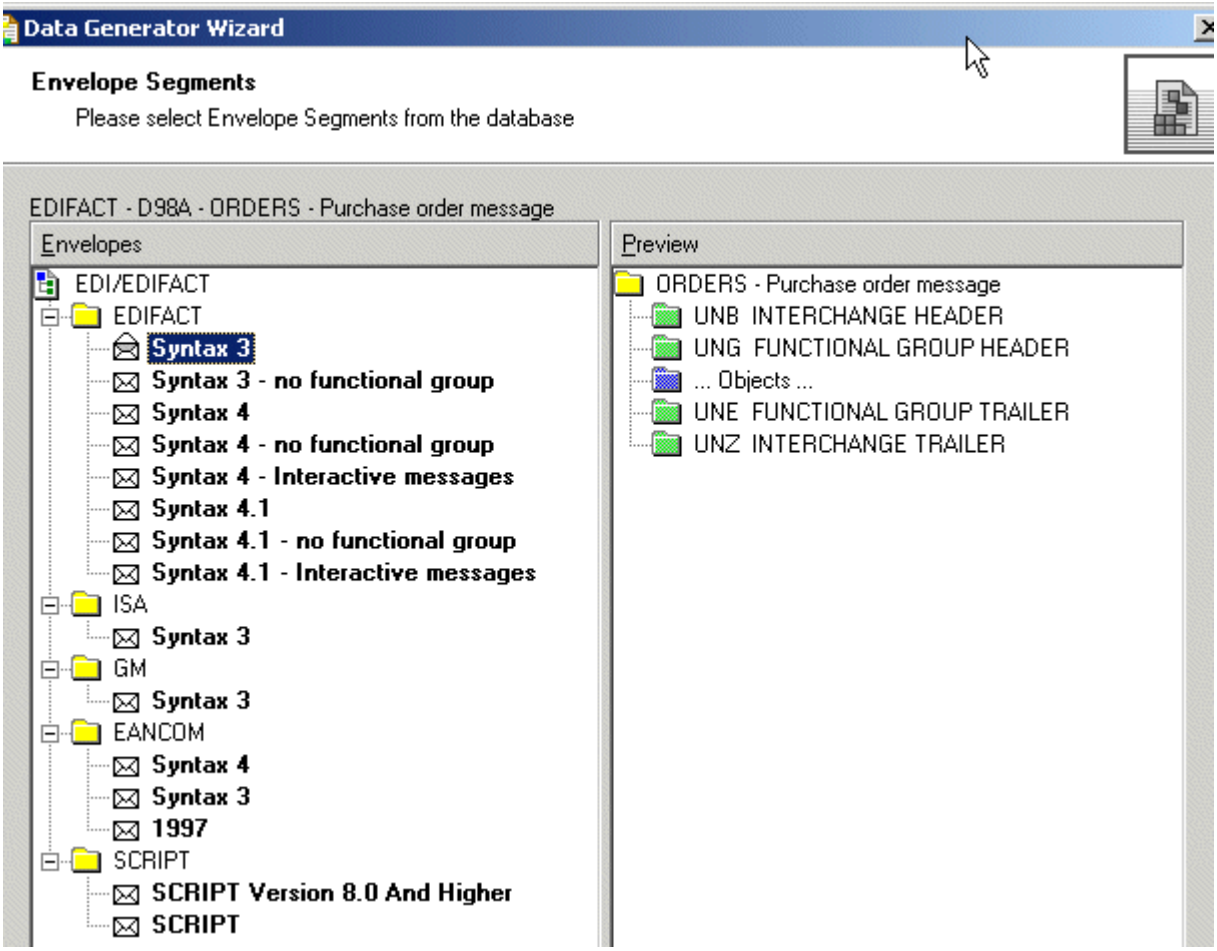
Task 3 Generate Data Using the ECS File

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 and click **Next**.

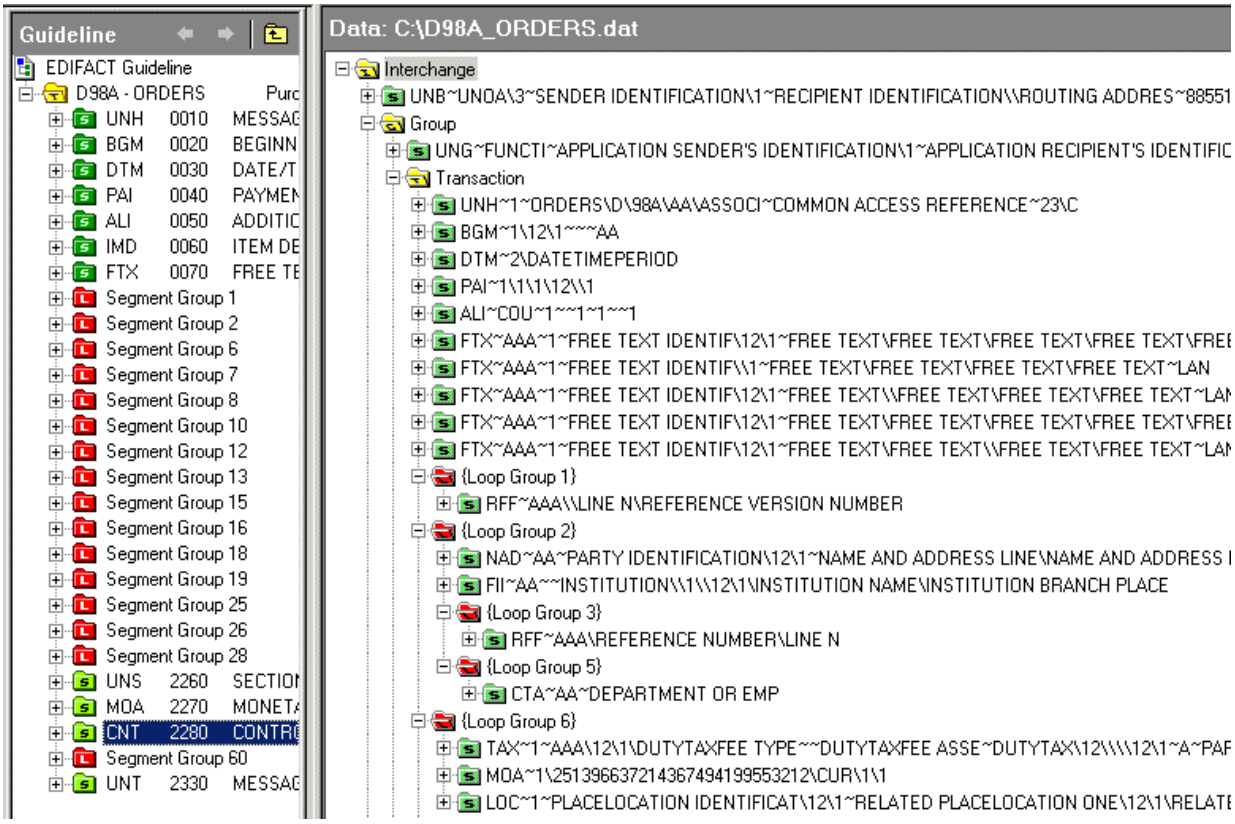


7. Select **Use directly from the Standards Database** and click **Next**.

   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.

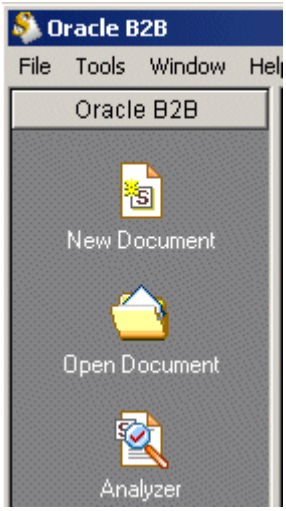


16. Save and close the file.

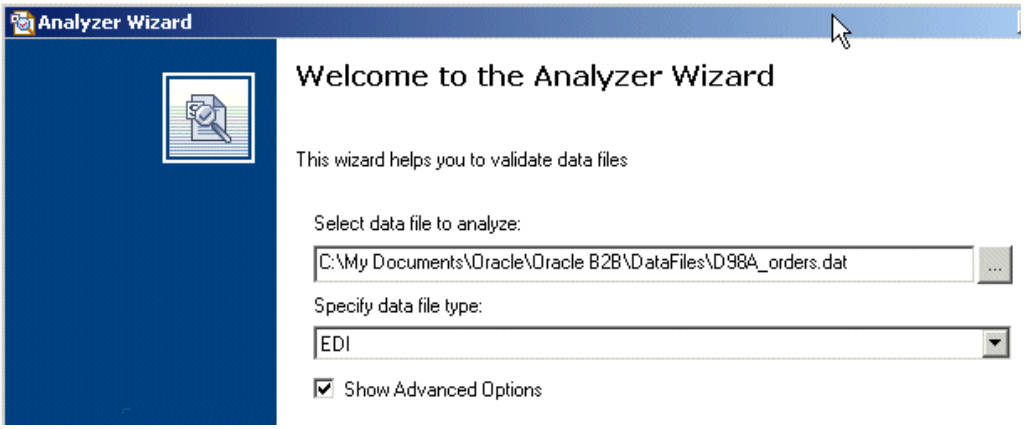
Task 4 Analyze the Data

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**.



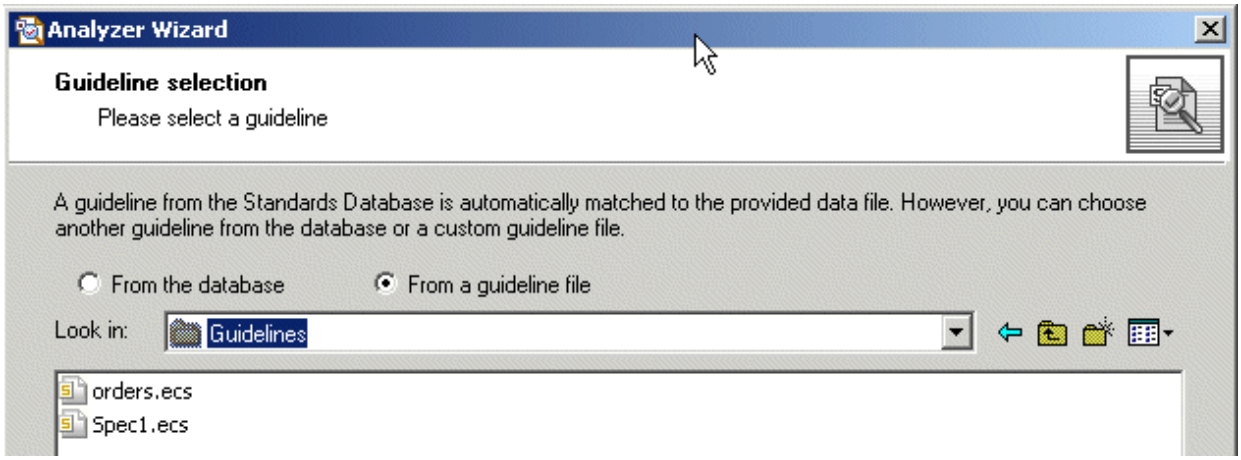
2. Browse for `D98A_orders.dat` and click **Next.**
3. Ensure that **Show Advanced Options** is selected and click **Next**.



4. In the Clean Up Data File dialog, click **Next**.

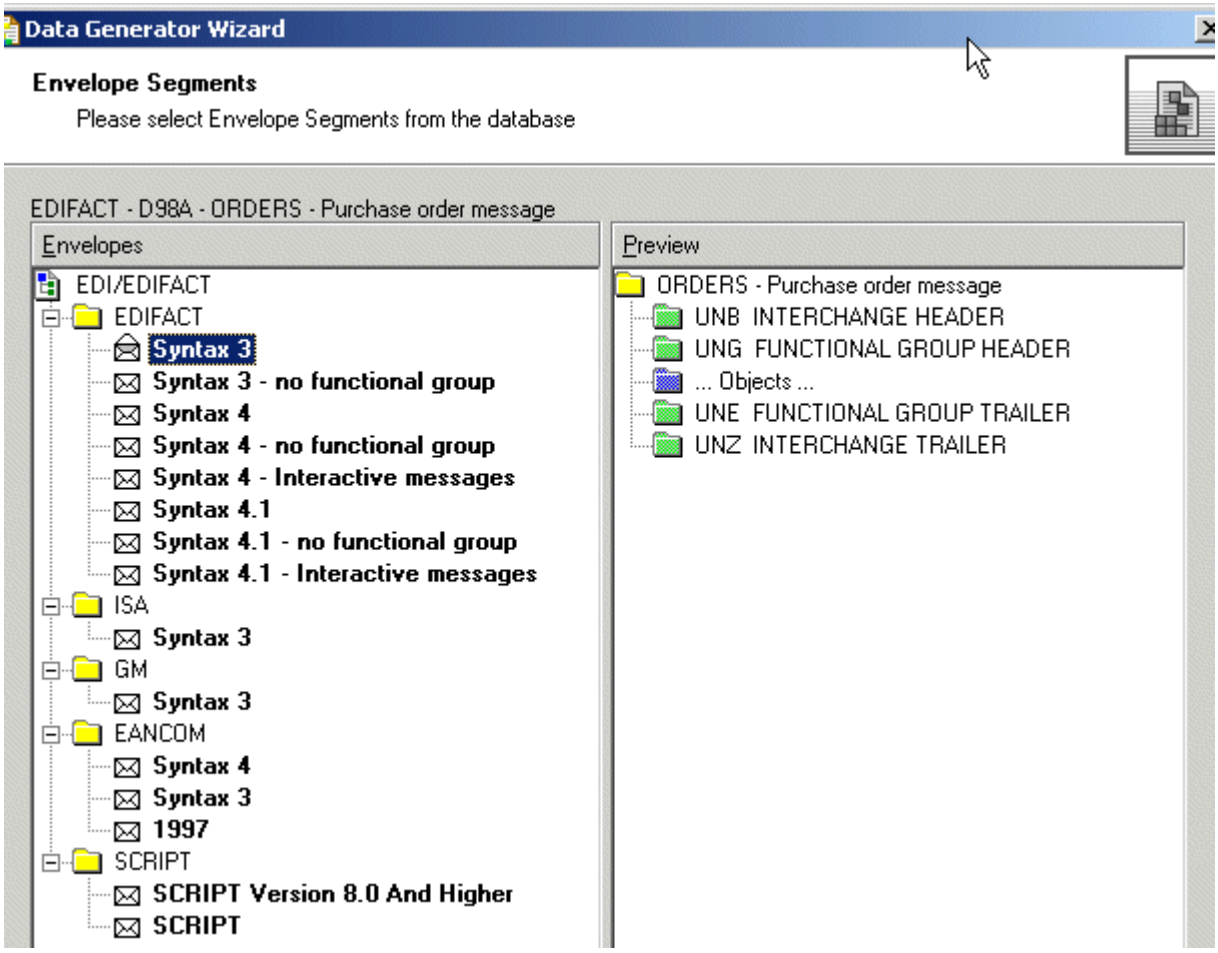
   No preprocessing is needed in this scenario.
5. In the Data Structure dialog, click **Next**.

   The entire document is validated by default.
6. Select the guideline file (ECS file) against which to check the data. 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 and click **Next**.

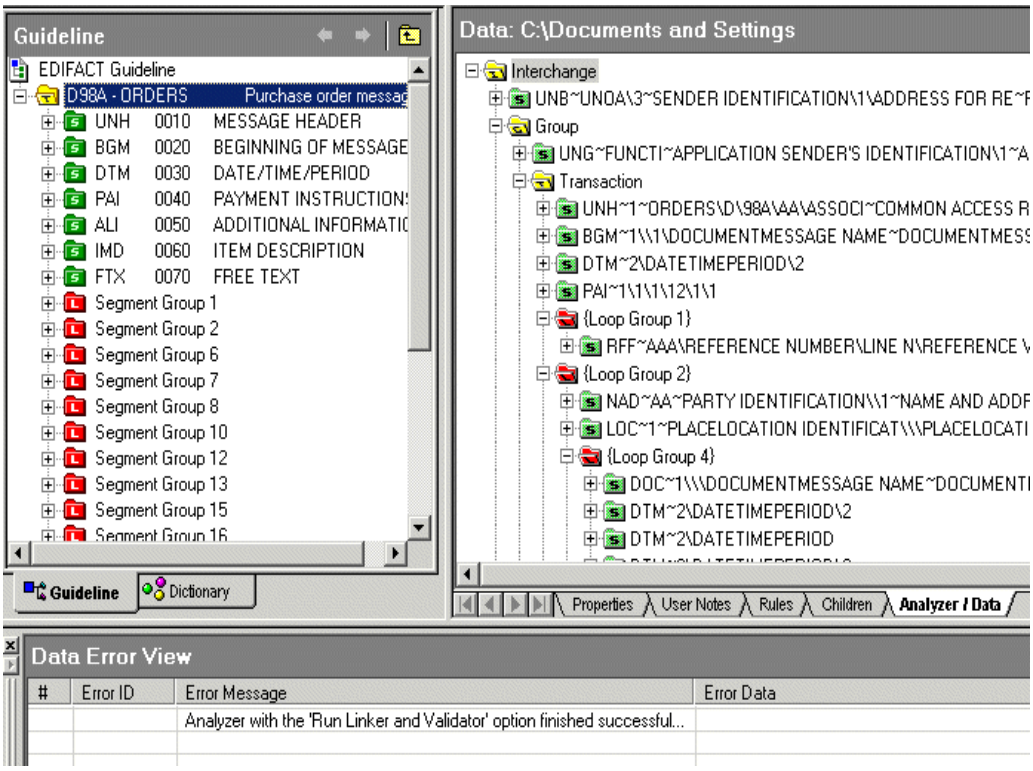


9. Select **Use directly from the Standards Database** and click **Next**.

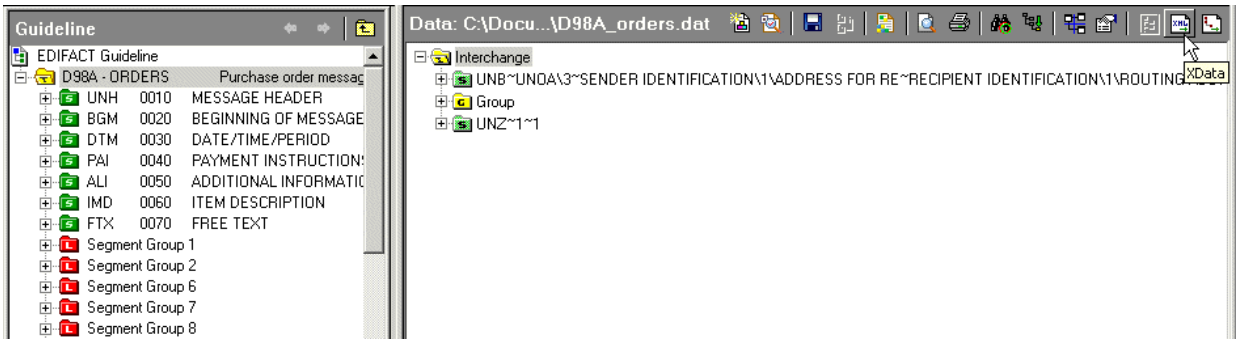
   The envelope segments are not incorporated in the guideline file.

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

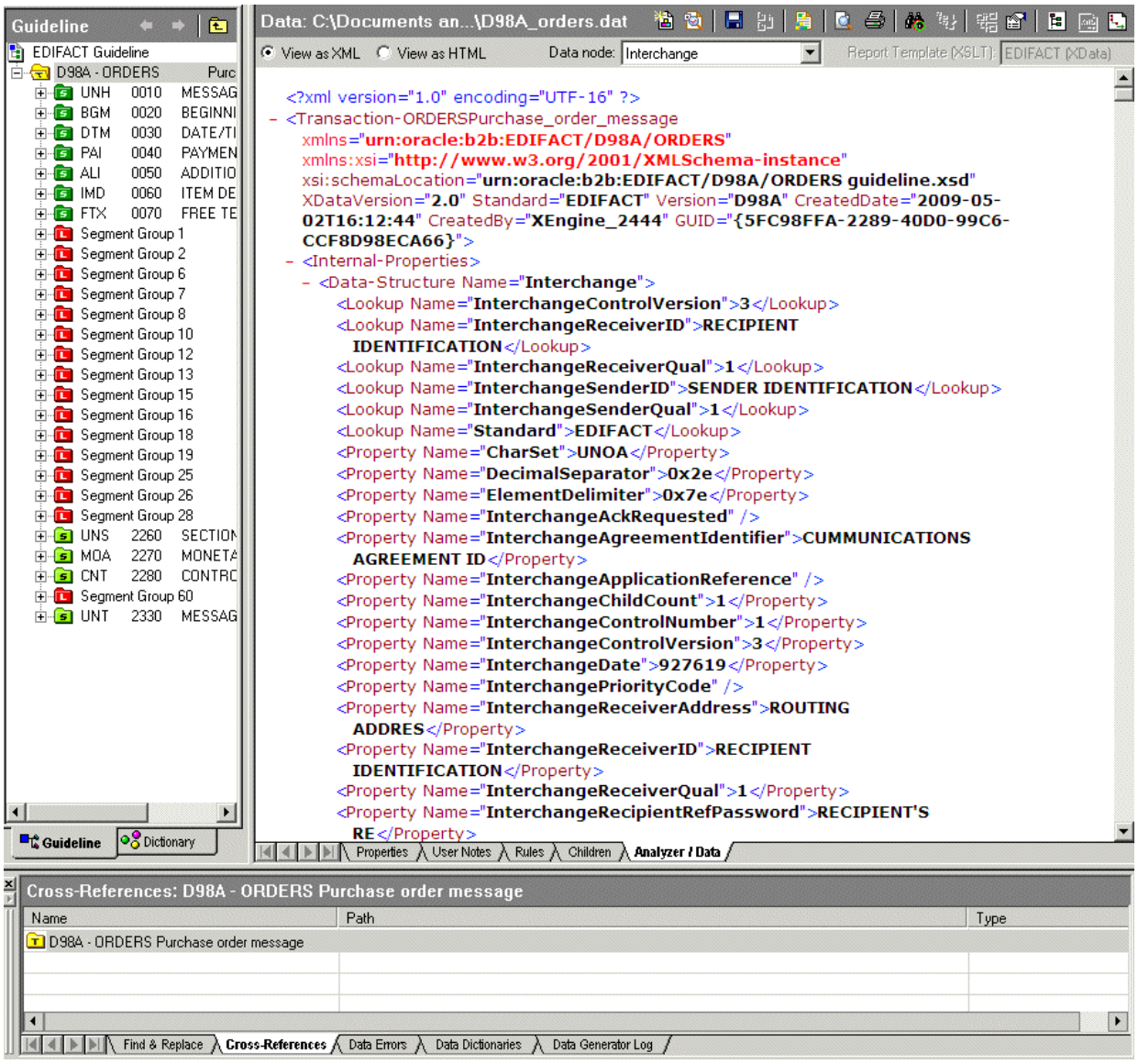
    The results, including any error messages, are displayed.



To view the data in XML format, click the **XML** icon in the upper right corner.



Use the **View as XML** (shown) and **View as HTML** options to view the data. Click the **Save Data As** icon to export the XML report as an XML file.

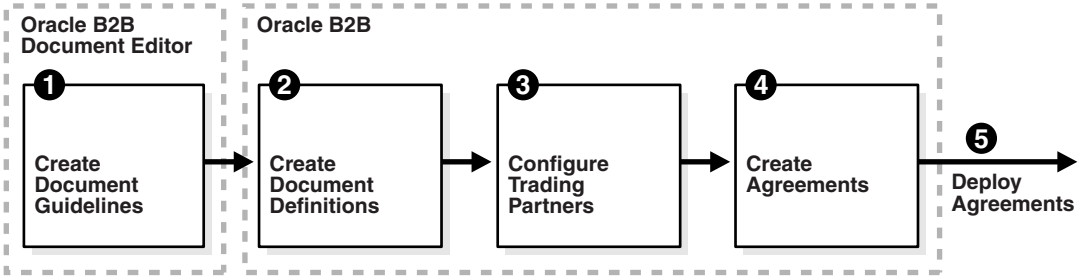

Guideline
EDIFACT Guideline
D98A - ORDERS
UNH 0010 MESSAG
BGM 0020 BEGINNI
DTM 0030 DATE/TI
PAI 0040 PAYMEN
ALI 0050 ADDITIO
IMD 0060 ITEM DE
FTX 0070 FREE TE
Segment Group 1
Segment Group 2
Segment Group 6
Segment Group 7
Segment Group 8
Segment Group 10
Segment Group 12
Segment Group 13
Segment Group 15
Segment Group 16
Segment Group 18
Segment Group 19
Segment Group 25
Segment Group 26
Segment Group 28
UNS 2260 SECTION
MOA 2270 MONETA
CNT 2280 CONTRC
Segment Group 60
UNT 2330 MESSAG
Guideline
Dictionary
Data: C:\Documents an...\D98A_orders.dat
View as XML
View as HTML
Data node: Interchange
Report Template (XSLT): EDIFACT (XData)
<?xml version="1.0" encoding="UTF-16" ?>
<Transaction-ORDERSPurchase_order_message
xmlns="urn:oracle:b2b:EDIFACT/D98A/ORDERS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oracle:b2b:EDIFACT/D98A/ORDERS guideline.xsd"
XDataVersion="2.0" Standard="EDIFACT" Version="D98A" CreatedDate="2009-05-02T16:12:44" CreatedBy="XEngine_2444" GUID="{5FC98FFA-2289-40D0-99C6-CCF8D98ECA66}">
<Internal-Properties>
<Data-Structure Name="Interchange">
<Lookup Name="InterchangeControlVersion">3</Lookup>
<Lookup Name="InterchangeReceiverID">RECIPIENT IDENTIFICATION</Lookup>
<Lookup Name="InterchangeReceiverQual">1</Lookup>
<Lookup Name="InterchangeSenderID">SENDER IDENTIFICATION</Lookup>
<Lookup Name="InterchangeSenderQual">1</Lookup>
<Lookup Name="Standard">EDIFACT</Lookup>
<Property Name="CharSet">UNOA</Property>
<Property Name="DecimalSeparator">0x2e</Property>
<Property Name="ElementDelimiter">0x7e</Property>
<Property Name="InterchangeAckRequested" />
<Property Name="InterchangeAgreementIdentifier">CUMMUNICATIONS AGREEMENT ID</Property>
<Property Name="InterchangeApplicationReference" />
<Property Name="InterchangeChildCount">1</Property>
<Property Name="InterchangeControlNumber">1</Property>
<Property Name="InterchangeControlVersion">3</Property>
<Property Name="InterchangeDate">927619</Property>
<Property Name="InterchangePriorityCode" />
<Property Name="InterchangeReceiverAddress">ROUTING ADDRES</Property>
<Property Name="InterchangeReceiverID">RECIPIENT IDENTIFICATION</Property>
<Property Name="InterchangeReceiverQual">1</Property>
<Property Name="InterchangeRecipientRefPassword">RECIPIENT'S RE</Property>
Properties
User Notes
Rules
Children
Analyzer / Data
Cross-References: D98A - ORDERS Purchase order message
Name
Path
Type
D98A - ORDERS Purchase order message
Find & Replace
Cross-References
Data Errors
Data Dictionaries
Data Generator Log

4

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 straightfoward 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).

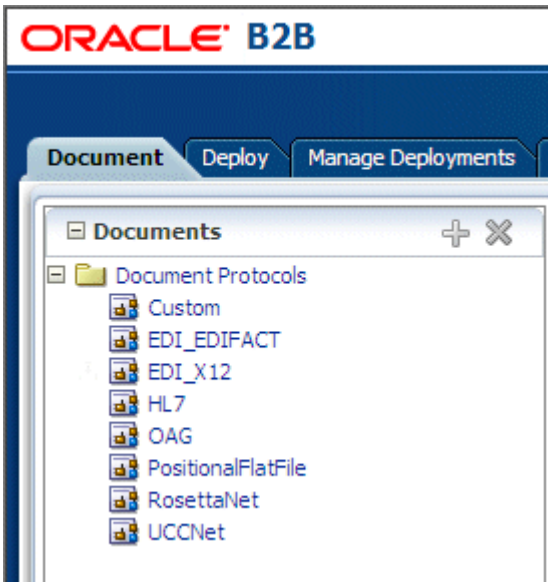
This chapter contains the following topics:

- Introduction to Document Protocols
- Creating Document Definitions
- Deleting a Document Definition

Introduction to Document Protocols

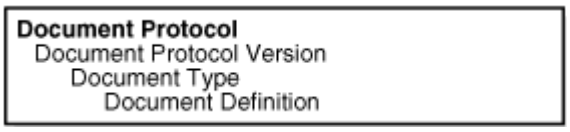
Figure 4–2 shows the document protocols supported in Oracle B2B. Using the Custom protocol and the many guideline documents in Oracle B2B Document Editor, you can define most protocols. When you add a new document protocol, it is always a Custom document.

Figure 4–2 Oracle B2B Document Protocols



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

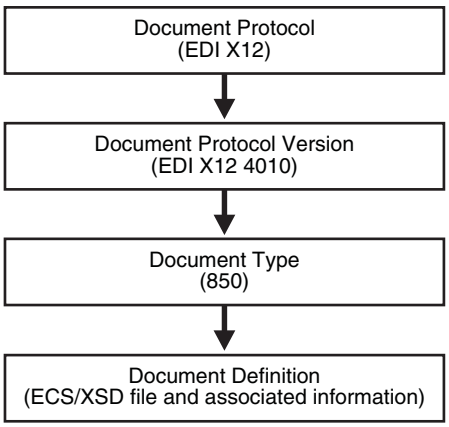
Figure 4–3 Document Hierarchy



A document protocol can consist of multiple document protocol versions. A document protocol version can consist of multiple document types. A document type can consist of multiple document definitions. Typically, you start with one document definition and customize it for different trading partners.

Figure 4–4 shows a document protocol hierarchy as it applies to EDI X12.

Figure 4–4 EDI X12 Document Hierarchy



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

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.

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

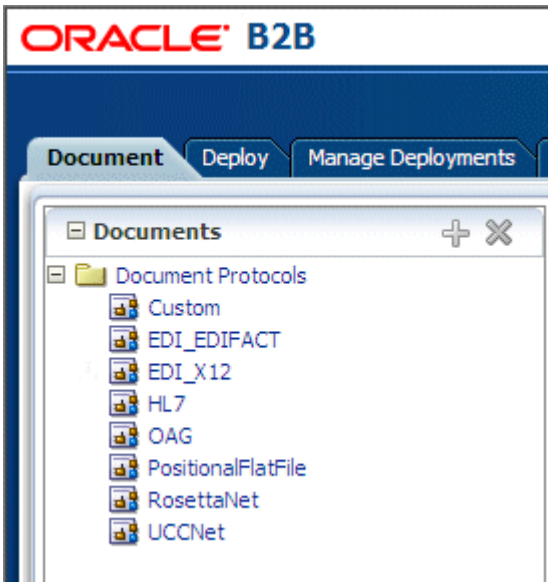
Creating Document Definitions

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.

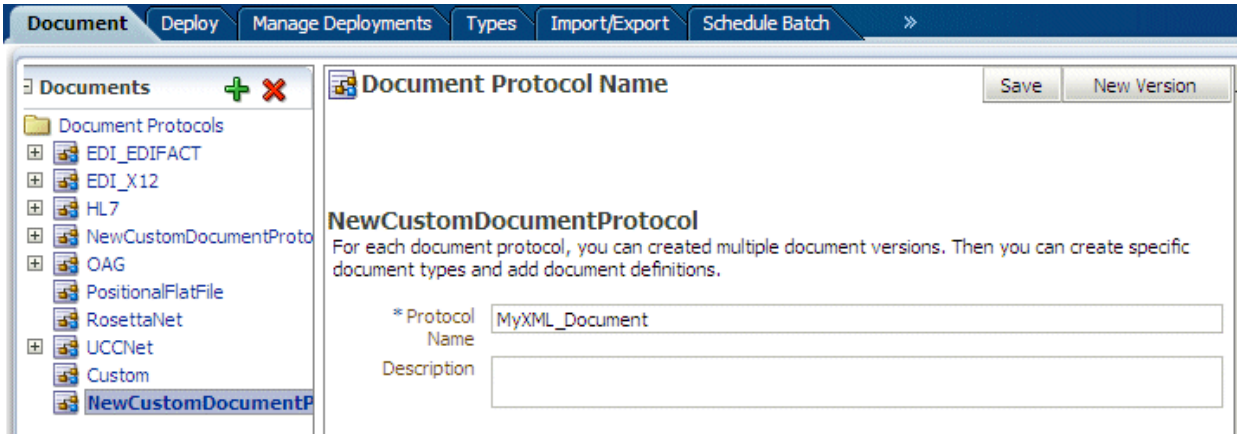
> **Note:** The document version, document type, and document definition are not editable after they are created. You must delete the specific document element (version, type, or definition) and create a new one. Updating the document elements after creation can lead to metadata inconsistency, metadata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

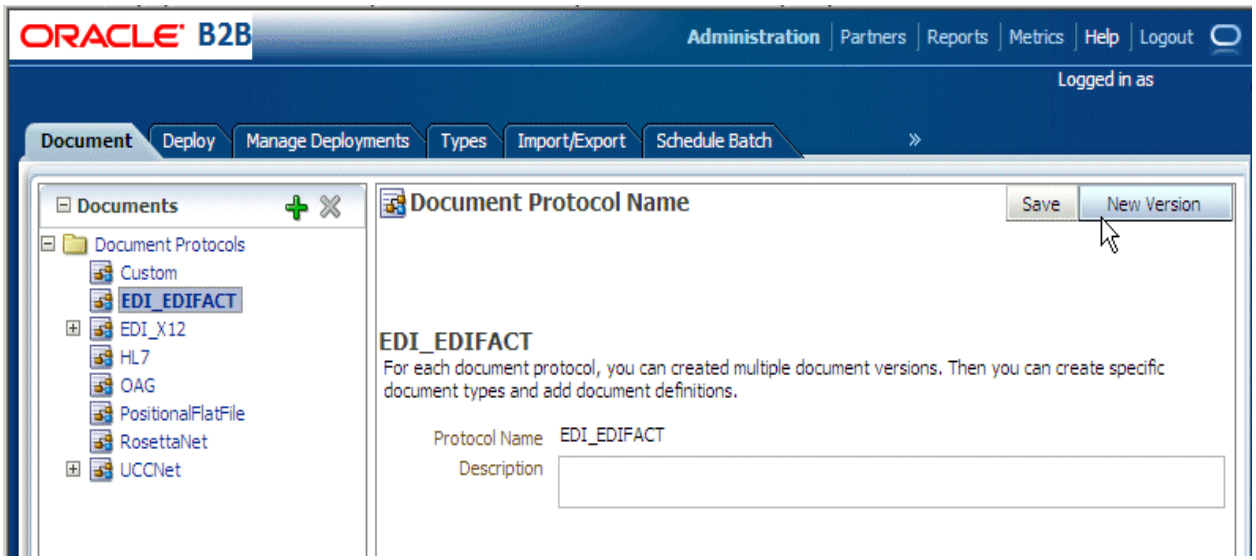


To create a new Custom document with a name that you provide, for example, MyXML_Document, click the **Document Protocols** folder, click **Add**, and provide a protocol name. Do not use an existing document protocol name.



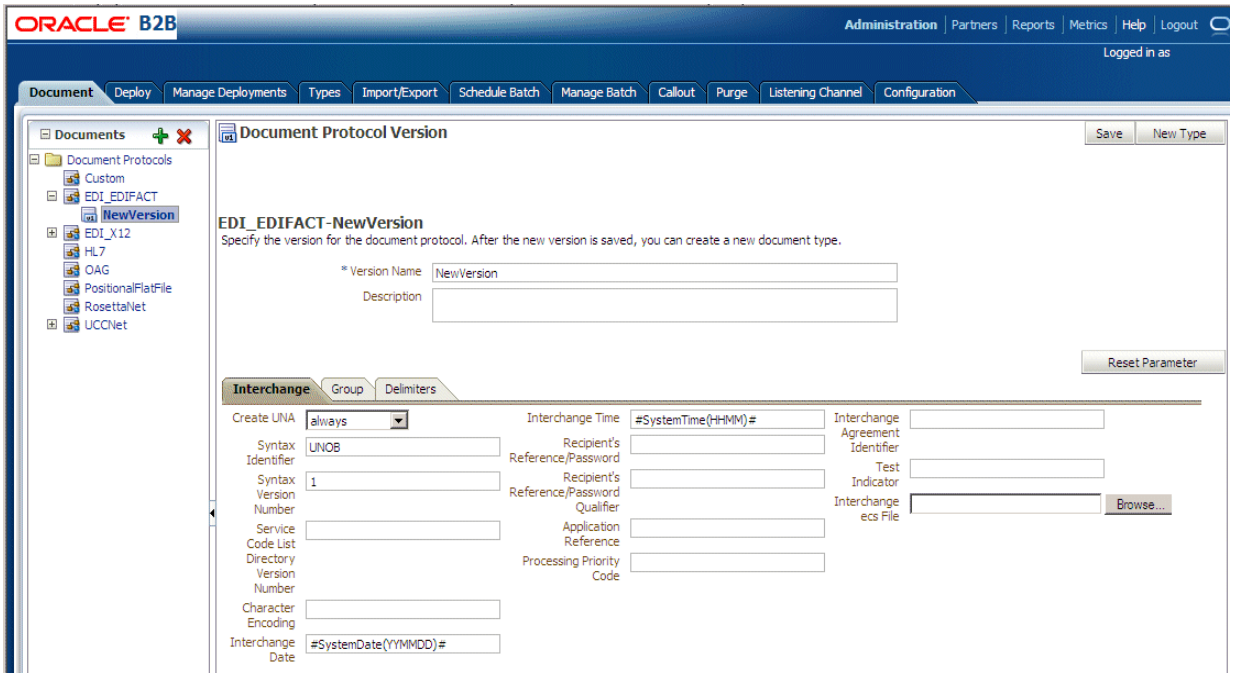
4. Click **New Version**.

   (An EDI EDIFACT document is shown 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.

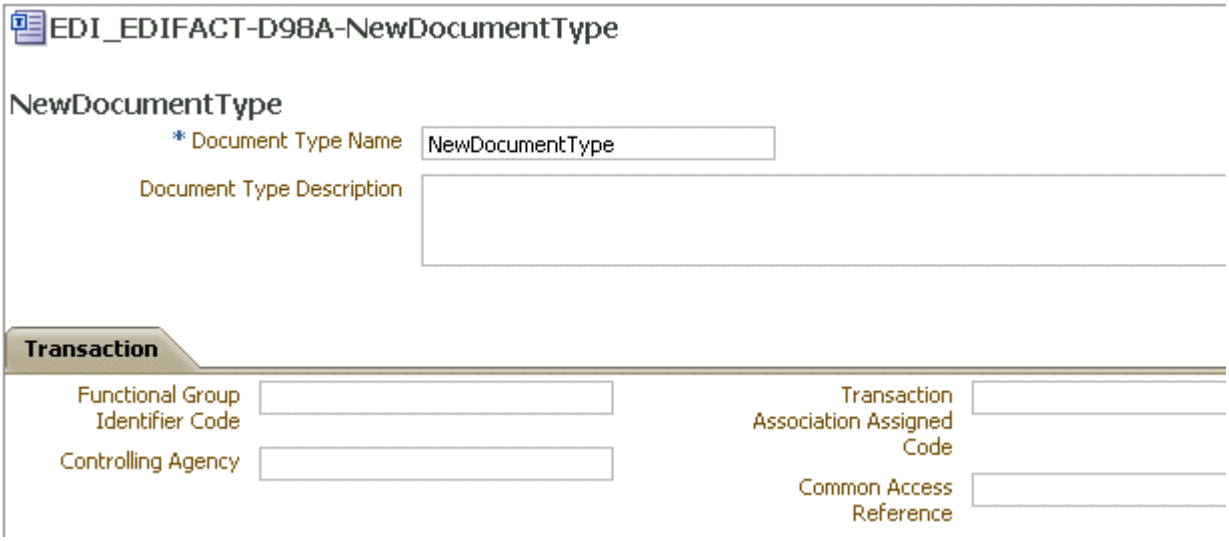


For parameter descriptions, see the following:

- Table 7–4, " Document Version Parameters for an EDI EDIFACT Document" on page 7-8
- Table 7–8, " Document Version Parameters for an EDI X12 Document" on page 7-13
- Table 7–11, " Document Version Parameters for an HL7 Document" on page 7-17

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**.

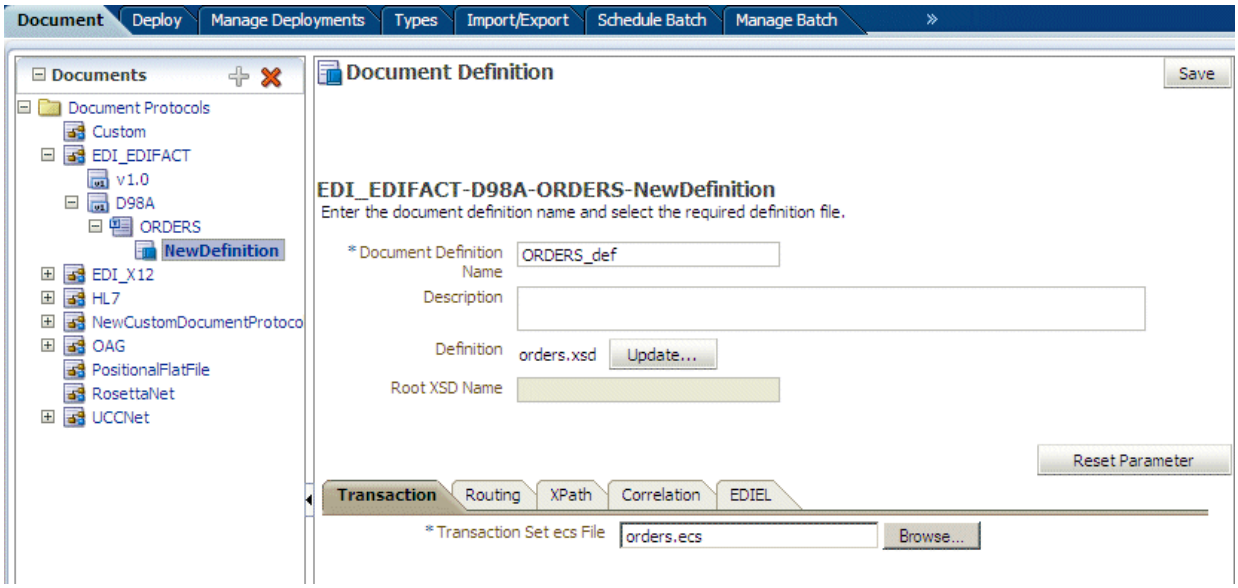
   The version is used for document identification and can be case sensitive and use a fixed syntax, depending on the protocol.



For parameter descriptions, see the following:

- Table 7–1, " Document Type Parameters for a Custom Document" on page 7-3
- Table 7–5, " Document Type Parameters for an EDI EDIFACT Document" on page 7-10.
- Table 7–9, " Document Type Parameters for an EDI X12 Document" on page 7-14
- Table 7–12, " Document Type Parameters for an HL7 Document" on page 7-19
- Table 7–17, " Document Type Parameters for a RosettaNet Document" on page 7-26

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 type parameters as applicable.



For parameter descriptions, see the following:

- Table 7–2, " Document Definition Parameters for a Custom Document" on page 7-4
- Table 7–6, " Document Definition Parameters for an EDI EDIFACT Document" on page 7-11
- Table 7–10, " Document Definition Parameters for an EDI X12 Document" on page 7-15.
- Table 7–13, " Document Definition Parameters for an HL7 Document" on page 7-19
- Table 7–14, " Document Definition Parameters for an OAG Document" on page 7-21
- Table 7–15, " Document Definition Parameters for a Positional Flat File" on page 7-23
- Table 7–18, " Document Definition Parameters for a RosettaNet Document" on page 7-27
- Table 7–20, " Document Definition Parameters for a UCCnet Document" on page 7-30

10. Click **Save**.

Deleting a Document Definition

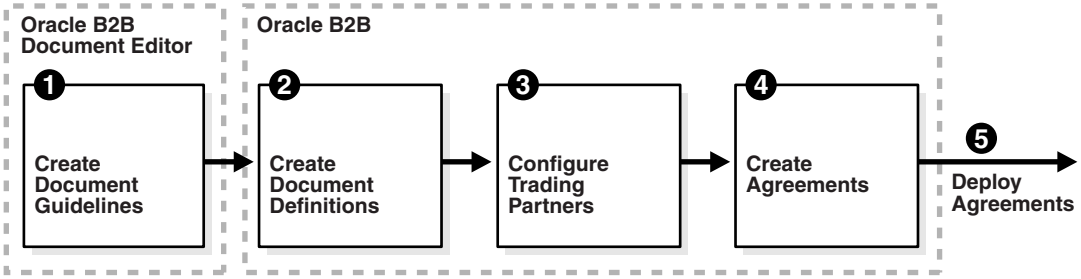
To delete a document definition, first delete all agreements that use that document definition and then remove the supported document definitions from the host and all remote trading partners that reference the definition.

5

Configuring Trading Partners

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:

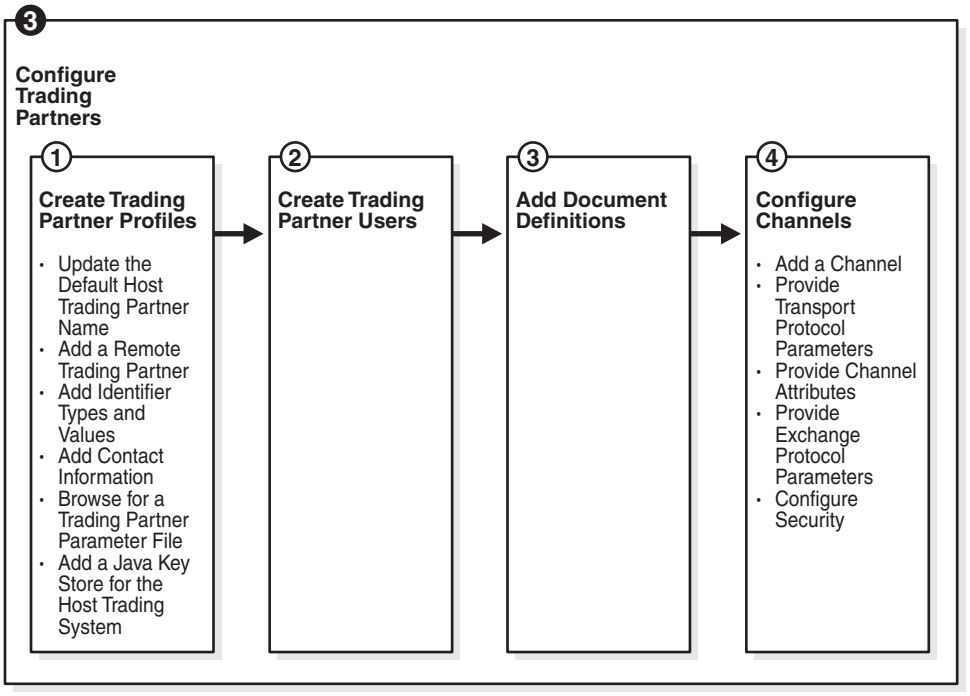
- Introduction to Trading Partners
- Creating Trading Partner Profiles
- Adding Trading Partner Users
- Adding Document Definitions
- Configuring Channels
- Using the Auto Create Agreement Feature
- Using Identifiers for Trading Partner Lookup

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 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



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.

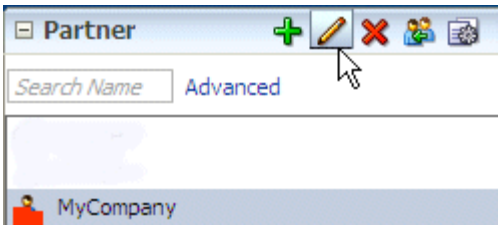
To create a trading partner profile, do the following:

- 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**.



4. Provide the host trading partner name and optional icon file, and click **OK**.

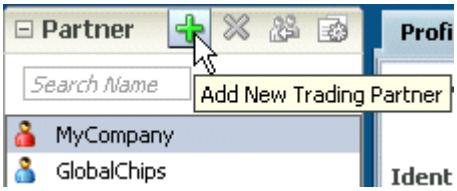
   The optional icon file must be a 16 x 16-pixel PNG file.

   The host trading partner name appears in the **Partner** list.

Task 2 Add a Remote Trading Partner

Do this for each remote trading partner.

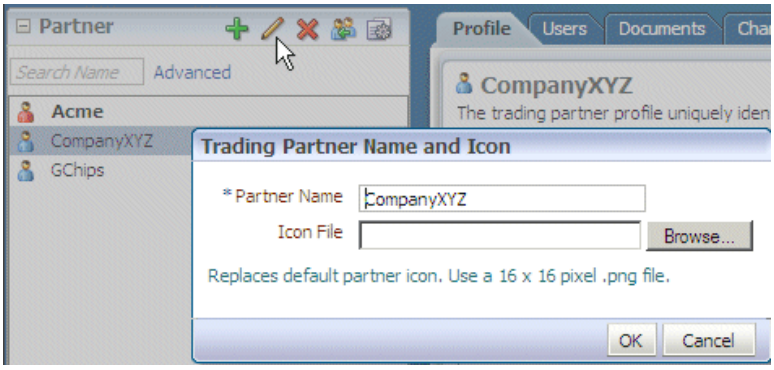
1. Click the **Partners** link.
2. Click **Add**.



3. Provide a partner name and click **OK**.

   The remote trading partner name appears in the **Partner** list.

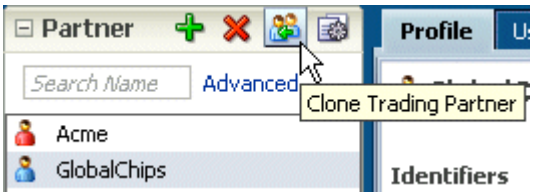
4. (Optional) Click **Edit** to add a 16 x16-pixel PNG file as an icon for the remote trading partner, and click **OK**.



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 5–3, 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 5–3 Cloning a Remote Trading Partner



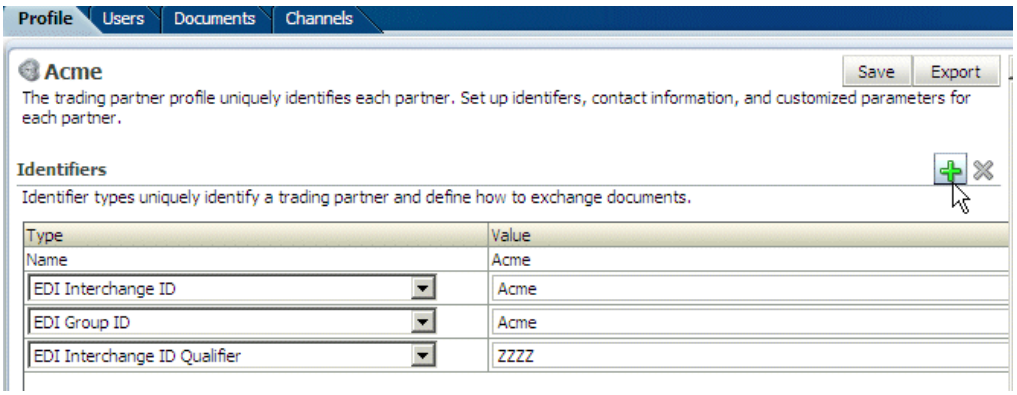
Note: Use the **Delete** icon to delete a remote trading partner. However, you cannot delete a remote trading partner that is part of a deployed trading partner agreement. You must first delete the agreement.

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 more information.

1. Click the **Partners** link.
2. Click the **Profile** tab.
3. Select a trading partner.
4. In the **Identifiers** area, click **Add**.



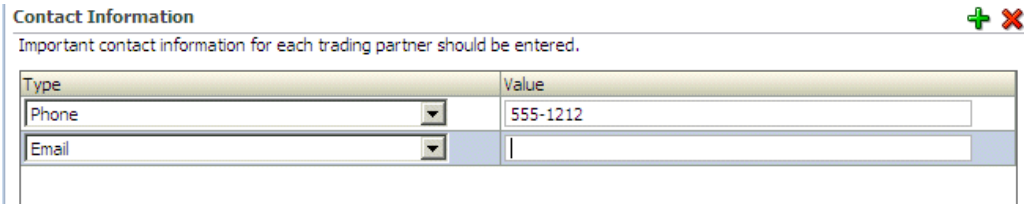
5. From the **Type** list, select an identifier type.

   For descriptions of the identifier types, see Table 9–1, " Identifier Types Defined in Oracle B2B" on page 9-2.
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. Or, you can create the contact type on the **Administration** > **Types** page. See "Creating Custom Contact Information Types" on page 9-4 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.

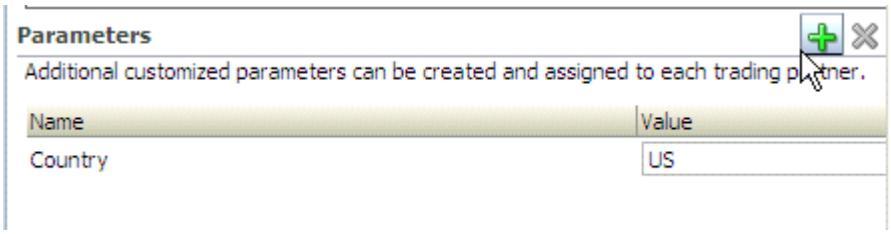


5. Click **Save**.

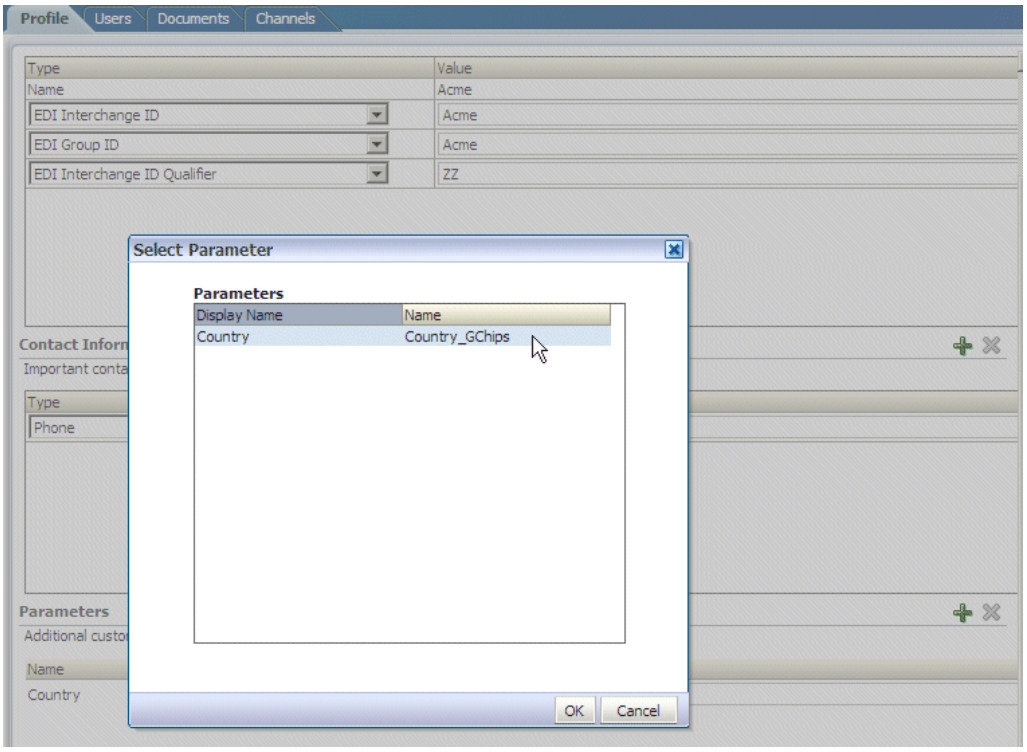
Task 5 Add a Trading Partner Parameter and Value

Before adding an optional trading partner parameter and value for a trading partner, you must create the parameter on the **Administration** > **Types** page. 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**.



4. Select a parameter and click **OK**.



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 5–3, " Channel Details and Associated Protocols" 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.

5. Click **Save**.

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.

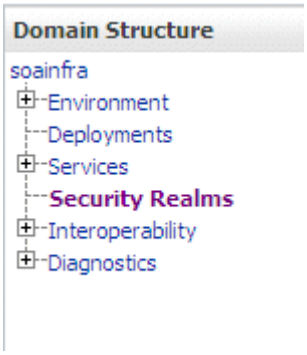
To add users, do the following:

- Task 1, "Create a New User in the Identity Store"
- Task 2, "Add a User in the Oracle B2B Interface"

Task 1 Create a New User in the Identity Store

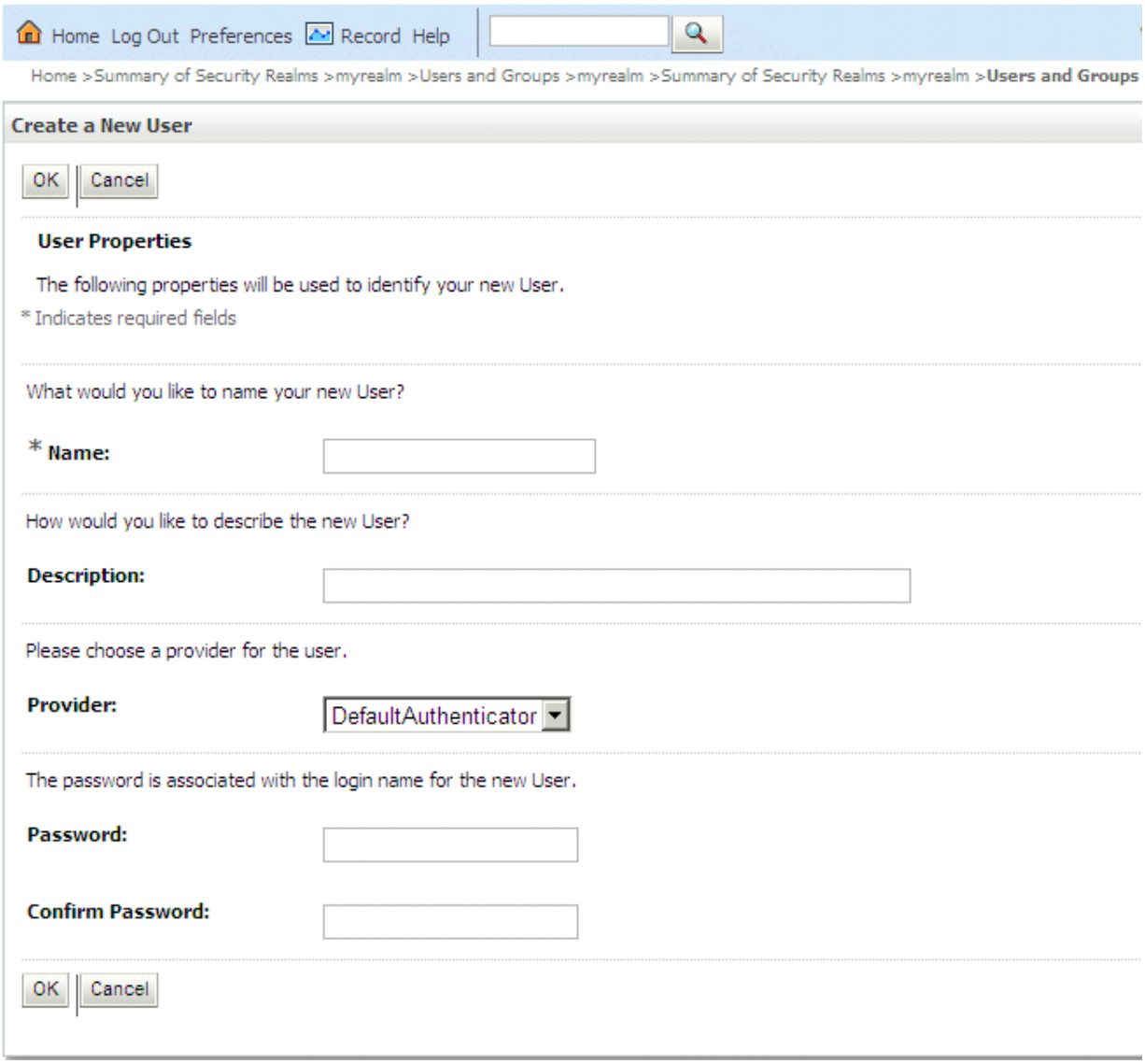
A user must exist in the Identity Store before you can provision the user in Oracle B2B. Although there are many tools that you can use to create users, one way is to use the **Security Realms** function in Oracle WebLogic Server Administration Console, as shown in Figure 5–4.

Figure 5–4 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** to add a user and user password, as shown in Figure 5–5.

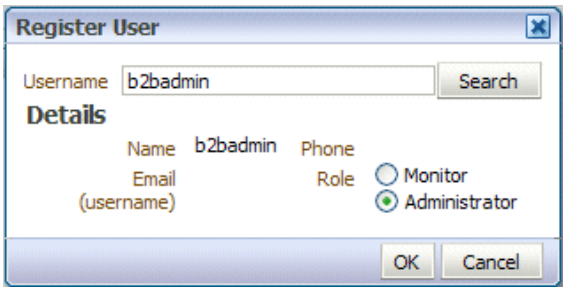
Figure 5–5 Oracle WebLogic Server Administration Console—Adding a New User



Task 2 Add a User in the Oracle B2B Interface

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 is it was created.
6. Select the **Monitor** or **Administrator** role and click **OK**.



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 you have added it, see "Changing Document Definitions" on page 7-32.

> **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 you can add it to the trading partner profile—see Chapter 4, "Creating Document Definitions."

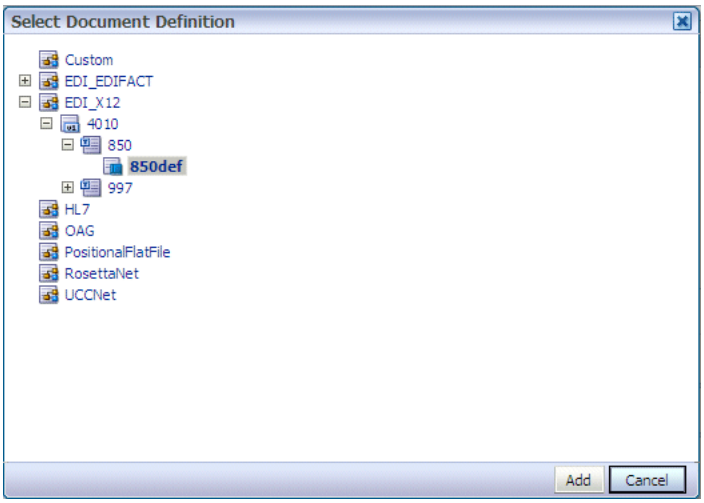
To add document definitions, do the following:

- Task 1, "Add Document Definitions"

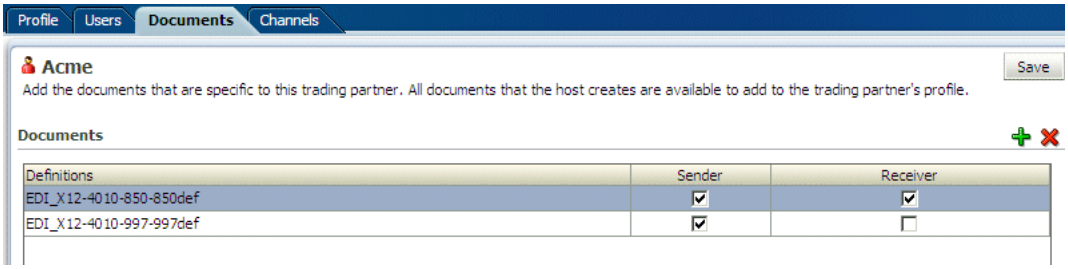
Task 1 Add Document Definitions

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, and click **Add**.



6. For each document listed, identify if the selected partner is the sender or receiver or both.



7. Click **Save**.

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.

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

Table 5–1 Channels Available in Oracle B2B

Protocol	Description
AS2-1.1	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 EDIINT AS2.
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 "About MLLP" on page 5-20 for more information.
ebMS-2.0 ebMS-1.0	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 (Preview mode for this release)	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.
Generic FTP-1.0	Transport by which messages are sent to or received from a file at a remote FTP server.
Generic SFTP-1.0	Transport by which messages are sent to or received from a file at a remote SFTP server.

Table 5–1 (Cont.) Channels Available in Oracle B2B

Protocol	Description
Generic JMS-1.0	Transport by which messages are sent to or received from a JMS queue or topic.
Generic HTTP-1.0	Transport by which messages are sent to or received from a Web server.
Generic Email-1.0	Transport by which messages are sent to or received from an e-mail server.

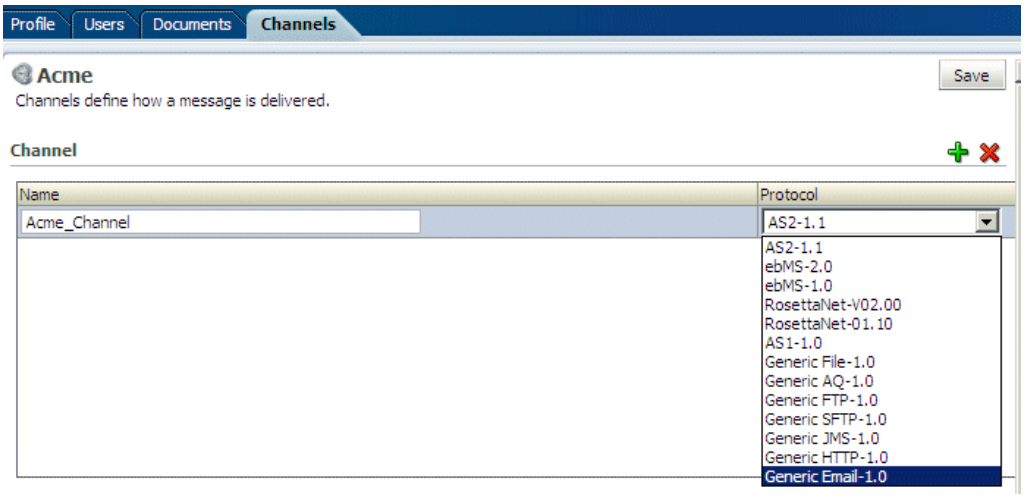
To configure a channel for a trading partner, do the following:

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



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

Channel Protocol Selected...	Transport Protocol Displayed...
AS2-1.1 ebMS-2.0, ebMS-1.0 RosettaNet-V02.00, RosettaNet-01.00 Generic HTTP-1.0	HTTP
AS1-1.0 Generic Email-1.0	Email
MLLP-1.0	TCP
Generic File-1.0	File
Generic AQ-1.0	AQ
Generic FTP-1.0	FTP
Generic SFTP-1.0	SFTP
Generic JMS-1.0	JMS

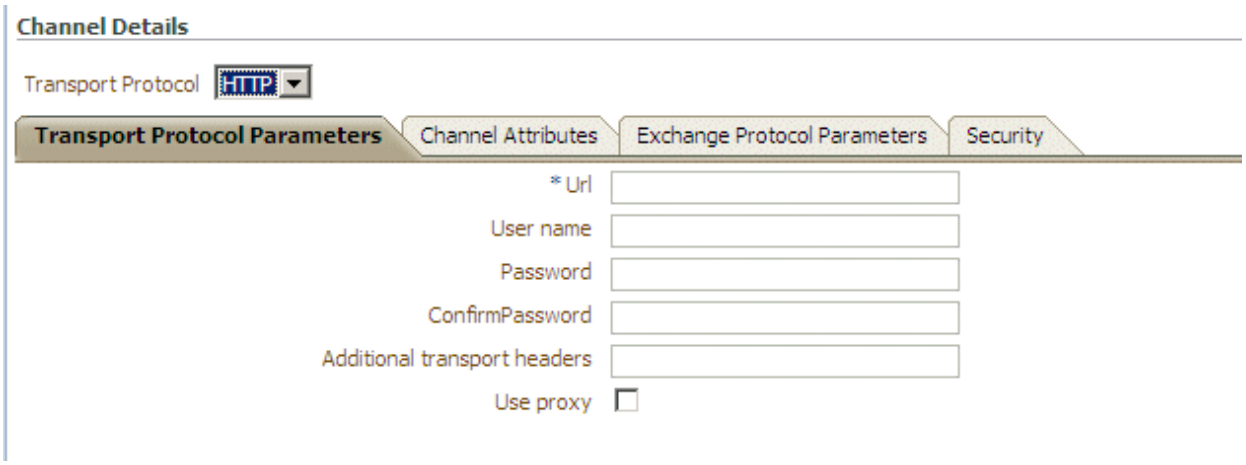
Task 2 Provide Transport Protocol Parameters

1. Click the **Transport Protocol Parameters** tab.
2. Provide transport protocol parameters, depending on the channel/transport protocols selected in Task 1.

   Table 5–3 describes the transport protocol parameters (listed in alphabetical order within the transport protocol parameters category) and the protocols to which the parameters apply.

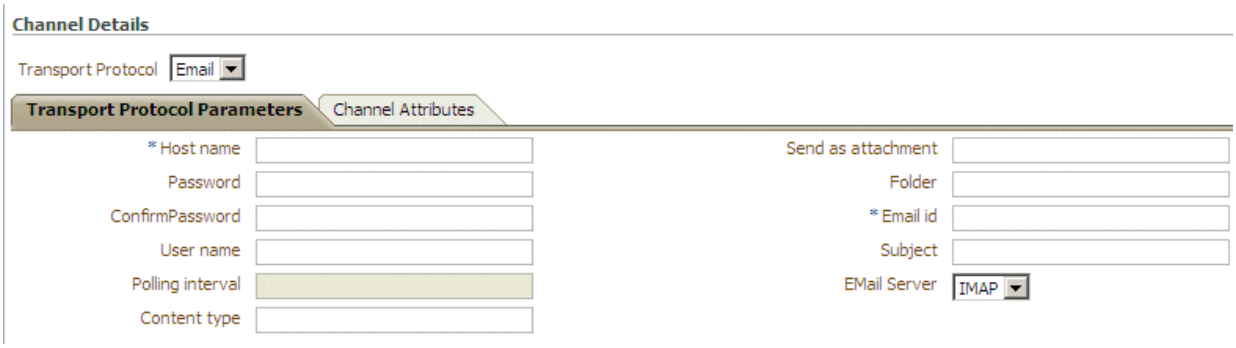
   - Figure 5–6 shows the HTTP transport protocol parameters.

Figure 5–6 HTTP Transport Protocol Parameters



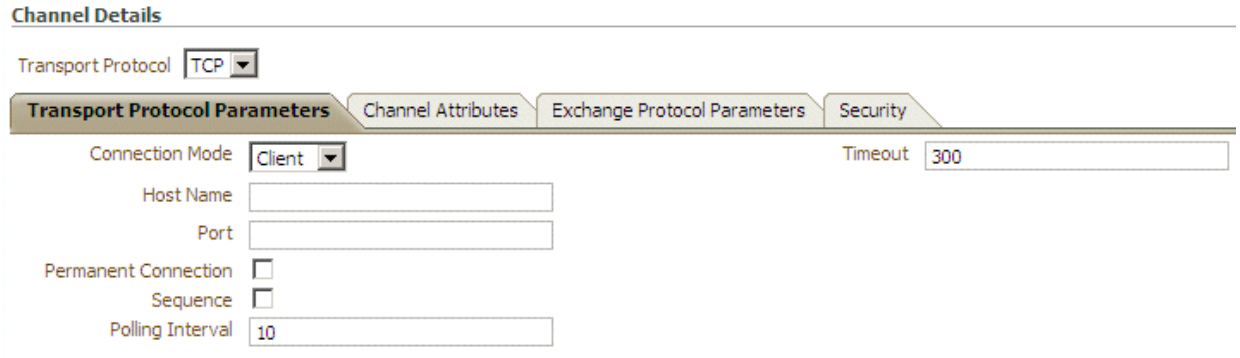
   - Figure 5–7 shows the Email transport protocol parameters.

Figure 5–7 Email Transport Protocol Parameters



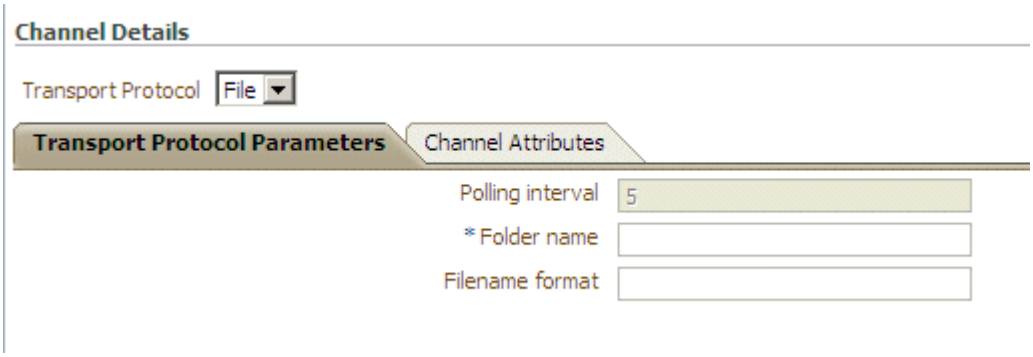
- Figure 5–8 shows the MLLP transport protocol parameters.

Figure 5–8 MLLP Transport Protocol Parameters



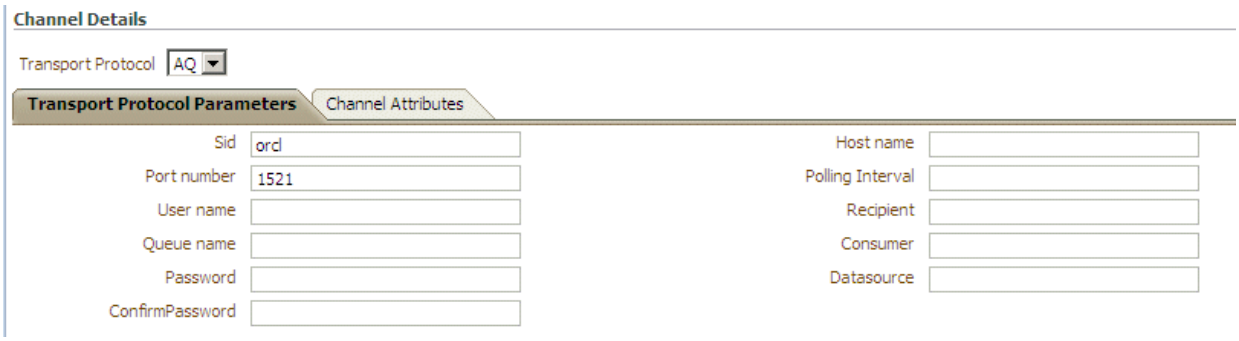
- Figure 5–9 shows the File transport protocol parameters.

Figure 5–9 File Transport Protocol Parameters



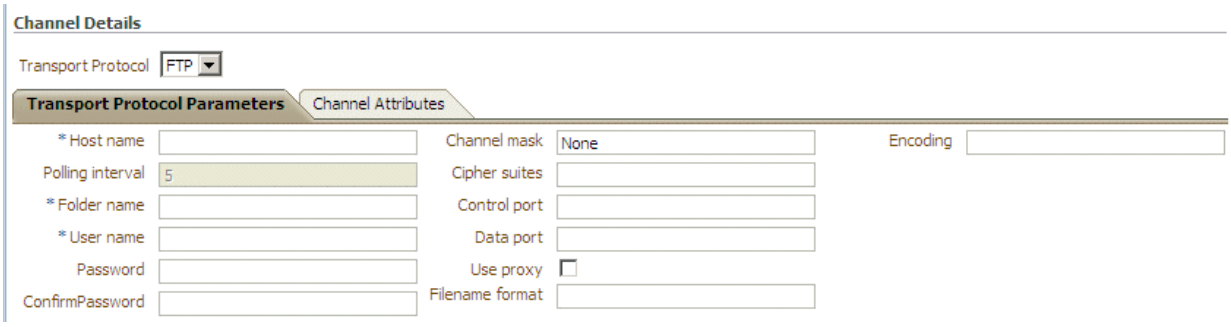
- Figure 5–10 shows the AQ transport protocol parameters.

Figure 5–10 AQ Transport Protocol Parameters



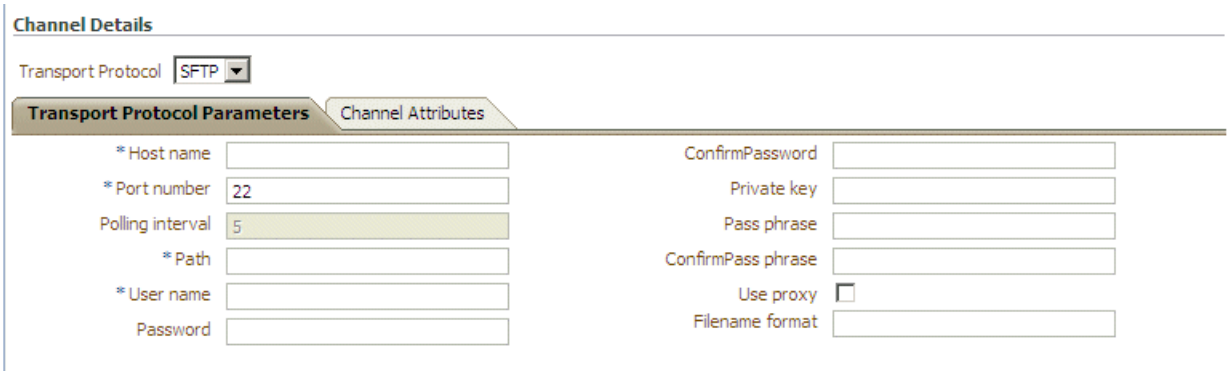
- Figure 5–11 shows the FTP transport protocol parameters.

Figure 5–11 FTP Transport Protocol Parameters



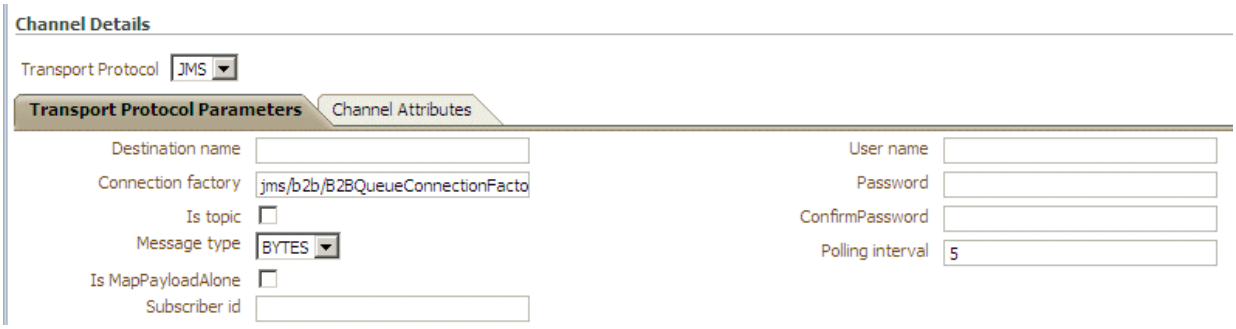
- Figure 5–12 shows the SFTP transport protocol parameters.

Figure 5–12 SFTP Transport Protocol Parameters



- Figure 5–13 shows the JMS transport protocol parameters.

Figure 5–13 JMS Transport Protocol Parameters



3. Click **Save**.

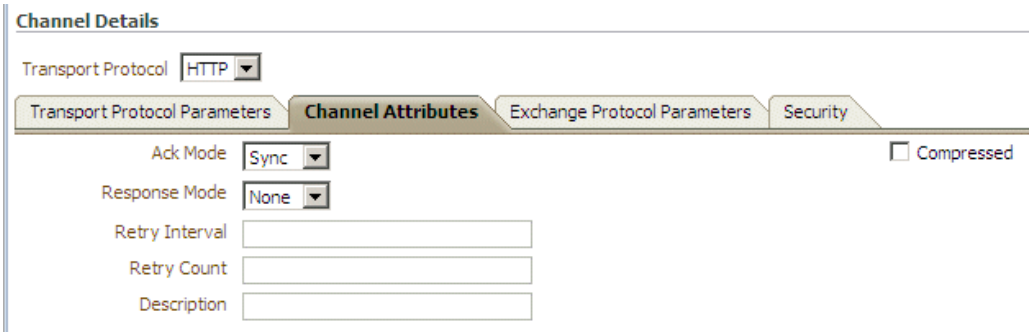
Task 3 Provide Channel Attributes

1. Click the **Channel Attributes** tab.
2. Provide channel attributes, depending on the channel/transport protocols selected in Task 1.

   Table 5–3 describes the channel attributes (listed in alphabetical order within the channel attributes category) and the protocols to which the attributes apply.

   - Figure 5–14 shows the HTTP channel attributes.

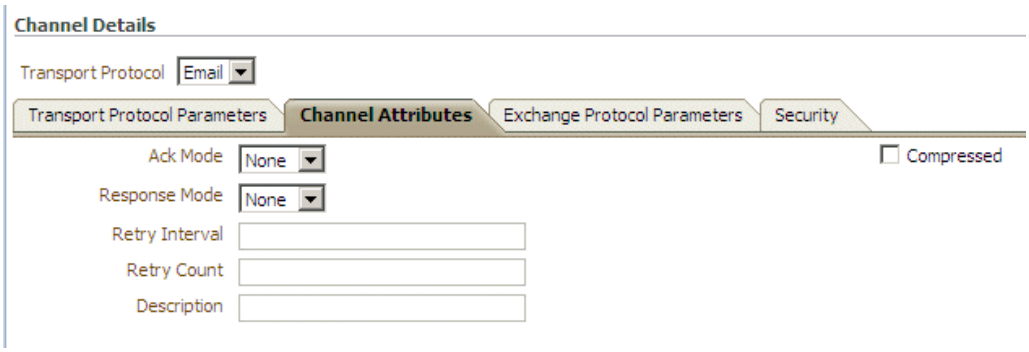
Figure 5–14 HTTP Channel Attributes



Note: For Generic HTTP-1.0, the Ack Mode, Response Mode, and Compressed attributes shown in Figure 5–14 are not available.

   - Figure 5–15 shows the Email channel attributes.

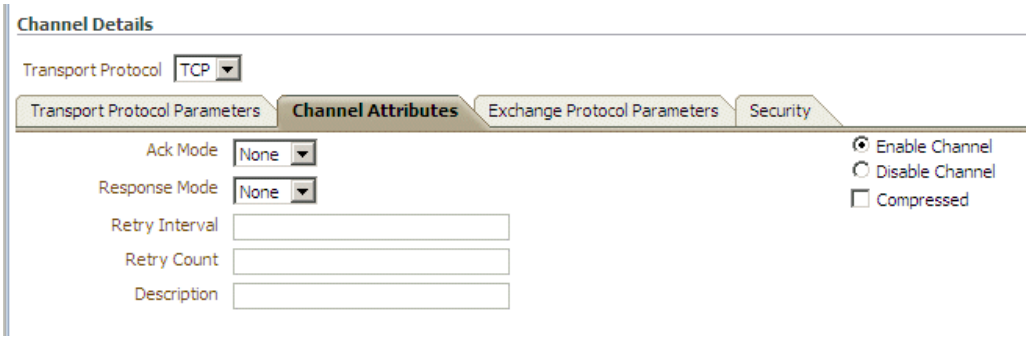
Figure 5–15 Email Channel Attributes



Note: For Generic Email-1.0, the Ack Mode, Response Mode, and Compressed attributes shown in Figure 5–15 are not available.

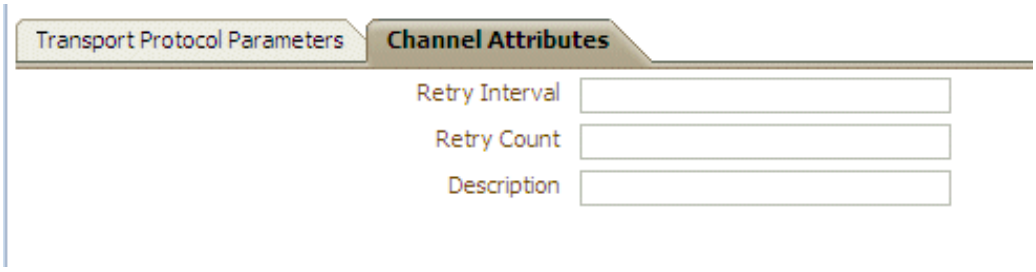
- Figure 5–16 shows the MLLP channel attributes

Figure 5–16 MLLP Channel Attributes



- Figure 5–17 shows the File, AQ, FTP, SFTP, and JMS channel attributes.

Figure 5–17 Channel Attributes for Generic File, AQ, FTP, SFTP, and JMS



3. Click **Save**.

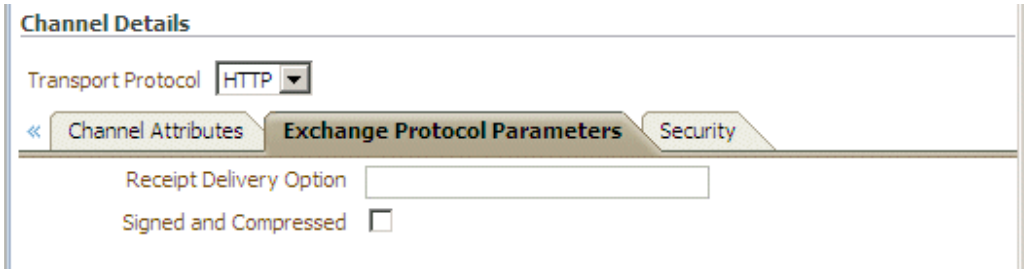
Task 4 Provide Exchange Protocol Parameters

1. Click the **Exchange Protocol Parameters** tab.
2. Provide exchange protocol parameters, depending on the channel/transport protocols selected in Task 1.

Table 5–3 describes the exchange protocol parameters (listed in alphabetical order within the exchange protocol parameters category) and the protocols to which the parameters apply.

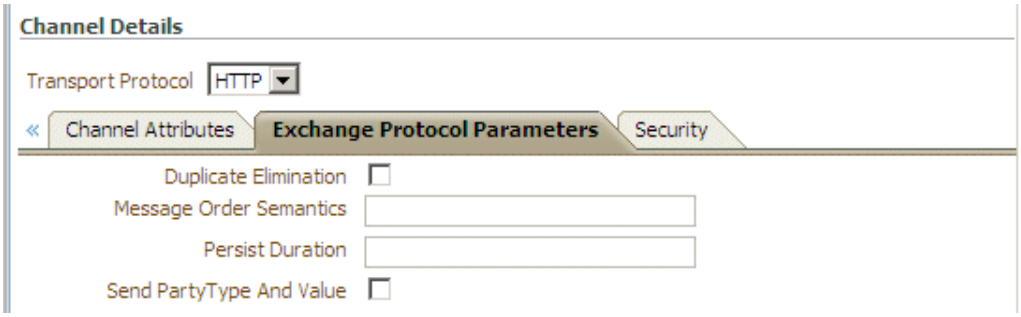
- Figure 5–18 shows HTTP - AS2-1.1 exchange protocol parameters.

Figure 5–18 Exchange Protocol Parameters for HTTP - AS2-1.1



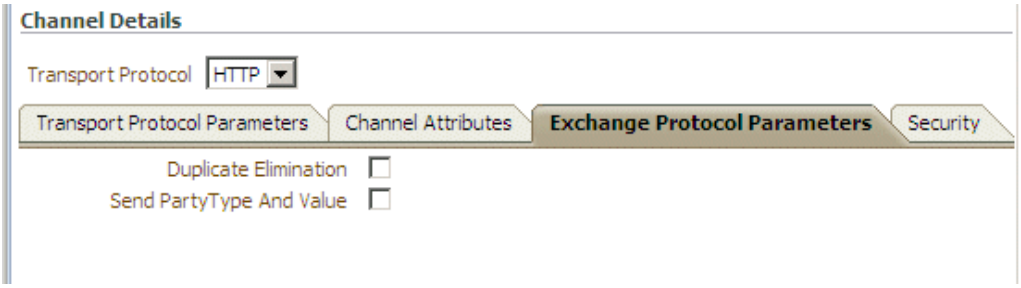
- Figure 5–19 shows HTTP - ebMS-2.0 exchange protocol parameters.

Figure 5–19 Exchange Protocol Parameters for HTTP - ebMS-2.0



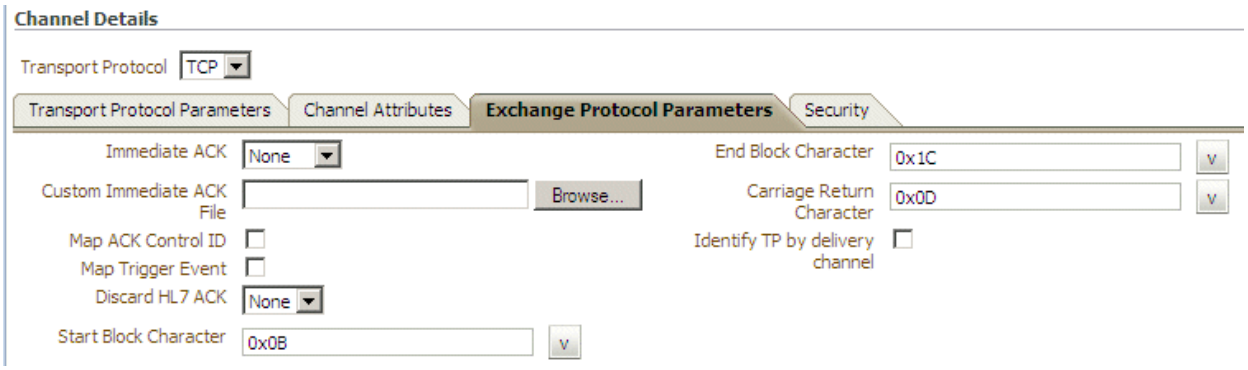
- Figure 5–20 shows HTTP - ebMS-1.0 exchange protocol parameters.

Figure 5–20 Exchange Protocol Parameters for HTTP - ebMS-1.0



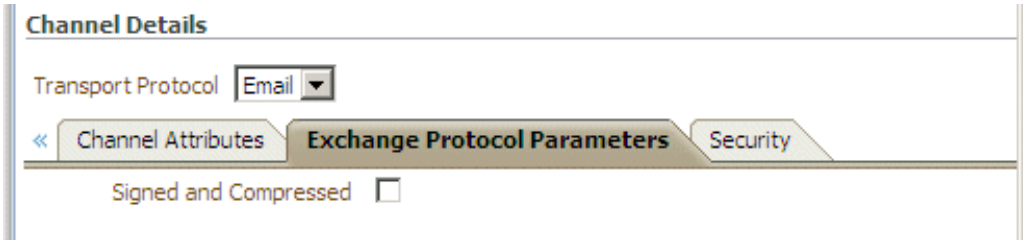
- Figure 5–21 shows the TCP - MLLP-1.0 exchange protocol parameters.

Figure 5–21 Exchange Protocol Parameters for TCP - MLLP-1.0



- Figure 5–22 shows the Email - AS1-1.0 exchange protocol parameters.

Figure 5–22 Exchange Protocol Parameters for Email - AS1-1.0



3. Click **Save**.

Task 5 Configure Security

1. Click the **Security** tab.
2. Provide security parameters, depending on the channel/transport protocols selected in Task 1.

   Table 5–3 describes the security parameters (listed in alphabetical order within the security category) and the protocols to which the parameters apply.

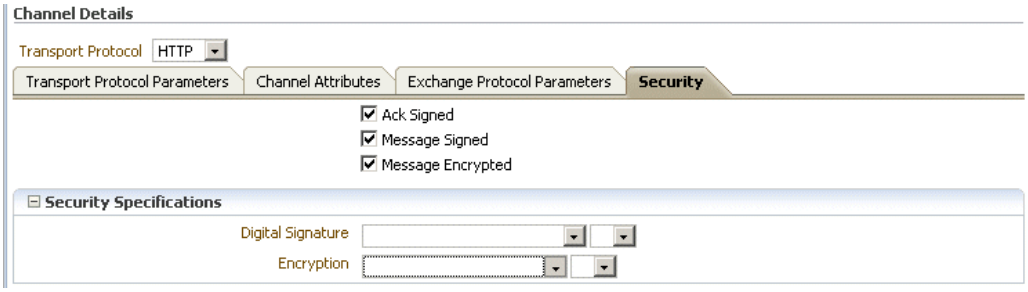
   The **Digital Signature** and **Encryption** lists are populated with the available certificates when the Key Store location is provided for the host trading partner. See Task 6, "Provide Key Store Information for the Host Trading Partner" for more information.

   Note: Message encryption using an AES setting is preferable, where available. See the security parameters in Table 5–3.

   Security parameters do not apply to the MLLP channel.

   - Figure 5–23 shows the security parameters for the AS2-1.1, ebMS-2.0, ebMS-1.0, RosettaNet-V02.00, and AS1-1.0 protocols.

Figure 5–23 Security Parameters for the AS2-1.1, ebMS-2.0, ebMS-1.0, RosettaNet-V02.00, and AS1-1.0 Protocols



- Figure 5–24 shows the security parameters for RosettaNet-01.10. For RosettaNet-01.10, the Message Encrypted parameter is not available.

Figure 5–24 Security Parameters for RosettaNet-01.10

Note: 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.

3. Click **Save**.

About MLLP

A permanent MLLP (server/client) delivery channel is bidirectional, that is, it can be used for sending and receiving messages. Other delivery channels are not bidirectional. An MLLP delivery channel is configured for the remote trading partner only. This channel can be either a server or a client channel, used to send or receive messages. You must configure both servers (sender and receiver) MLLP (server/client) channels either in permanent mode or in transient channel 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, Acme can have the server/client MLLP permanent channel and GlobalChips can have the server/client MLLP permanent channel. MLLP channels configured in permanent-transient and transient-permanent modes are not valid. Because MLLP is a bidirectional channel, you do not create an MLLP listening channel. You can use the same MLLP delivery channel for sending and receiving messages.

Adding Channel Details

Channel details include transport protocol parameters, channel attributes, exchange protocol parameters, and security specifications. Table 5–3 describes these details.

Table 5–3 Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
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 (trading partner address, such as a URI)	-
Additional transport headers	The custom HTTP headers used to send messages to a trading partner	AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) Generic HTTP (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)
Channel mask	To enable SSL for FTP, enter one of the following: ▪ `Control`—Encrypts the control channel ▪ `Data`—Encrypts the data channel ▪ `Both`—Encrypts both the data and control channels The default is **None** (no SSL).	Generic FTP (optional)
Cipher suites	Provide the preferred cipher for encryption.	Generic FTP (optional)
Connection factory	The JNDI location or Java class name for the connection factory, as in `jms/b2b/B2BQueueConnectionFactory`.	Generic JMS (optional)
Connection Mode	Select from Client or Server.	MLLP-1.0 (required; for remote trading partners only)
Consumer	The client that receives the message.	Generic AQ (optional)
Content type	The content type of the payload being sent over e-mail. The default content type is `text/plain`; other examples include `application/xml` and `application/edi`. This value is used only for the delivery channel (to send e-mail) and not for the listening channel. On the listening channel side, intelligence is built into the transport adapter to deal with different content types, so no configuration is required.	AS1 (optional) Generic Email (optional)
Control port	Provide a value to change the default FTP port value (21)	Generic FTP (optional)
Data port	The static port used for an active FTP connection	Generic FTP (optional)
Data source	The JNDI name of the database data source	Generic AQ (optional)
Destination name	The JMS destination name	Generic JMS (optional)
Email ID	The destination e-mail	AS1 (required) Generic Email (required)

Table 5–3 (Cont.) Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
Email Server	Select **IMAP** or **POP3**.	AS1 (required) Generic Email (required)
Encoding	The encoding to be used for the file transfer	Generic FTP (optional)

Filename format	The following file name formats can be used:
	`%FROM_PARTY%`
	`%TO_PARTY%`
	`%DOCTYPE_NAME%`
	`%DOCTYPE_REVISION%`
	`%MSG_ID%`
	`%TIMESTAMP%`
	The following file name format can be used for ebMS documents only:
	`%ACTIONNAME%`
	These file name formats can be used in any combination; for example,
	`%TO_PARTY%_%DOCTYPE_NAME%_%DOCTYPE_REVISION%.dat`
	produces something like `Acme_4010_850.dat`. Any file extension is allowed.
		Generic File (optional)
		Generic FTP (optional)
		Generic SFTP (optional)

Folder	An absolute directory path is recommended.	AS1 (optional) Generic Email (optional)
Folder name	An absolute directory path is recommended.	Generic File (required) Generic FTP (required)
Host name	The trading partner's transport or e-mail server exchanging messages. 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).	AS1 (required) Generic AQ (optional) Generic FTP (required) MLLP-1.0 (required; for remote trading partners only) Generic SFTP (required) Generic Email (required)
Is Map Payload Alone	Indicates that the JMS map message contains only the payload	Generic JMS (optional)
Is topic	Select to indicate that JMS is communicating with a topic (not a queue).	Generic JMS (optional)
Message type	Select a JMS message type: **BYTES**, **TEXT**, or **MAP**.	Generic JMS (optional)
Pass phrase and Confirm pass phrase	If you enter a private key file location, and if the private key file is pass-phrase protected, then enter the pass phrase.	Generic SFTP (optional)

Table 5–3 (Cont.) Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
Password and Confirm Password	To use password authentication, provide a Key Store password, which is used for HTTP basic authentication.	AS1 (optional) AS2 (optional) Generic AQ (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) Generic FTP (optional) Generic HTTP (optional) Generic SFTP (optional) Generic JMS (optional) Generic Email (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)
Path	The absolute directory path where messages are sent from or received.	Generic SFTP (required)
Permanent Connection	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.	MLLP-1.0 (optional; for remote trading partners only)
Polling interval	The time interval in milliseconds during which Oracle B2B polls the server for inbound messages.	AS1 (optional) Generic File (not available) Generic AQ (optional) Generic FTP (not available) MLLP-1.0 (optional; for remote trading partners only) Generic SFTP (not available) Generic JMS (optional) Generic Email (not available)
Port number (or Port)	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. 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.	Generic AQ (optional) MLLP-1.0 (required; for remote trading partners only) Generic SFTP (required)
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 (optional)
Queue name	The AQ queue name	Generic AQ (optional)
Recipient	The AQ recipient	Generic AQ (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.	AS1 (optional) Generic Email (optional)

Table 5–3 (Cont.) Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
Sequence	If enabled, all inbound MLLP messages are sequenced. This feature is in preview mode for this release.	MLLP-1.0 (optional; for remote trading partners only)
SID	System ID to identify an Oracle database	Generic AQ (optional)
Subject	The subject header of the e-mail message	AS1 (optional) Generic Email (optional)
Subscriber ID	The JMS subscriber ID is required if JMS is communicating with a topic.	Generic JMS
Timeout	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).	MLLP-1.0 (optional; for remote trading partners only)
URL	The HTTP or HTTPS endpoint URL of the trading partner.	AS2 (required) ebMS-2.0 (required) ebMS-1.0 (required) Generic HTTP (required) RosettaNet-V02.00 (required) RosettaNet-01.10 (required)
User name	The user name to connect to the target server, used for HTTP basic authentication.	AS1 (optional AS2 (optional) Generic AQ (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) Generic FTP (required) Generic HTTP (optional) Generic SFTP (required) Generic JMS (optional) Generic Email (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)
Use proxy	Select a proxy server if used.	Generic FTP (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) Generic HTTP (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional) Generic SFTP (optional)
Channel Attributes	The channel is the communication interface between the host trading partner's host application and its installation	-

Table 5–3 (Cont.) Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
Ack Mode	Select **Sync**, **Async**, or **None**, for the mode in which the trading partner receives messages. Select **None** for all generic exchanges.	AS1 (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)
Compressed	Select for message compression.	AS1 (optional) AS2 (optional)
Description	Optional	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) Generic SFTP (optional) Generic JMS (optional) Generic Email (optional)
Enable/Disable Channel	The channel is the communication interface between the host trading partner's host application and its installation.	Generic Email (Required) MLLP-1.0 (required; for remote trading partners only)
Internal **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.	Select this option if the channel is internal to the host trading partner's enterprise.	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) Generic SFTP (optional) Generic JMS (optional) Generic Email (optional)

Table 5–3 (Cont.) Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
Response Mode	Select **Sync**, **Async**, or **None**.	AS1 (required) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)
Retry Count	The number of times that Oracle B2B retries to send the message.	AS1 (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) Generic File (optional) Generic AQ (optional) Generic FTP (optional) Generic HTTP (optional) MLLP-1.0 (optional; for remote trading partners only) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional) Generic SFTP (optional) Generic JMS (optional) Generic Email (optional)
Retry Interval	The time interval in seconds during which Oracle B2B attempts to resend the message. A time interval of 2 minutes increments the HH:MM:SS timestamp as follows: If the sent timestamp is 3:42:58, then 42 seconds is incremented by 2 minutes and the retry is sent at 3:44:00. The seconds are dropped in the retry increment. Subsequent retries are at 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.	AS1 (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) Generic File (optional) Generic AQ (optional) Generic FTP (optional) Generic HTTP (optional) MLLP-1.0 (optional; for remote trading partners only) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional) Generic SFTP (optional) Generic JMS (optional) Generic Email (optional)
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, encryption, and compression.	-
Carriage Return Character	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).	MLLP-1.0 (optional; for remote trading partners only)
Custom Immediate ACK File	Browse for a file with a customized acknowledgment.	MLLP-1.0 (optional; for remote trading partners only)

Table 5–3 (Cont.) Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
Discard HL7 ACK	Stops the incoming acknowledgment at the transport level if the selected code is in MSA.2. An entry is made for the wire message report.	MLLP-1.0 (optional; for remote trading partners only)
Duplicate Elimination	If enabled, a duplicate elimination header is added for an outbound message. This flag does not apply to the inbound message flow.	ebMS-2.0 (optional) ebMS-1.0 (optional)
End Block Character	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).	MLLP-1.0 (optional; for remote trading partners only)
Identify TP by Delivery Channel	The trading partner is identified using the delivery channel.	MLLP-1.0 (optional; for remote trading partners only)
Immediate ACK	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. ■ Simple: B2B sends the predefined acknowledgment message to the sender and does not parse the message. ■ Custom: B2B reads the custom HL7 acknowledgment message based on a configurable file content.	MLLP-1.0 (optional; for remote trading partners only)
Map ACK Control ID	Select to enable the mapping of the message header of the business message to the message header of the *immediate* acknowledgment.	MLLP-1.0 (optional; for remote trading partners only)
Map Trigger Event	Sends an immediate acknowledgment with a trigger event.	MLLP-1.0 (optional; for remote trading partners only)
Message Order Semantics	A placeholder for CPP/CPA; not involved during run time.	ebMS-2.0 (optional)
Persist Duration	A placeholder for CPP/CPA; not involved during run time.	ebMS-2.0 (optional)
Receipt Delivery Option	This parameter is used to configure a URL to which MDN has to be sent back in the case of an asynchronous mode.	AS2 (optional)

Table 5–3 (Cont.) Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
Send Party Type and Value	If enabled, the send party type and value from the message header are sent to the back-end application.	ebMS-2.0 (optional) ebMS-1.0 (optional)
Signed and Compressed	If selected, the message is first signed, and then compressed.	AS1 (optional)
Start Block Character	This value can be only one character. The start block character does not appear in the wire message payload. The default value is 0X08 (hexadecimal).	MLLP-1.0 (optional; for remote trading partners only)
Security Parameters	-	-
Ack Signed	Select this option to ensure that the responder acknowledges receipt of the messages; nothing needs to be provided.	AS1 (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)
Digital Signature	To use a digital signature certificate, the Key Store *must* have the corresponding private key. If **Message Signed** is selected, then select one of the following for AS1 and AS2: SMIME 3.0 with MD5 - RSA SMIME 3.0 with SHA1 - RSA If **Message Signed** is selected, then select one of the following for ebMS-2.0 and ebMS-1.0: XMLDSIG with SHA1 - RSA XMLDSIG with SHA1 - DSA If **Message Signed** is selected, then select one of the following for RosettaNet-V02.00: SMIME 3.0 with MD5 - RSA SMIME 3.0 with SHA1 - RSA SMIME 2.0 with MD5 - RSA SMIME 2.0 with SHA1 - RSA XMLDSIG with SHA1 - RSA XMLDSIG with SHA1 - DSA If **Message Signed** is selected, then select one of the following for RosettaNet-01.10: SMIME 3.0 with MD5 - RSA SMIME 3.0 with SHA1 - RSA SMIME 2.0 with MD5 - RSA SMIME 2.0 with SHA1 - RSA	AS1 AS2 ebMS-2.0 ebMS-1.0 RosettaNet-V02.00 RosettaNet-01.10

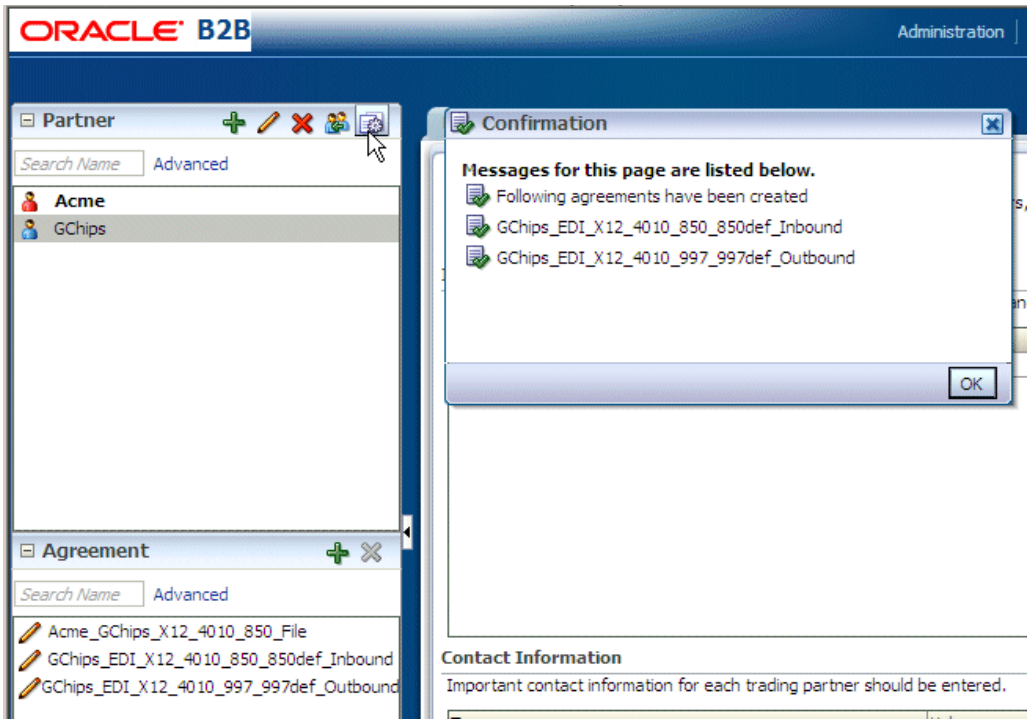
Table 5–3 (Cont.) Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
Encryption	To use an encryption certificate, no private key entry is needed. If **Message Encrypted** is selected, then select one of the following for AS1 and AS2: SMIME 3.0 with DES SMIME 3.0 with 3DES SMIME 3.0 with RC2 - 40 SMIME 3.0 with RC2 - 64 SMIME 3.0 with RC2 - 128 If **Message Encrypted** is selected, then select one of the following for ebMS-2.0 and ebMS-1.0: XMLENC with 3DES - RSA-v1.5 XMLENC with AES-128 RSA-OAEP XMLENC with AES-192 RSA-OAEP XMLENC with AES-256 RSA-OAEP	AS1 AS2 ebMS-2.0 ebMS-1.0 RosettaNet-V02.00 (optional)
Message Encrypted	Select this option to enable message encryption. This option requires you to select an encryption schema in the **Encryption** field.	AS1 (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) RosettaNet-V02.00 (optional)
Message Signed	Select this option to provide a digital signature in the Digital Signature field.	AS1 (optional) AS2 (optional) ebMS-2.0 (optional) ebMS-1.0 (optional) RosettaNet-V02.00 (optional) RosettaNet-01.10 (optional)

Using the Auto Create Agreement Feature

In the **Partner** area, shown in Figure 5–25, you can use the **Auto Create Agreement** icon to create an agreement for a remote trading partner.

Figure 5–25 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.

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:

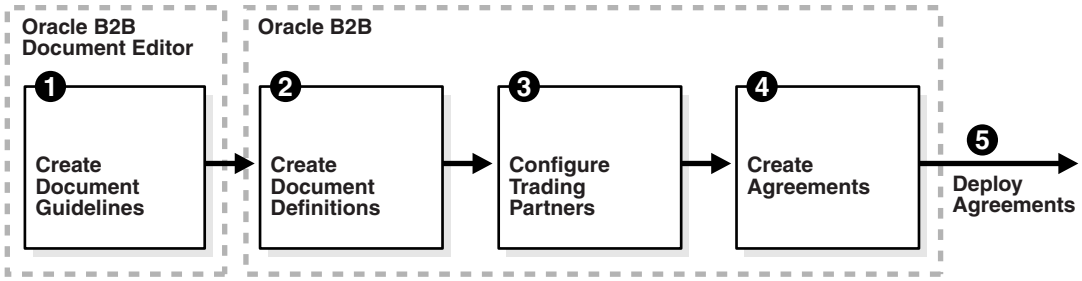
- For the AS2-1.1 exchange protocol, the AS2 identifier is used.
- For the EDI X12 document protocol, the Sender Group ID and Sender Interchange ID are used.

6

Creating and Deploying Trading Partner Agreements

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:

- Introduction to Agreements
- Creating an Agreement
- Deploying an Agreement
- 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

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.

Creating an Agreement

Figure 6–2 shows the Oracle B2B interface for working with agreements.

Figure 6–2 Creating an Agreement

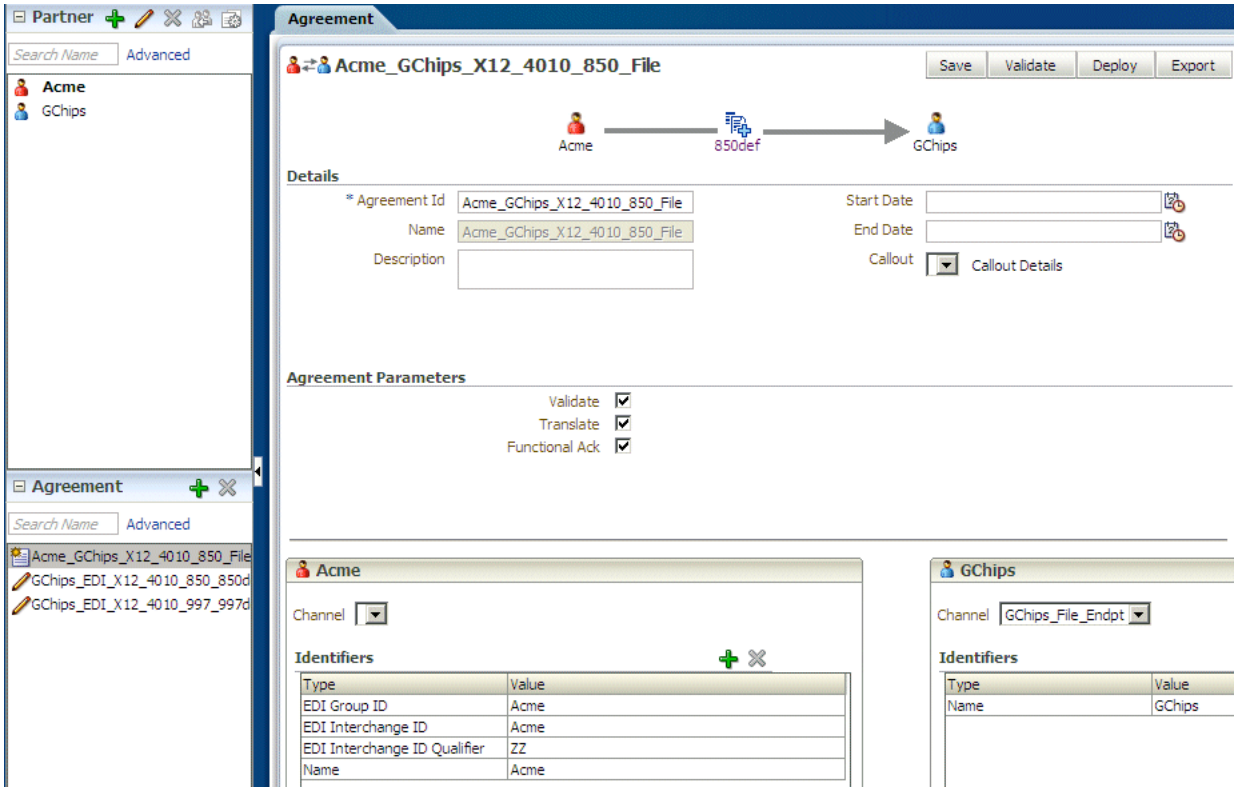
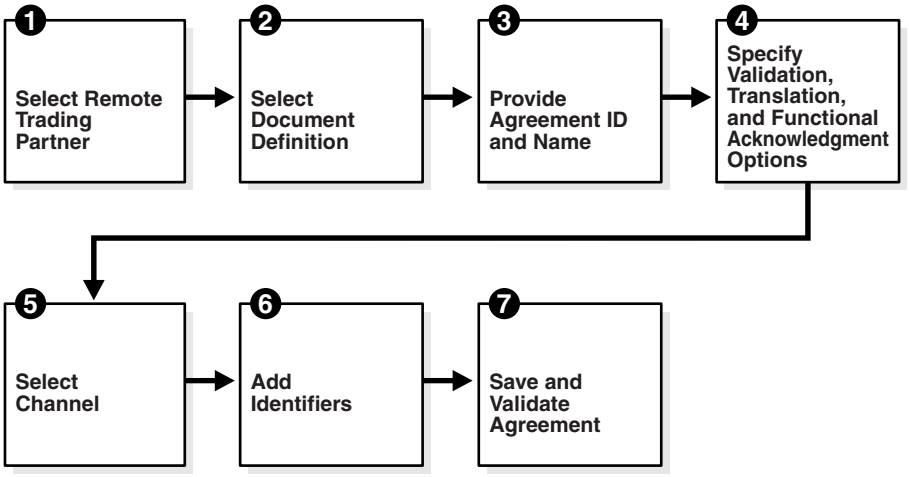


Figure 6–3 shows the steps to create an agreement.

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



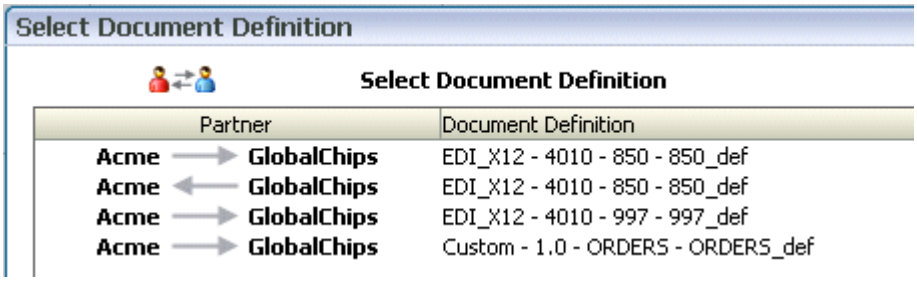
Step 1: Identify the remote trading partner

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



For an exchange in which you need both outbound and inbound agreements, you 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 6–1 Agreement Options

Option	Description
Validate	Select to enable validation of the document against the configured ECS file
Translate	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 is unable to correlate the business message with the functional acknowledgment, irrespective of the value of the B2B Handle FA property.
Functional Ack	Select to enable the functional acknowledgment for success or error criterion.

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. See Chapter 9, "Creating Types," for more information.

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, 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 "Deploying an Agreement" on page 6-5 to continue.

Deploying an Agreement

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

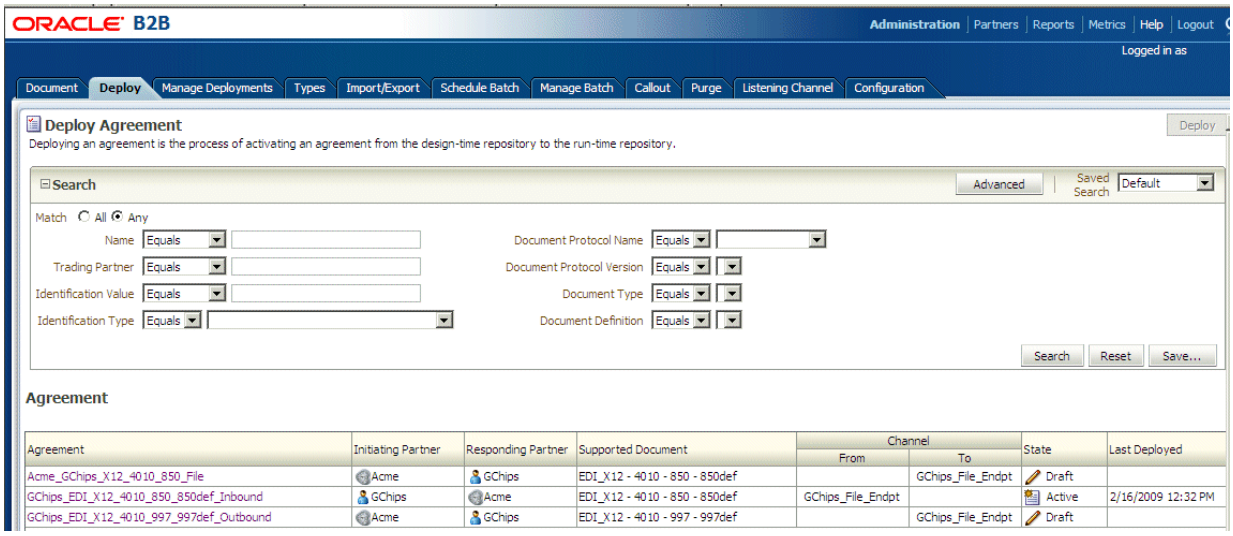
After deploying an agreement, use the **Manage Deployments** tab and the **Reports** tab. See the following for more information:

- 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.

Figure 6–5 The Deploy Tab—Lists Valid Agreements



Note: Turn off validation during deployment by setting the property `oracle.tip.b2b.deploy.validation=false` in `b2b-config.xml`, which is in

DOMAIN_HOME`/config/soa-infra/configuration/`

Turning off validation is useful when deploying large numbers of agreements, where you are certain that the data is valid. It requires a SOA Server restart.

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**.

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.

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.

Part III

Oracle B2B Administration

This part describes how to use Oracle B2B administration features.

This part contains the following chapters:

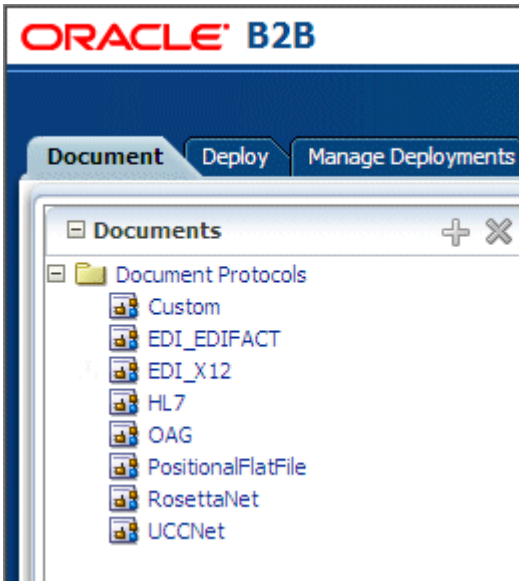
- Chapter 7, "Using Document Protocols"
- Chapter 8, "Managing Deployments"
- Chapter 9, "Creating Types"
- Chapter 10, "Importing and Exporting Data"
- Chapter 11, "Batching EDI Messages"
- Chapter 12, "Managing Callouts"
- Chapter 13, "Purging Data"
- Chapter 14, "Configuring Listening Channels"
- Chapter 15, "Configuring B2B System Parameters"

7

Using Document Protocols

A document protocol defines the document type of the message payload. Oracle B2B document protocols are shown in Figure 7–1; however OAG, PositionalFlatFile, UCCNet, and non-XML Custom documents are in preview mode in this release. See "Features in Preview Mode" on page 1-4 for the complete list of preview-mode protocols.

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:

- Using the Custom Document Protocol
- Using the EDI EDIFACT Document Protocol
- Using the EDI X12 Document Protocol
- Using the HL7 Document Protocol
- Using the OAG Document Protocol
- Using the Positional Flat File Document Protocol
- Using the RosettaNet Document Protocol

- Using the UCCnet Document Protocol
- Summary of Document Protocol Parameter Types
- Changing Document Definitions
- Using Document Routing IDs

For related information, see the following:

- Chapter 3, "Creating Guideline Files"
- Chapter 4, "Creating Document Definitions"

Using the Custom Document Protocol

Oracle B2B supports custom document protocols to create documents needed for proprietary transactions, including document definitions for XML and non-XML messages. With XML messages, you have the advantage of schema enforcement (XSDs). With non-XML messages, you can create trading partner agreements for specific message types. The non-XML implementation of the Custom document protocol is in preview mode in this release.

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. For non-XML documents such as a flat file, you can specify start and end positions or a document routing ID.

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

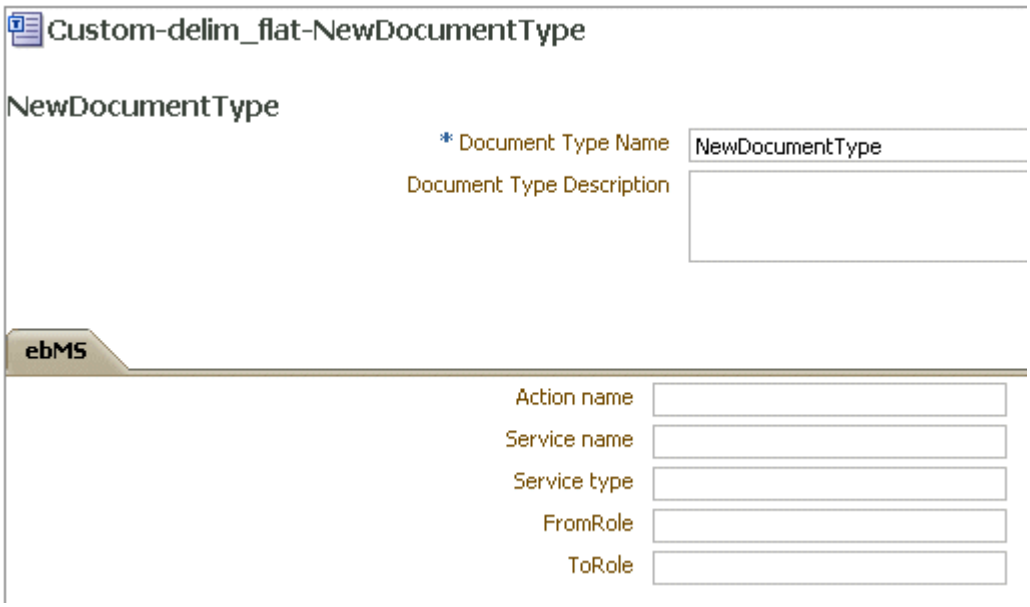


Table 7–1 describes the document type parameters for a Custom document.

Table 7–1 Document Type Parameters for a Custom Document

Parameter	Description
ebMS Tab	-
Action name	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.
Service name	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.
Service type	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.
From Role	The trading partner that sends the message. A value provided here overrides the **Identifiers** values supplied on the **Profile** tab.
To Role	The trading partner that receives the message. A value provided here overrides the **Identifiers** values supplied on the **Profile** tab.

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

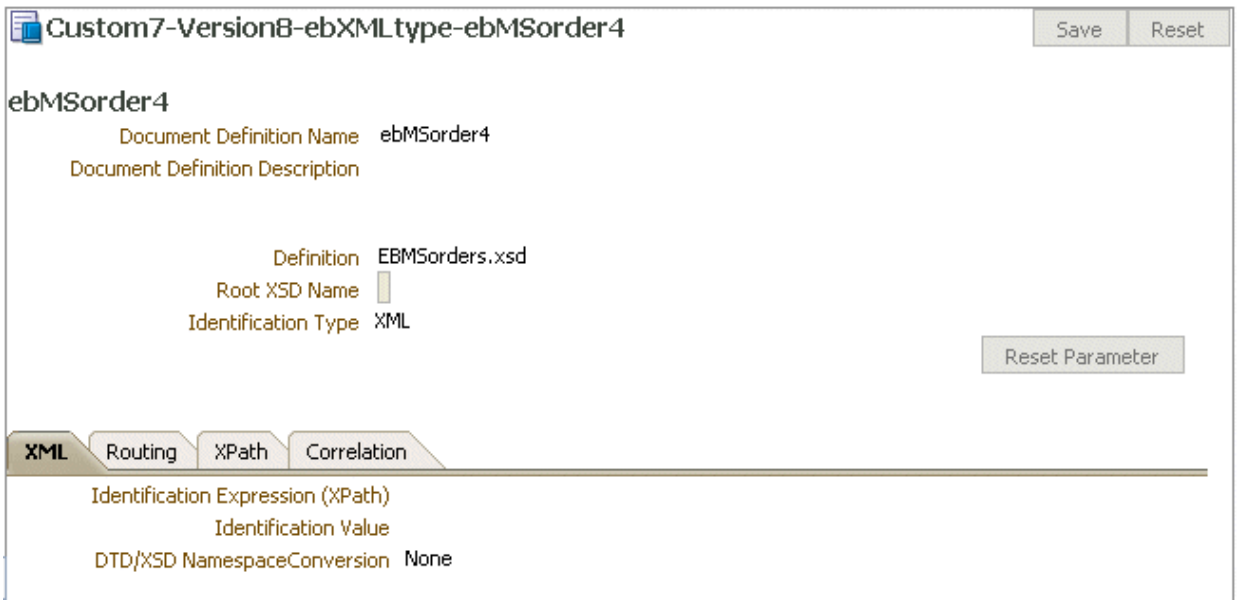


Figure 7–4 shows the document definition parameters for a flat-file Custom document.

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

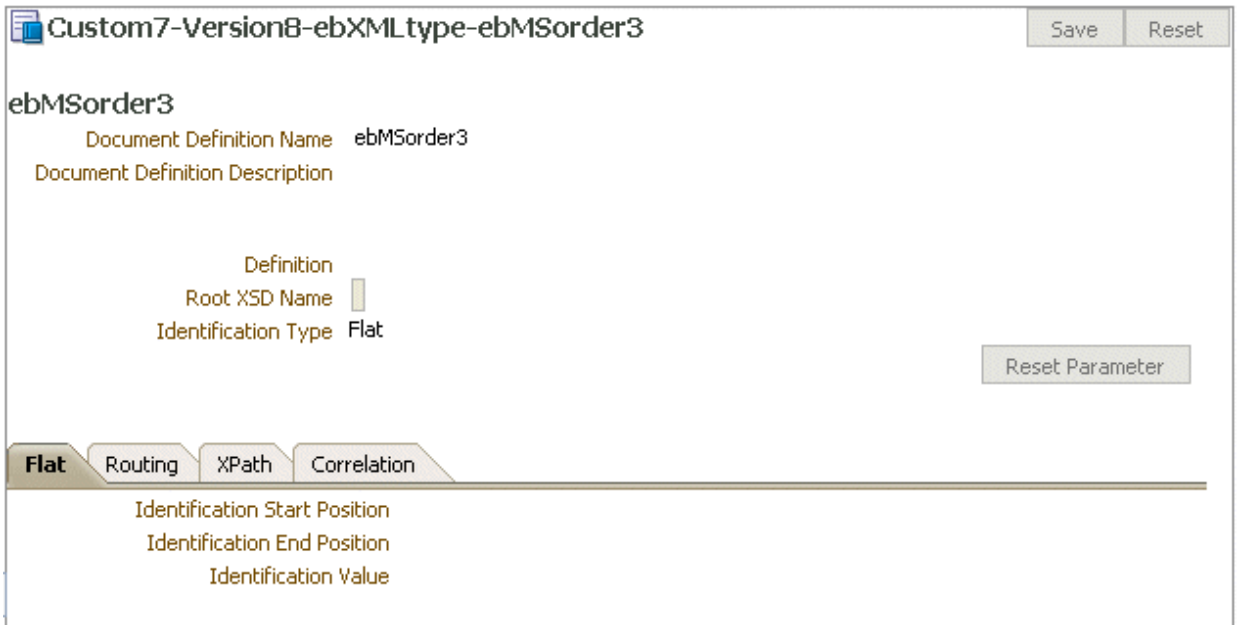


Table 7–2 describes the document definition parameters for a Custom document.

Table 7–2 Document Definition Parameters for a Custom Document

Parameter	Description
XML Tab	(Available if **XML** is selected from **Identification Type**)
Identification Expression (XPath)	Locates a node in the XML payload
Identification Value	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.
DTD/XSD NamespaceConversion	Select from **None**, **Both**, **Inbound**, or **Outbound**.
Routing Tab	-
Document Routing ID	Sets the consumer name to the back-end application
XPath Tab	See "How to Configure the XPath Expression for a Custom XML Document" on page 7-5 for more information
XPath Name1	The XML XPath name for retrieving the value from the payload
XPath Expression1	The XML XPath expression for retrieving the value from the payload
XPath Name2	The XML XPath name for retrieving the value from the payload
XPath Expression2	The XML XPath expression for retrieving the value from the payload
XPath Name3	The XML XPath name for retrieving the value from the payload
XPath Expression3	The XML XPath expression for retrieving the value from the payload
Correlation Tab	-
Correlation From XPath Name	The name of the correlation property for initiating the correlation

Table 7–2 (Cont.) Document Definition Parameters for a Custom Document

Parameter	Description
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation
Correlation To XPath Name	The name of the correlation property for the correlation
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation
Flat Tab (preview mode)	-
Identification Start Position	Used in combination with the end position to retrieve a value from the payload between the start and end positions
Identification End Position	Used in combination with the start position to retrieve a value from the payload between the start and end positions
Identification Value	A value between the start and end positions

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.

You have the following options when configuring an XPath expression:

- 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

Option 1: Specify the XPath and the Matching Value

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

Field	Value
Identification Value	12345
Identification Expression	//*[local-name() = 'TransactionID']/text()

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 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" ns1:caseCategoryID="1">
       <ns1:OfficialProvisionNumber>String</ns1:OfficialProvisionNumber>
    </ns:Case>
  </Body>
</Message>
```

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:

Field	Value
Identification Value	*Leave blank.*
Identification Expression	/*[local-name()='envelope']/body/transaction/command/*[local-name()='registerCommand']

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 xmins: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_EB3CDC749A1F2BABE03014906CC4605A</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>
```

Option 3: Check the Value of an Attribute

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

Field	Value
Identification Value	US
Identification Expression	//*/@country

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"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="PO.xsd">
  <name>B2B Buyer</name>
  <street>100 Oracle Parkway</street>
  <city>Redwood City</city>
  <state>CA</state>
  <zip>94065</zip>
</MyAddress>
```

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 7–3 Examples of EDI EDIFACT Transaction Sets Supported in Oracle B2B

Set	Description	Version
ORDERS	Purchase Order Message	D98A
ORDRSP	Purchase Order Response Message	D98A
CONTRL	Syntax and Service Report Message	D3

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.

Figure 7–5 Document Version Parameters for an EDI EDIFACT Document

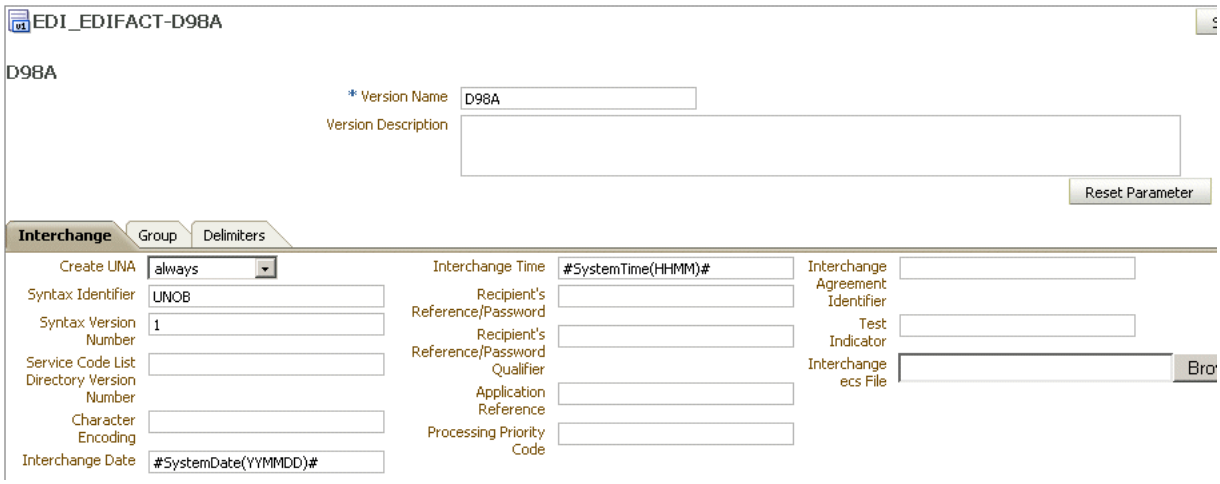


Table 7–4 describes the document version parameters for an EDI EDIFACT document.

Table 7–4 Document Version Parameters for an EDI EDIFACT Document

Parameter	Description
Interchange Tab	-
Create UNA	Select from **always**, **never**, or **delimiter-based**. If **delimiter-based** is selected, then UNA is created if the specified delimiters are different from the EDIFACT default value. The **Never** option does not generate UNA for outbound EDIFACT documents, even if nondefault delimiters are used. The **Never** option for inbound messages cannot work for B2B if an EDIFACT document is received without UNA and with nondefault delimiters.
Syntax Identifier	Coded identification of the agency controlling syntax and syntax level used in an interchange. EDI position UNB 010 010 S001 0001. The value **UNOB** is supplied.
Syntax Version Number	Version number of the syntax identified in the syntax identifier (0001). EDI position UNB 010 020 S001 0002. The value **1** is supplied.
Service Code List Directory Version Number	Version number of the service code list directory. EDI position UNB 010 030 S001 0030.
Character Encoding	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.
Interchange Date	Local date when an interchange or a group was prepared. EDI position UNB 030 010 S004 0017. The value **#SystemDate(YYMMDD)#** is supplied.
Interchange Time	Local time of day when an interchange or a group was prepared. EDI position UNB 030 020 S004 0019. The value **#SystemTime(HHMM)#** is supplied.
Recipient's Reference/Password	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.
Recipient's Reference/Password Qualifier	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.

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

Parameter	Description
Application Reference	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.
Processing Priority Code	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.
Interchange Agreement Identifier	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.
Test Indicator	Indication that the structural level containing the test indicator is a test. EDI position UNB 110.
Interchange ecs File	Use the **Browse** 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.
Group Tab	-
Create Functional Group	Indication of function group (UNG) creation. The value **TRUE** is supplied.
Date of Group Preparation	Local date when an interchange or a group was prepared. EDI position UNG 040 010. The system date stamp is supplied.
Time of Group Preparation	Local time of day when an interchange or a group was prepared. EDI position UNG 040 020. The system time stamp is supplied.
Controlling Agency	Code identifying a controlling agency. EDI position UNG 070 010. The value **UN** is supplied.
Group Association Assigned Code	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.
Application Password	Password to the recipient's division, department or sectional application system/process. EDI position UNG 080.
Group 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.
Delimiters Tab	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.
Segment Delimiter	EDIFACT segment delimiter. The value **0x27** is supplied.
Element Delimiter	EDIFACT element delimiter. The value **0x2b** is supplied.
Subelement Delimiter	EDIFACT subelement delimiter. The value **0x3a** is supplied.
Decimal Separator	EDIFACT decimal separator. The value **0x2e** is supplied.
Release Character	EDIFACT release character. The value **0x3f** is supplied.
Replacement Character	EDIFACT replacement character. The value **0x7c** is supplied.
Repeating Separator	EDIFACT repeating separator. The value **0x2a** is supplied.

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.

Figure 7–6 Document Type Parameters for an EDI EDIFACT Document

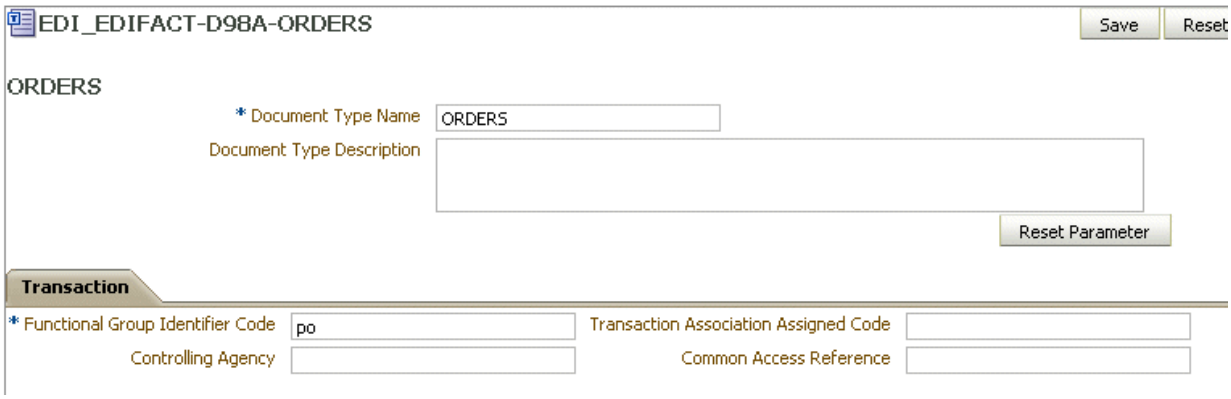


Table 7–5 describes the document type parameters for an EDI EDIFACT document.

Table 7–5 Document Type Parameters for an EDI EDIFACT Document

Parameter	Description
Transaction Tab	-
*Functional Group Identifier Code	Code identifying one type of message in a functional group. EDI position UNG 010 0038. Required.
Controlling Agency	Code identifying the agency controlling the specification, maintenance and publication of the message type. EDI position UNH 020 040 S009 0051.
Transaction Association Assigned Code	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 S009 0057.
Common Access Reference	Reference serving as a key to relate all subsequent transfers of data to the same business case or file. EDI position UNH 030 0068.

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.

Figure 7–7 Document Definition Parameters for an EDI EDIFACT Document

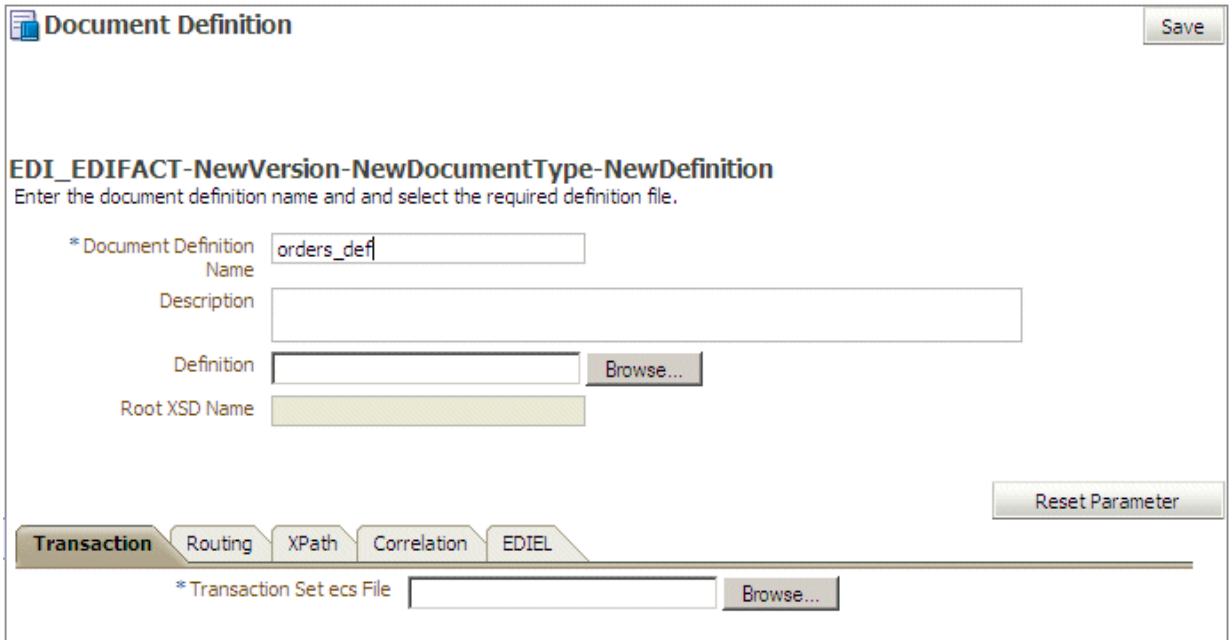


Table 7–6 describes the document definition parameters for an EDI EDIFACT document.

Table 7–6 Document Definition Parameters for an EDI EDIFACT Document

Parameter	Description
Transaction Tab	-
*Transaction Set ecs File	Use the **Browse** button to select the ecs file.
Routing Tab	-
Document Routing ID	Sets the consumer name to the back-end application
XPath Tab	See "How to Configure the XPath Expression for a Custom XML Document" on page 7-5 for more information.
XPath Name1	The XML XPath name for retrieving the value from the payload
XPath Expression1	The XML XPath expression for retrieving the value from the payload
XPath Name2	The XML XPath name for retrieving the value from the payload
XPath Expression2	The XML XPath expression for retrieving the value from the payload
XPath Name3	The XML XPath name for retrieving the value from the payload
XPath Expression3	The XML XPath expression for retrieving the value from the payload
Correlation Tab	-
Correlation From XPath Name	The name of the correlation property for initiating the correlation
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation
Correlation To XPath Name	The name of the correlation property for the correlation
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation
EDIEL Tab (preview)	-

Table 7–6 (Cont.) Document Definition Parameters for an EDI EDIFACT Document

Parameter	Description
FA Assoc Assigned Code	Code for the functional acknowledgment
FA Message Version Number	Version number for the functional acknowledgment
FA Message Release Number	Release number for the functional acknowledgment
Remove FA Segments	Remove functional acknowledgment segments

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 7–7 Examples of EDI X12 Transaction Sets Supported in Oracle B2B

Set	Description	Version
850	Purchase Order	4010
855	Purchase Order Acknowledgment	4010
997	Functional Acknowledgment	4010

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

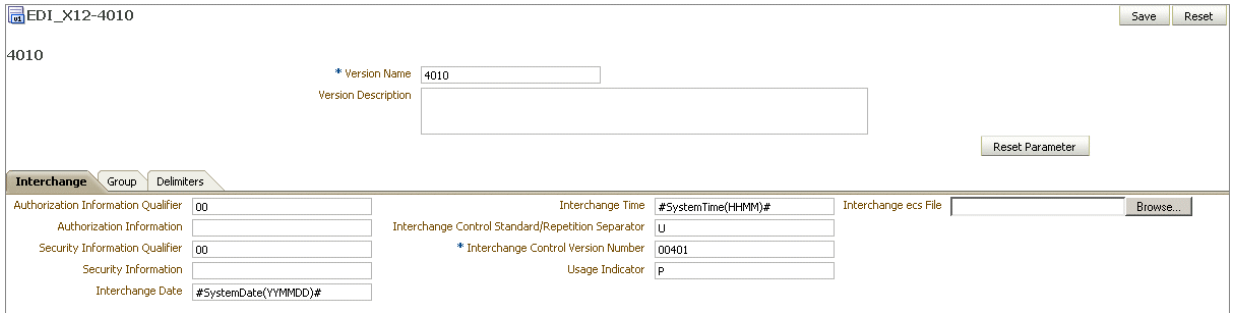


Table 7–8 describes the document version parameters for an EDI X12 document.

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

Parameter	Description
Interchange Tab	-
Authorization Information Qualifier	Code to identify the type of information in the authorization information. EDI position ISA 01. The value **00** is supplied.
Authorization Information	Information used for additional identification or authorization of the sender or the data in the interchange. The authorization information qualifier sets the type of information. EDI position ISA 02.
Security Information Qualifier	Code to identify the type of information in the security information. EDI position ISA 03. The value **00** is supplied.
Security Information	Information used to identify the security information about the interchange sender or the data in the interchange. The security information qualifier sets the type of information. EDI position ISA 04.
Interchange Date	Date of the interchange. EDI position ISA 09. The system date stamp is supplied (`#SystemDate(YYMMDD)#`).
Interchange Time	Time of the interchange. EDI position ISA 10.The system time stamp is supplied (`#SystemTime(HHMM)#`).
Interchange Control Standard/Repetition Separator	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 is ISA 11. The value **U** is supplied.
*Interchange Control Version Number	Code specifying the version number of the interchange control segments. EDI position ISA 12. The value **00401** is supplied.
Usage Indicator	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.
Interchange ecs File	Use the **Browse** 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.
Group Tab	-
Functional Group Date	Date sender generated a functional group of transaction sets. EDI position GS 04. The system date stamp is supplied (`#SystemDate(CCYYMMDD)#`).
Functional Group Time	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 (`#SystemTime(HHMM)#`).
Responsible Agency Code	Code used in conjunction with data element 480 to identify the issuer of the standard. EDI position GS 06. The value **X** is supplied.
Version/Release/Industry Identifier Code	Code indicating the version, release, subrelease, and industry identifier of the EDI standard being used, including the GS and GE segments; if 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 code in DE455 in GS segment is T, then other formats are allowed.
Group ecs File	Use the **Browse** 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.
Delimiters Tab	Note: Click **Select Hexadecimal Characters** next to any of the delimiter fields to provide values. See Table 7–4 for more about delimiters.
Segment Delimiter	The value **0x7e** is supplied.
Element Delimiter	The value **0x2a** is supplied.
Subelement Delimiter	The value **0x5c** is supplied.

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

Parameter	Description
Decimal Separator	The value **0x2e** is supplied.
Replacement Character	The value **0x7c** is supplied.
Repeating Separator	The value **0x5e** is supplied.

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.

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

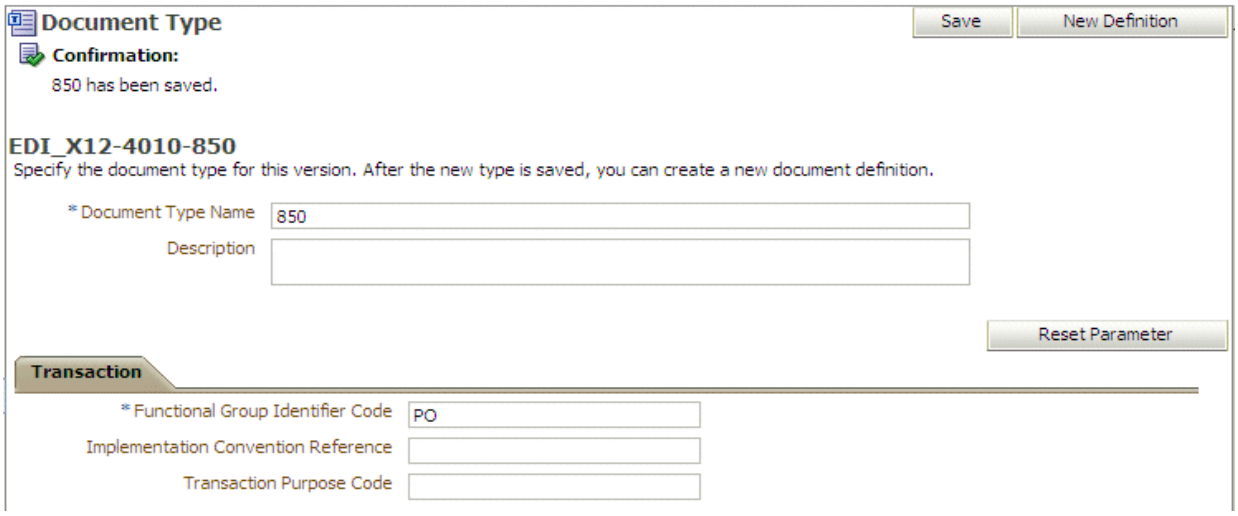


Table 7–9 describes the document type parameters for an EDI X12 document.

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

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

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.

Figure 7–10 Document Definition Parameters for an EDI X12 Document

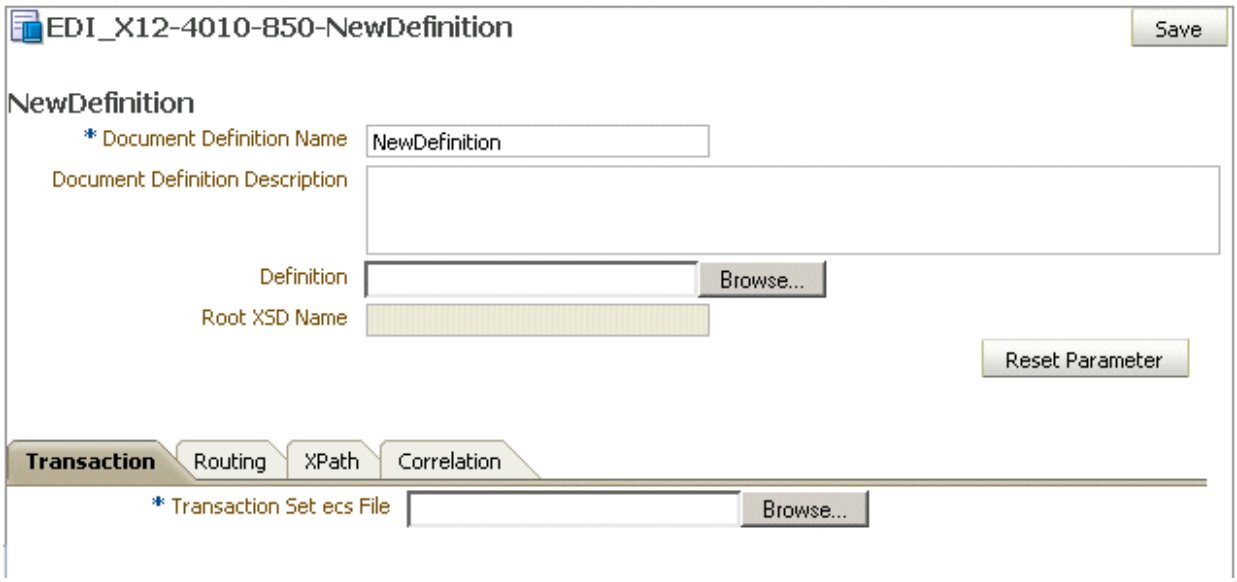


Table 7–10 describes the document definition parameters for an EDI X12 document.

Table 7–10 Document Definition Parameters for an EDI X12 Document

Parameter	Description
Transaction Tab	-
Transaction Set ecs File	Use the **Browse** button to select the ecs file.
Routing Tab	-
Document Routing ID	Sets the consumer name to the back-end application
XPath Tab	See "How to Configure the XPath Expression for a Custom XML Document" on page 7-5 for more information.
XPath Name1	The XML XPath name for retrieving the value from the payload
XPath Expression1	The XML XPath expression for retrieving the value from the payload
XPath Name2	The XML XPath name for retrieving the value from the payload
XPath Expression2	The XML XPath expression for retrieving the value from the payload
XPath Name3	The XML XPath name for retrieving the value from the payload
XPath Expression3	The XML XPath expression for retrieving the value from the payload
Correlation Tab	-
Correlation From XPath Name	The name of the correlation property for initiating the correlation.
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation.
Correlation To XPath Name	The name of the correlation property for the correlation.
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation.

Note: The `b2b.FAHandleByB2B` property can be set in `b2b-config.xml`. See "Setting b2b.FAHandleByB2B for EDI EDIFACT and EDI X12" on page B-2 for information about behavior and limitations when this property is set to false.

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

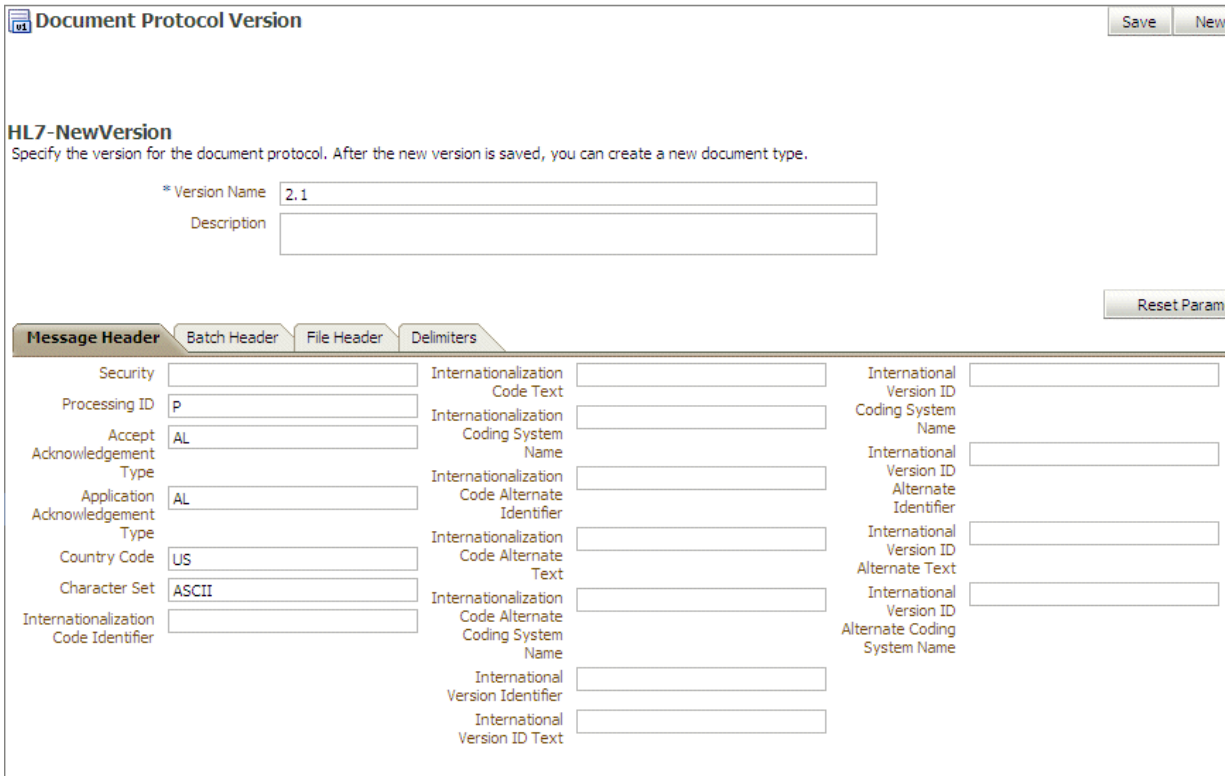


Table 7–11 describes the document version parameters for an HL7 document.

Table 7–11 Document Version Parameters for an HL7 Document

Parameter	Description
Message Header Tab	-
Security	In some applications of HL7, this field is used to implement security features.
Processing ID	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.
Accept Acknowledgement Type	Sets the conditions under which application acknowledgments are required to be returned in response to the message. The value **AL** (always) is supplied. 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.
Application Acknowledgment Type	MSH.16. The value **AL** (always) is supplied.
Country Code	Sets the country of origin for the message. The value US is supplied.
Character Set	Sets the character set for the entire message. The value ASCII is supplied.
Internalization Code Identifier	MSH.19
Internalization Code Text	MSH.19
Internationalization Coding System Name	MSH.19
Internationalization Code Alternate Identifier	MSH.19
Internationalization Code Alternate Text	MSH.19
Internationalization Code Alternate Coding System Name	MSH.19
International Version Identifier	MSH.12
International Version ID Text	MSH.12
International Version ID Coding System Name	MSH.12
International Version ID Alternate Identifier	MSH.12
International Version ID Alternate Text	MSH.12
International Version ID Alternate Coding System Name	MSH.12
Batch Header Tab	-
Create Batch Header	Check the box to create batch headers.

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

Parameter	Description
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)#`).
Delimiters Tab	Note: Click **Select Hexadecimal Characters** next to any of the delimiter fields to provide values. See Table 7–4 for more about delimiters.
Element Delimiter	A single character that follows the segment identifier and separates each data element in a segment except the last. The value **0x7c** is supplied.
Escape Character	The value **0x5c** is supplied.
Repeating Separator	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.
Segment Delimiter	A syntax character indicating the end of a segment (a logical grouping of data fields) within a message. The value **0x0d** is supplied.
Subcomponent Delimiter	The value 0x26 is supplied.
Subelement Delimiter	The value 0x5e is supplied.

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

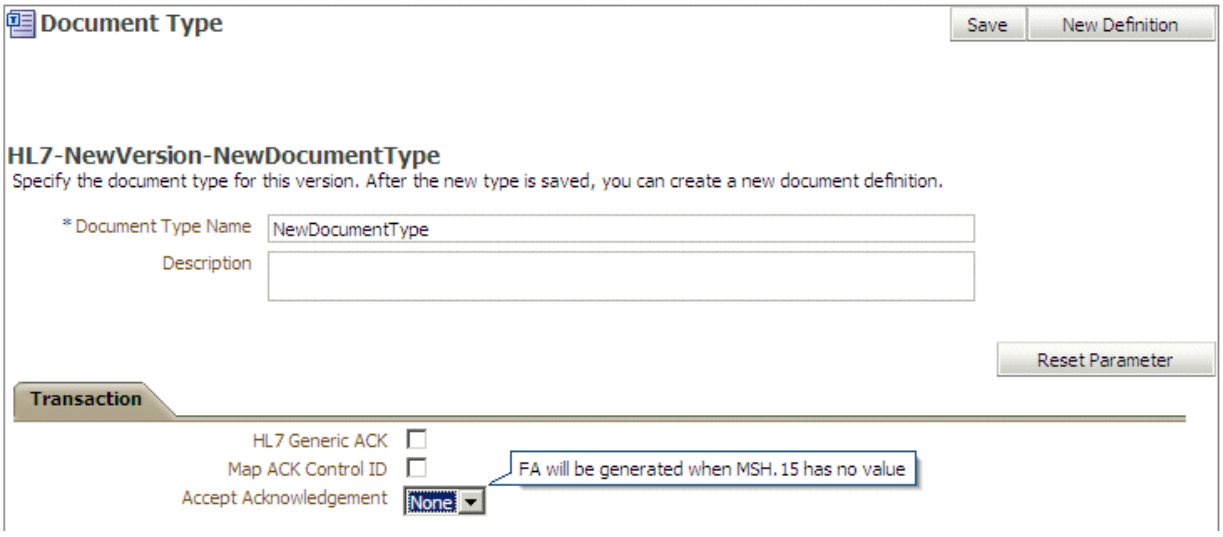


Table 7–12 describes the document type parameters for an HL7 document.

Table 7–12 Document Type Parameters for an HL7 Document

Parameter	Description
Transaction Tab	-
HL7 Generic ACK	Oracle B2B can send an generic ACK immediately upon receiving an HL7 message
Map ACK Control ID	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.
Accept Acknowledgement	A functional acknowledgment is generated when MSH.15 has no value. Select **None** to take no action. Acknowledgment generation is dependent on the value in MSH.15 of the business message. Select **AL** (always) to generate the acknowledgment under any conditions. Select **ER** (error/reject) to generate the acknowledgment when the message errors or is rejected. Select **SU** (successful completion) to generate the acknowledgment when the message is successfully processed.

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.

Figure 7–13 Document Definition Parameters for an HL7 Document

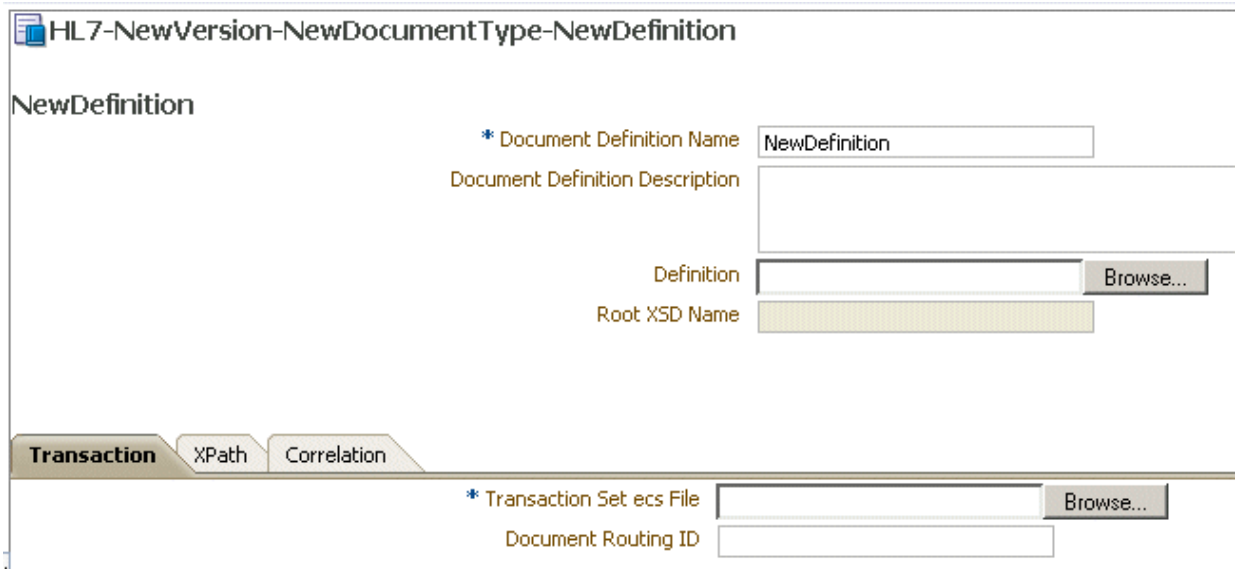


Table 7–13 describes the document definition parameters for an HL7 document.

Table 7–13 Document Definition Parameters for an HL7 Document

Parameter	Description
Transaction Tab	-
*Transaction Set ecs File	Use the **Browse** button to find the ecs file.
Document Routing ID	Sets the consumer name to the back-end application
XPath Tab	See "How to Configure the XPath Expression for a Custom XML Document" on page 7-5 for more information.
XPath Name1	The XML XPath name for retrieving the value from the payload

Table 7–13 (Cont.) Document Definition Parameters for an HL7 Document

Parameter	Description
XPath Expression1	The XML XPath expression for retrieving the value from the payload
XPath Name2	The XML XPath name for retrieving the value from the payload
XPath Expression2	The XML XPath expression for retrieving the value from the payload
XPath Name3	The XML XPath name for retrieving the value from the payload
XPath Expression3	The XML XPath expression for retrieving the value from the payload
Correlation Tab	-
Correlation From XPath Name	The name of the correlation property for initiating the correlation
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation
Correlation To XPath Name	The name of the correlation property for the correlation
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation

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.

Using the OAG Document Protocol

Oracle B2B implements Open Applications Group (OAG) standards, a robust XML standard used across many industries. This standard defines messages as business object documents (BODs). This document protocol is in preview mode for this release.

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

http://www.oagi.org

Document Version Parameters

No parameters need to be set when you create the document version for an OAG document.

Document Type Parameters

No parameters need to be set when you create the document type for an OAG document.

Document Definition Parameters

When you create an OAG document definition, you can set various parameters. Figure 7–14 shows document definition parameters for an OAG document.

Figure 7–14 Document Definition Parameters for an OAG Document

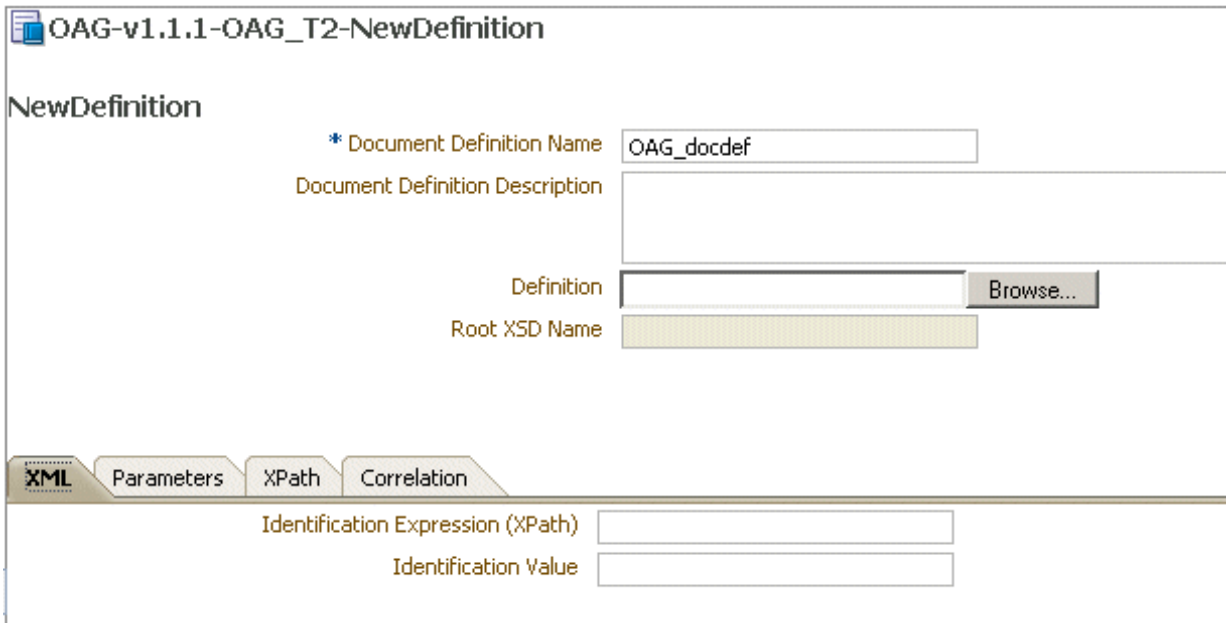


Table 7–14 describes the document definition parameters for an OAG document.

Table 7–14 Document Definition Parameters for an OAG Document

Parameter	Description
XML Tab	-
Identification Expression (XPath)	Locates a node in the XML payload
Identification Value	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.
Routing Tab	-
Document Routing ID	Sets the consumer name to the back-end application
XPath Tab	See "How to Configure the XPath Expression for a Custom XML Document" on page 7-5 for more information.
XPath Name1	The XML XPath name for retrieving the value from the payload
XPath Expression1	The XML XPath expression for retrieving the value from the payload
XPath Name2	The XML XPath name for retrieving the value from the payload
XPath Expression2	The XML XPath expression for retrieving the value from the payload
XPath Name3	The XML XPath name for retrieving the value from the payload
XPath Expression3	The XML XPath expression for retrieving the value from the payload
Correlation Tab	-

Table 7–14 (Cont.) Document Definition Parameters for an OAG Document

Parameter	Description
Correlation From XPath Name	The name of the correlation property for initiating the correlation
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation
Correlation To XPath Name	The name of the correlation property for the correlation
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation

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. This document protocol is in preview mode for this release.

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 7–15 shows document definition parameters for a positional flat file.

Figure 7–15 Document Definition Parameters for a Positional Flat File (including SAP iDocs)

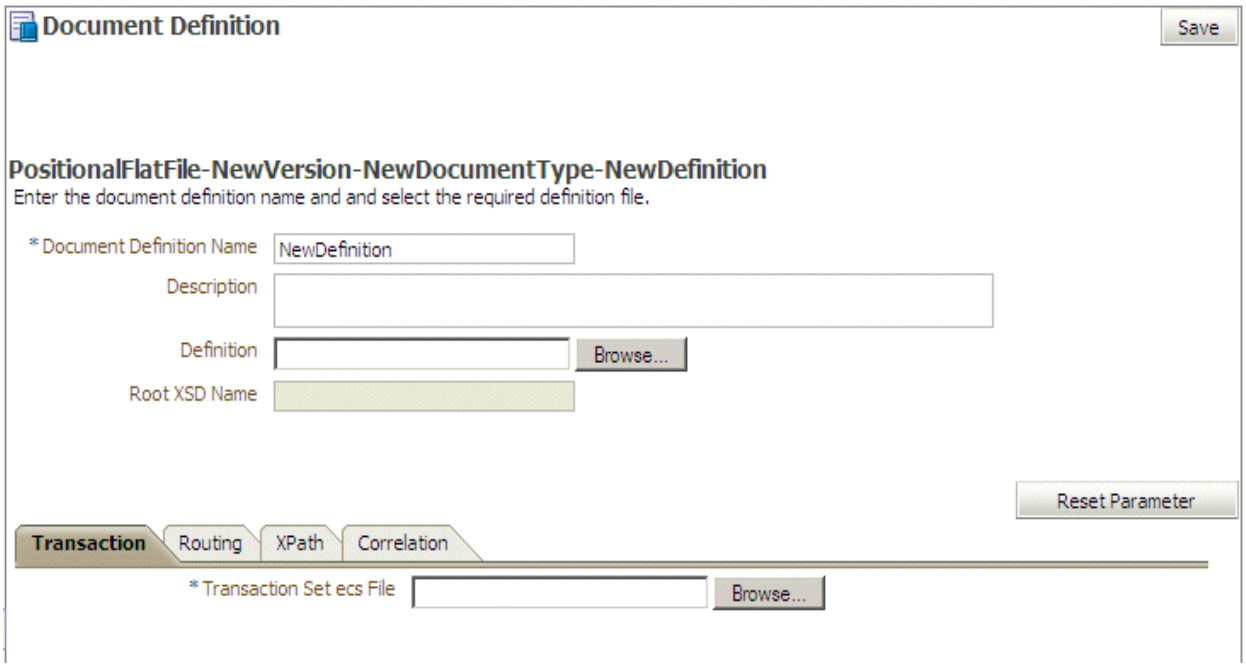


Table 7–15 describes the document definition parameters for a positional flat file.

Table 7–15 Document Definition Parameters for a Positional Flat File

Parameter	Description
Parameters Tab	-
*Transaction Set ecs File	Use the **Browse** button to find the ecs file.
Routing Tab	-
Identification Expression (XPath)	Not applicable
Identification Value	Not applicable
Identification Start Position	Used in combination with the end position to retrieve a value from the payload between the start and end positions
Identification End Position	Used in combination with the start position to retrieve a value from the payload between the start and end positions
Document Routing ID	Sets the consumer name to the back-end application
XPath Tab	See "How to Configure the XPath Expression for a Custom XML Document" on page 7-5 for more information.
XPath Name1	The XML XPath name for retrieving the value from the payload
XPath Expression1	The XML XPath expression for retrieving the value from the payload
XPath Name2	The XML XPath name for retrieving the value from the payload
XPath Expression2	The XML XPath expression for retrieving the value from the payload
XPath Name3	The XML XPath name for retrieving the value from the payload
XPath Expression3	The XML XPath expression for retrieving the value from the payload
Correlation Tab	-

Table 7–15 (Cont.) Document Definition Parameters for a Positional Flat File

Parameter	Description
Correlation From XPath Name	The name of the correlation property for initiating the correlation
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation
Correlation To XPath Name	The name of the correlation property for the correlation
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation

Using the RosettaNet Document Protocol

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 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.

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

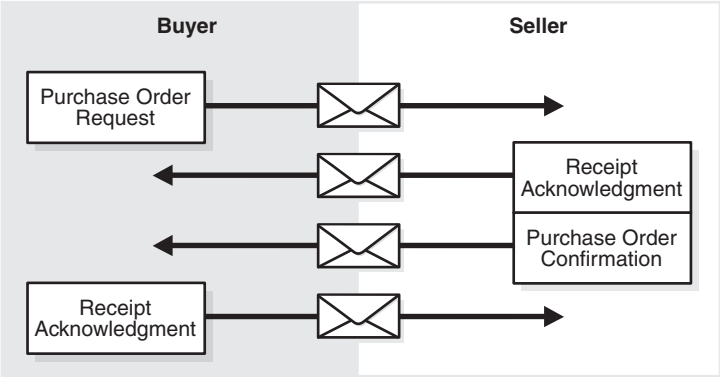
`http://www.rosettanet.org`

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 7–16.

Figure 7–16 PIP 3A4 Message Exchange Between Buyer and Seller



A PIP sequence combines a cluster, segment, and type. The PIP sequence 3A4, for example, encodes the information shown in Table 7–16.

Table 7–16 PIP 3A4 Breakdown

Element	Description
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 7–17 shows document type parameters for a RosettaNet document.

Figure 7–17 Document Type Parameters for a RosettaNet Document

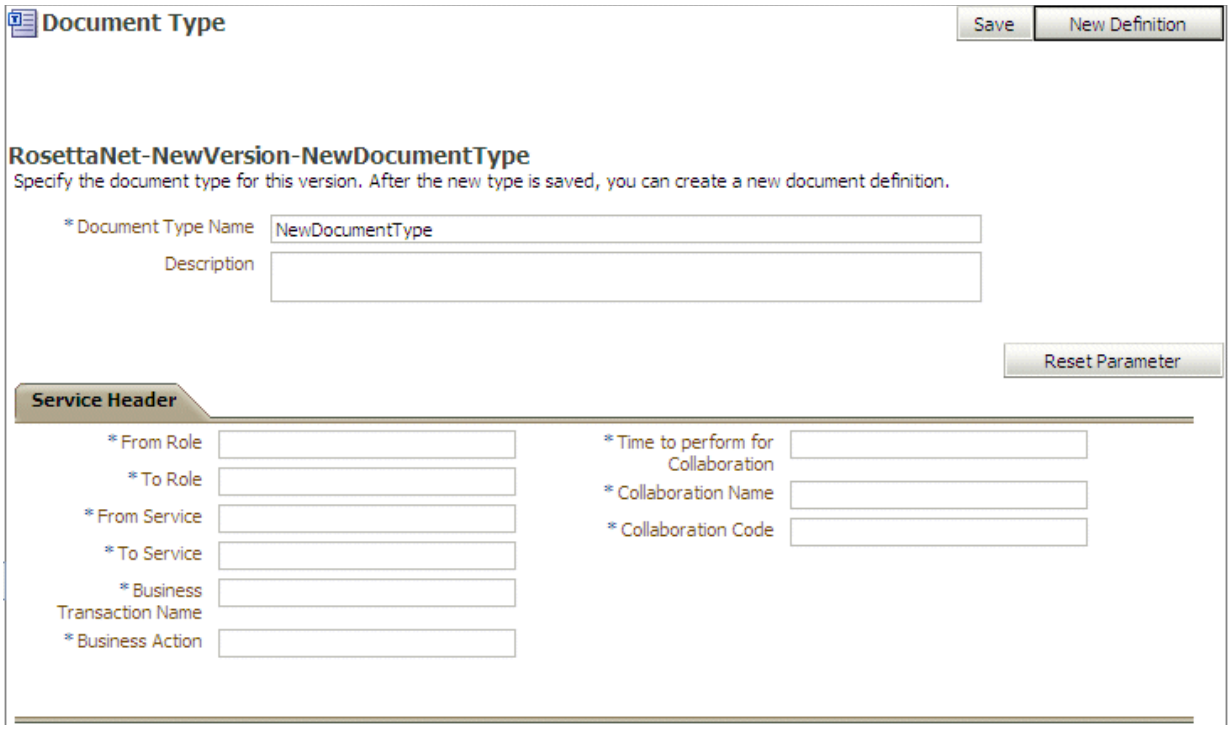


Table 7–17 describes document type parameters for a RosettaNet document.

Table 7–17 Document Type Parameters for a RosettaNet Document

Parameter	Description
Service Header Tab	-
*From Role	The trading partner that sends the message (in Partner Role Description of the PIP).
*To Role	The trading partner that receives the message (the role the trading partner receiving the message plays in the PIP).
*From Service	The service that sends the message.
*To Service	The service to which the message is sent.
*Business Transaction Name	The name of the business transaction is required.
*Business Action	The name of the business action is required. The value must be consistent with the Global Business Action Code.
*Time to Perform for Collaboration	The time to perform the business action is required.
*Collaboration Name	A name for the set of roles (buyer and seller) collaborating through a set of agreed-on business transactions by exchanging business documents. Required.
*Collaboration Code	The collaboration code is required.

Document Definition Parameters

When you create a RosettaNet document definition, you can set various parameters. Figure 7–18 shows the document definition parameters for a RosettaNet document.

Figure 7–18 Document Definition Parameters for a RosettaNet Document

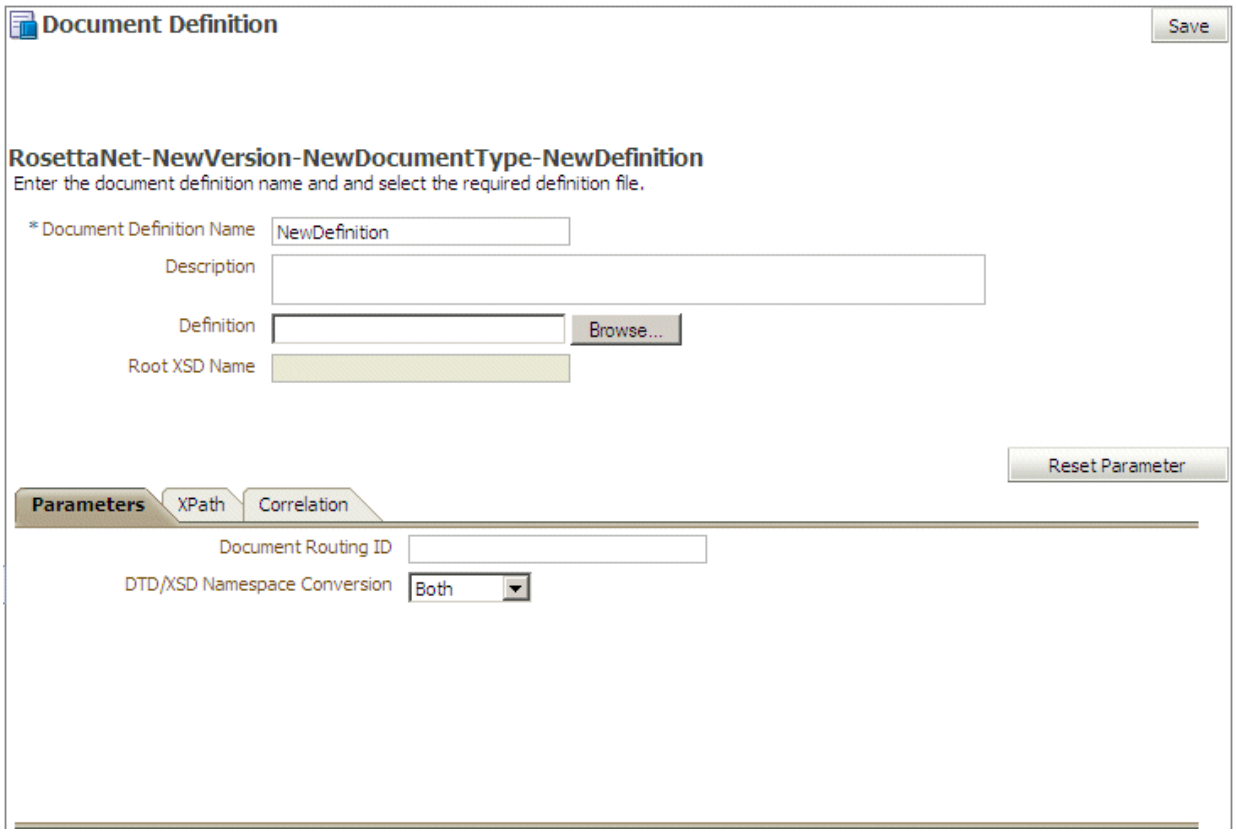


Table 7–18 describes the document definition parameters for a RosettaNet document.

Table 7–18 Document Definition Parameters for a RosettaNet Document

Parameter	Description
Parameters Tab	-
Document Routing ID	Sets the consumer name to the back-end application
DTD/XSD Namespace	A converted document can optionally replace the original RosettaNet document. Select **Both** to replace the RosettaNet document with the converted document for both the inbound and outbound messages. Select **Inbound** to replace the RosettaNet document with the converted document for the inbound message. Select **Outbound** to replace the RosettaNet document with the converted document for the outbound message. Select **None** for no replacement. None passes the DTD instance as-is. Inbound converts the instance DTD to XSD. Outbound converts the instance XSD to DTD. Both convert both inbound and outbound formats.
XPath Tab	See "How to Configure the XPath Expression for a Custom XML Document" on page 7-5
XPath Name1	The XML XPath name for retrieving the value from the payload
XPath Expression1	The XML XPath expression for retrieving the value from the payload
XPath Name2	The XML XPath name for retrieving the value from the payload
XPath Expression2	The XML XPath expression for retrieving the value from the payload
XPath Name3	The XML XPath name for retrieving the value from the payload
XPath Expression3	The XML XPath expression for retrieving the value from the payload
Correlation Tab	Correlation is required for a two-action PIP, for example, a 3A4

Table 7–18 (Cont.) Document Definition Parameters for a RosettaNet Document

Parameter	Description
Correlation From XPath Name	The name of the correlation property for initiating the correlation. For example, Pip3A4PurchaseOrderRequest in /*[local-name()='Pip3A4PurchaseOrderRequest']/*[local-name()='thisDocumentIdentifier']/text()
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation.
Correlation To XPath Name	The name of the correlation property for the correlation. Correlation-to represents the other message that takes part in the correlation. For example, Pip3A4PurchaseOrderConfirmation in/*[local-name()='Pip3A4PurchaseOrderConfirmation']/*[local-name()='requestingDocumentIdentifier']/text()
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation.

RosettaNet Dictionaries

The RosettaNet Business Dictionary provides a common vocabulary and a common set of properties to use in XML documents. For example, trading partners using the RosettaNet Business Dictionary might agree to use the term DRAM for memory chip. The RosettaNet Technical Dictionary is not supported in Oracle B2B.

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.

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. This document protocol is in preview mode for this release. Table 7–19 lists the UCCnet document types supported in Oracle B2B.

Table 7–19 UCCnet Document Types

Standard
registerCommand
confirmCommand
linkCommand
checkComplianceCommand
documentCommand
documentIdentificationCommand
notificationStateCommand
queryCommand
registerLinkCommand
publicationCommand
publishCommand
catalogueItemMaintenanceCommand
priceCommand
validateCommand
registerOwnershipCommand
subscriptionCommand
notifyCommand
response

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

`http://www.1sync.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 7–19 shows document definition parameters for a UCCnet document.

Figure 7–19 Document Definition Parameters for a UCCnet Document

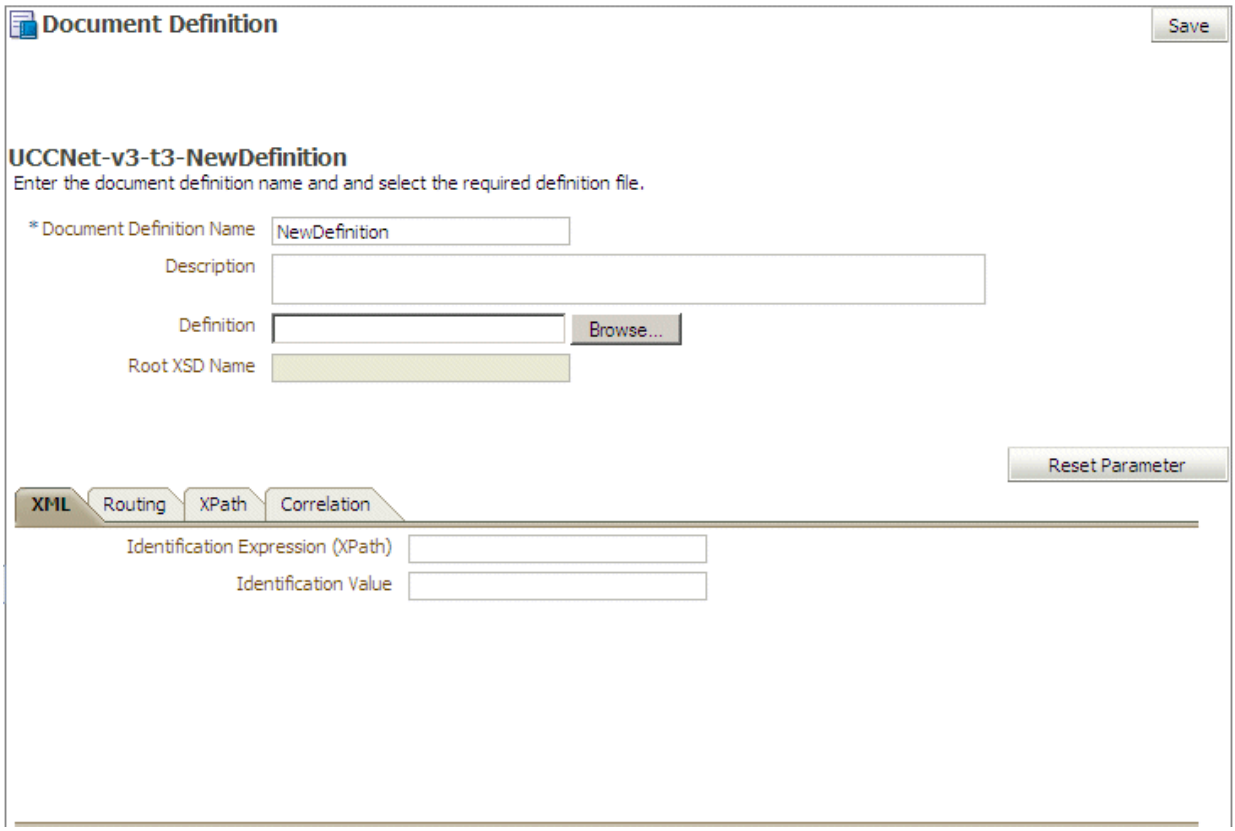


Table 7–20 describes the document definition parameters for a UCCnet document.

Table 7–20 Document Definition Parameters for a UCCnet Document

Parameter	Description
XML Tab	-
Identification Expression (XPath)	Locates a node in the XML payload
Identification Value	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.
Routing Tab	-
Document Routing ID	Sets the consumer name to the back-end application
XPath Tab	See "How to Configure the XPath Expression for a Custom XML Document" on page 7-5
XPath Name1	The XML XPath name for retrieving the value from the payload
XPath Expression1	The XML XPath expression for retrieving the value from the payload
XPath Name2	The XML XPath name for retrieving the value from the payload
XPath Expression2	The XML XPath expression for retrieving the value from the payload
XPath Name3	The XML XPath name for retrieving the value from the payload
XPath Expression3	The XML XPath expression for retrieving the value from the payload

Table 7–20 (Cont.) Document Definition Parameters for a UCCnet Document

Parameter	Description
Correlation Tab	-
Correlation From XPath Name	The name of the correlation property for initiating the correlation
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation
Correlation To XPath Name	The name of the correlation property for the correlation
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation

Summary of Document Protocol Parameter Types

Table 7–21 summarizes the types of document parameters available for each document protocol.

Table 7–21 Types of Document Protocol Parameters Available in Oracle B2B

Protocol	Document Version Parameters	Document Type Parameters	Document Definition Parameters
Custom	none	ebMS (Table 7–1)	XML (Table 7–2) Flat (Table 7–2) Routing (Table 7–2) XPath (Table 7–2) Correlation (Table 7–2)
EDI EDIFACT	Interchange (Table 7–4) Group (Table 7–4) Delimiters (Table 7–4)	Transaction (Table 7–5)	Transaction (Table 7–6) Routing (Table 7–6) XPath (Table 7–6) Correlation (Table 7–6) EDIEL (Table 7–6)
EDI X12	Interchange (Table 7–8) Group (Table 7–8 Delimiters (Table 7–8)	Transaction (Table 7–9)	Transaction (Table 7–10) Routing (Table 7–10) XPath (Table 7–10) Correlation (Table 7–10)
HL7	Message header (Table 7–11) Batch header (Table 7–11) File header (Table 7–11) Delimiters (Table 7–11)	Transaction (Table 7–12)	Transaction (Table 7–13) XPath (Table 7–13) Correlation (Table 7–13)
OAG	none	none	XML (Table 7–14) Routing (Table 7–14) XPath (Table 7–14) Correlation (Table 7–14)

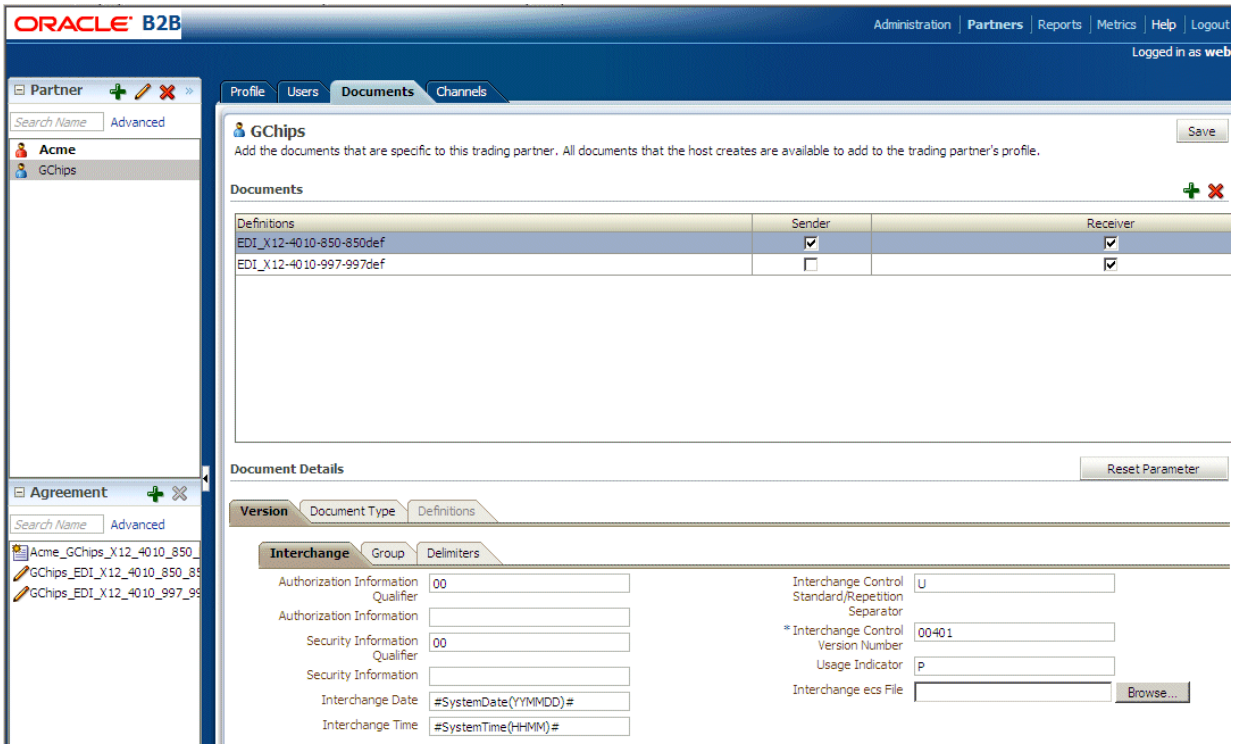
Table 7–21 (Cont.) Types of Document Protocol Parameters Available in Oracle B2B

Protocol	Document Version Parameters	Document Type Parameters	Document Definition Parameters
Positional flat file	none	none	Parameters (Table 7–15) Routing (Table 7–15) XPath (Table 7–15) Correlation (Table 7–15)
RosettaNet	none	Service header (Table 7–17)	Parameters (Table 7–18) XPath (Table 7–18) Correlation (Table 7–18)
UCCnet	none	none	XML (Table 7–20) Routing (Table 7–20) XPath (Table 7–20) Correlation (Table 7–20)

Changing Document Definitions

Document details—document protocol versions and document type parameters—can be changed for a remote trading partner from the **Partners** > **Documents** tab, as shown in Figure 7–20. Host administrators can change any document details here, and remote administrators can change document details for their own data.

Figure 7–20 Changing Document Details



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.

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.

Changing Document Definitions After Importing Metadata

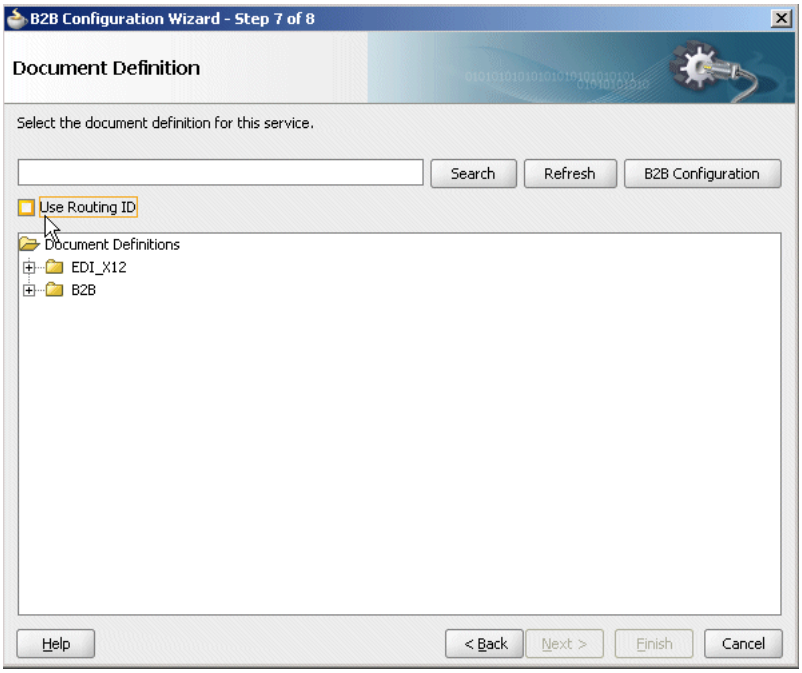
If you import B2B metadata and then change the document from the **Administration** > **Document** tab, then you must also make the same changes to the supported document definition for the host and remote trading partners from the **Partners** > **Documents** tab. Use the **Version**, **Document Type**,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

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 7–21.

Figure 7–21 Document Routing ID Option in Oracle JDeveloper



When using AQ, if you set the routing ID value instead of using the default b2buser, do not set it to a numeric value. Use a combination of alphabetic and numeric values.

8

Managing Deployments

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:

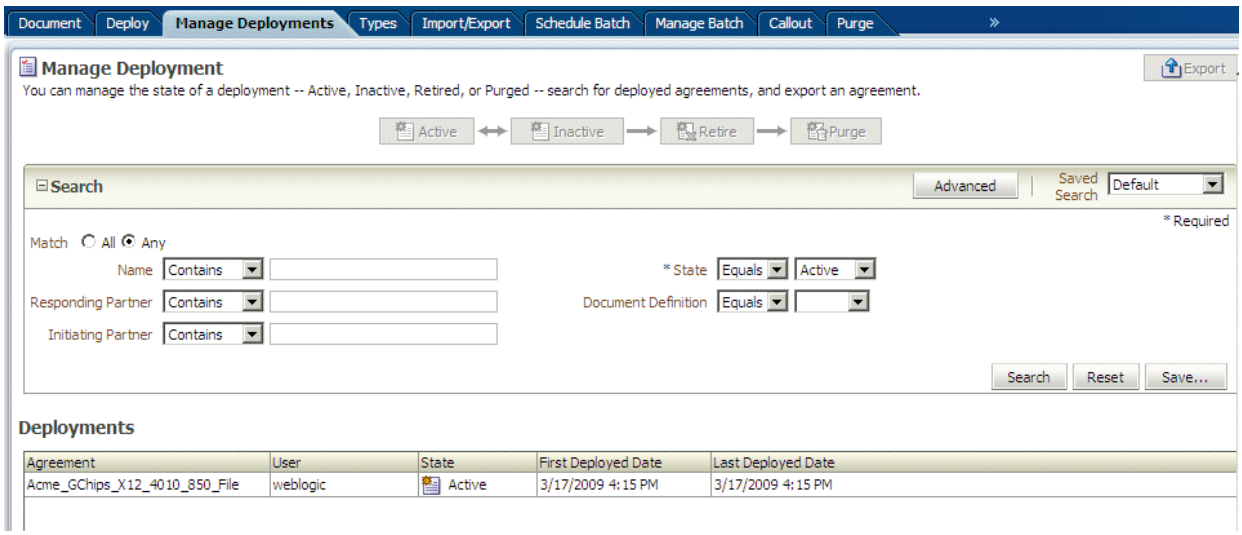
- Introduction to Agreement Deployment States
- Managing Deployed Agreements

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

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



Managing Deployed Agreements

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

Table 8–1 Deployed Agreement States

State	Description	When to Use
Active	The agreement has been successfully deployed and is ready to process messages. From an Active state, a deployed agreement can move to an Inactive state only.	When you are ready to receive or send messages using the agreement.
Inactive	The agreement can be changed to Active or Retired states. The agreement will not accept any new messages. However, all in-flight messages will be processed successfully. From an Inactive state, a deployed agreement can be moved to a Retired state or can be moved back to an Active state.	When a newer version of the same agreement is made Active, the previous version is changed to the Inactive state automatically. Also, when you do not want to receive new messages, but want to continue the in-flight messages, you can change the agreement to Inactive.
Retired	The agreement cannot be redeployed. No messages will be processed. From a Retired state, a deployed agreement can be purged only.	When you no longer want to receive or send messages using this agreement
Purged	The agreement is deleted from the system.	When you want to clean up unused agreements. Differs from Retired agreements, where you can still see the agreement in the system for information purposes.

Searching for Deployed Agreements

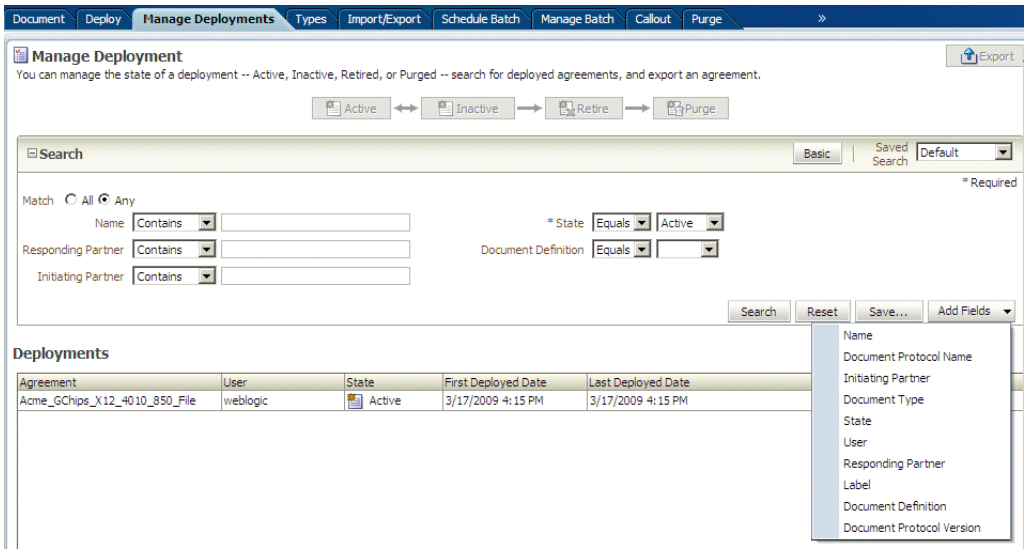
Use the search parameters described in Table 8–2 to search for deployed agreements.

Table 8–2 Search Parameters for Searching on Deployed Agreements

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

- Click **Reset** to return the search parameters shown in Table 8–2 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.

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.

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 keystore password and update the keystore location (if needed) for the host trading partner in the B2B interface. If the export file is imported back or the keystore and its location have not changed on the target computer, then the keystore password and location may be identical to the first keystore and keystore 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.

9

Creating Types

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:

- Creating Custom Identifier Types
- Creating Custom Contact Information Types
- Creating Custom Trading Partner Parameter Types

See "Creating Trading Partner Profiles" on page 5-2 for information on *adding* custom types and values to a trading partner profile.

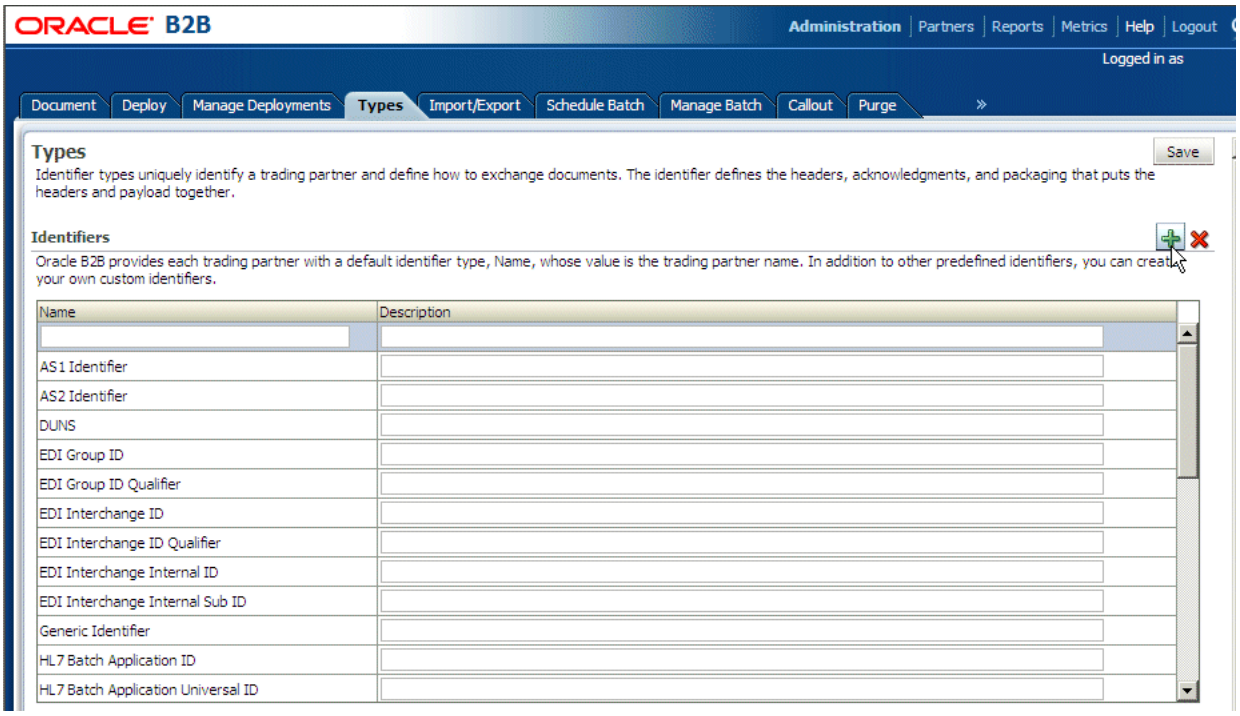
Creating Custom Identifier Types

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.



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 deleted 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

Name	Description
AS1 Identifier (Preview mode for this release)	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.
AS2 Identifier	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.
DUNS	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.
EDI Group ID	Used to identify multiple branches within a trading partner's company. The group ID can be the same as the interchange ID.
EDI Group ID Qualifier	Used to specify the function of the EDI Group ID.
EDI Interchange ID	A unique identifier for a trading partner that can come from different sources. For example, if the trading partner has a Dun & Bradstreet number, that number can be used for the interchange ID. In most cases, the selected VAN assigns the interchange ID.

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

Name	Description
EDI Interchange ID Qualifier	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.
EDI Interchange Internal ID	Identifies the trading partner based on the EDI interchange internal ID.
EDI Interchange Internal Sub ID	Identifies the trading partner based on the EDI interchange internal sub-ID.
Generic Identifier	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.
HL7 Batch Application ID	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.
HL7 Batch Application Universal ID	Identifies the trading partner based on the HL7 batch application universal ID.
HL7 Batch Application Universal ID Type	Identifies the trading partner based on the HL7 batch application universal ID type.
HL7 Batch Facility ID	Identifies the trading partner based on the HL7 batch facility ID.
HL7 Batch Facility Universal ID	Identifies the trading partner based on the HL7 batch facility universal ID.
HL7 Batch Facility Universal ID Type	Identifies the trading partner based on the HL7 batch facility universal ID type.
HL7 File Application ID	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.
HL7 File Application Universal ID	Identifies the trading partner based on the HL7 file application universal ID.
HL7 File Application Universal ID Type	Identifies the trading partner based on the HL7 file application universal ID type.
HL7 File Facility ID	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.
HL7 File Facility Universal ID	Identifies the trading partner based on the HL7 file facility universal ID.
HL7 File Facility Universal ID Type	Identifies the trading partner based on the HL7 file facility universal ID type.
HL7 Message Application ID	Identifies the sending/receiving application.
HL7 Message Application Universal ID	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.
HL7 Message Application Universal ID Type	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.
HL7 Message Facility ID	Identifies the trading partner based on the HL7 message facility ID.

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

Name	Description
HL7 Message Facility Universal ID	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.
HL7 Message Facility Universal ID Type	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.
MLLP ID	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.
Name	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.
ebMS Identifier	This type, OASIS ebXML Messaging Services (ebXML), specifies a secure and reliable way to exchange messages using HTTP, HTTPS, SOAP, XMLDsig, and XMLEncrypt. The ebMS Identifier and the Name identifier are required for ebMS exchanges.

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.

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.

10

Importing and Exporting Data

For design-time data, use the Oracle B2B interface to import and export B2B repositories.

This chapter contains the following topics:

- Importing and Exporting the Design-Time Repository
- What Is Copied When You Import or Export from the Import/Export Tab
- About the Exported File

See Chapter 18, "B2B Command Line Tools," for information on importing and exporting data using `ant`.

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.

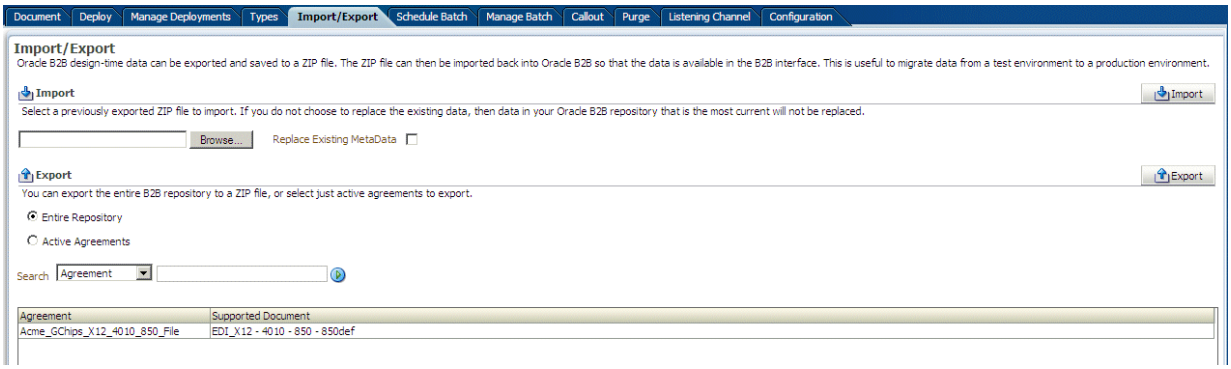
You can exporting data from other areas of the Oracle B2B interface also:

- Click **Partners** > **Profile** to export trading partner data. See "Creating Trading Partner Profiles" on page 5-2 for more information.
- Click **Partners** and then an agreement to export the agreement. See "Deleting and Exporting Agreements" on page 6-6 for more information.
- Click **Administration** > **Manage Deployments** to export deployed agreements. See "Exporting an Active Agreement" on page 8-3 for more information.

You can also import sample files that use the following document types: Custom, EDI EDIFACT, EDI X12, HL7, and RosettaNet. See "Oracle B2B Samples" on page 1-8 for the download location and information about the scenarios presented in the samples.

Figure 10–1 shows 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**.

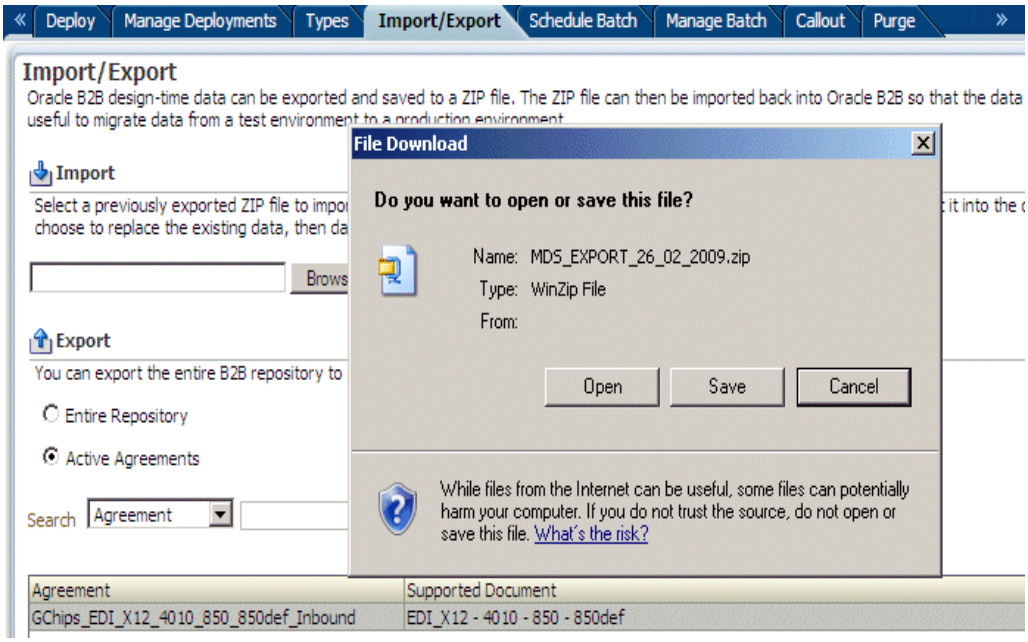
   The entire repository includes all data in the B2B design-time repository—agreements in all states, all trading partner configurations, and so on.

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** or **Save**.

   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.



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

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 is not copied when you export a repository. Use the command line utility to export user data. See "Exporting Data" on page 18-3 for more information. 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.

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.

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.

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:

- Setting Up a Batch
- Managing Batched Messages

See the following for more information about EDI:

- "Using the EDI EDIFACT Document Protocol" on page 7-7
- "Using the EDI X12 Document Protocol" on page 7-12

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.

Figure 11–1 shows where you set up a batch transmission of EDI messages.

Figure 11–1 Scheduling a Batch

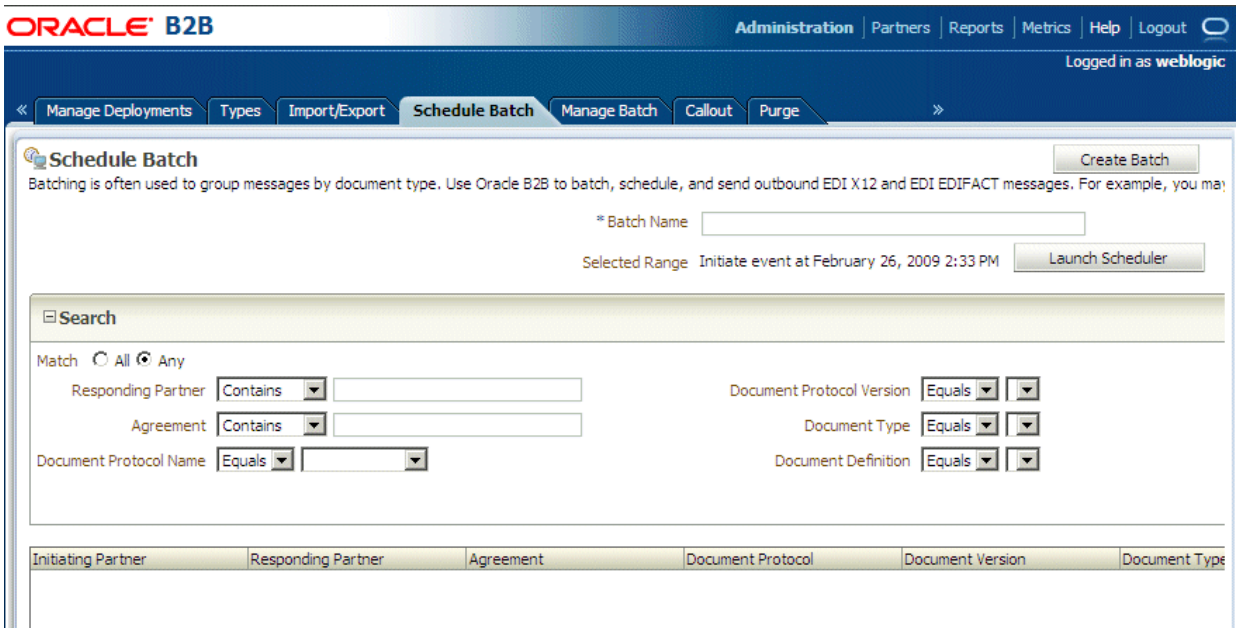
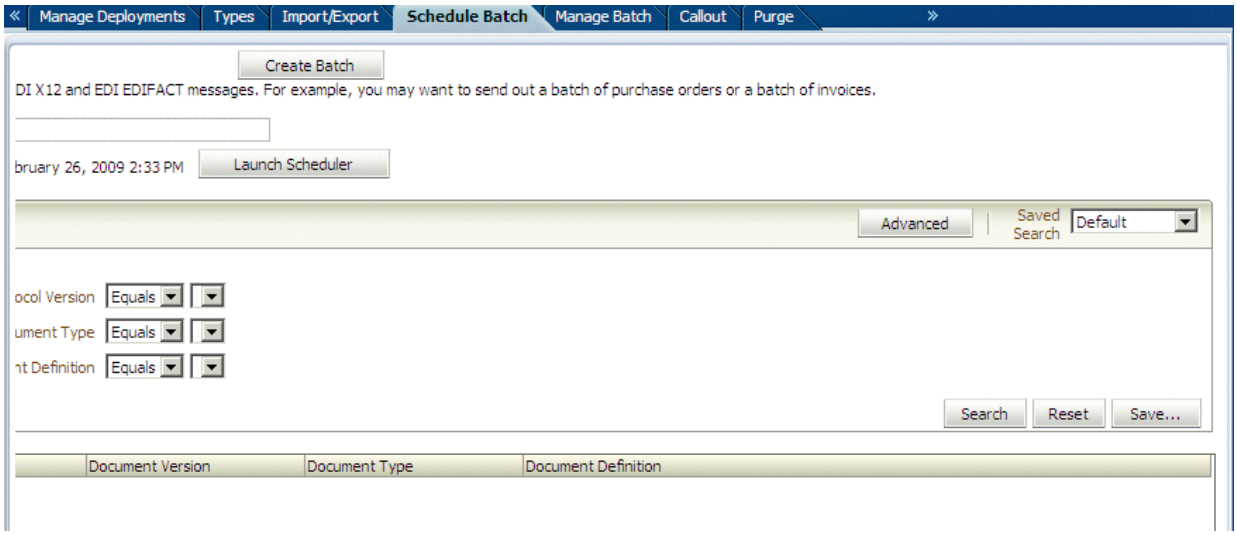


Figure 11–2 (right side of the **Schedule Batch** tab) shows where you can do an advanced search for agreements.

Figure 11–2 Scheduling a Batch—Advanced Search



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

Parameter	Description
Match All or Any	If you select **All**, then fields with values are matched using an *and* condition. If you select **Any**, then fields with values are matched using an *or* condition.
Responding Partner	Select **Starts With**, **Contains**, **Equals**, or **Ends With**, and type the appropriate portion of the name of the responding trading partner.
Agreement	Select **Starts With**, **Contains**, **Equals**, or **Ends With**, and type the appropriate portion of the name of the agreement.
Document Protocol Name	Select **EDI_EDIFACT** or **EDI_X12**.
Document Protocol Version	Select a document protocol version that you previously created.
Document Type	Select a document type that you previously created.
Document Definition	Select a document definition that you previously created.

4. Click **Search**.

   Active, deployed agreements (outbound) that meet your search criteria are displayed.

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

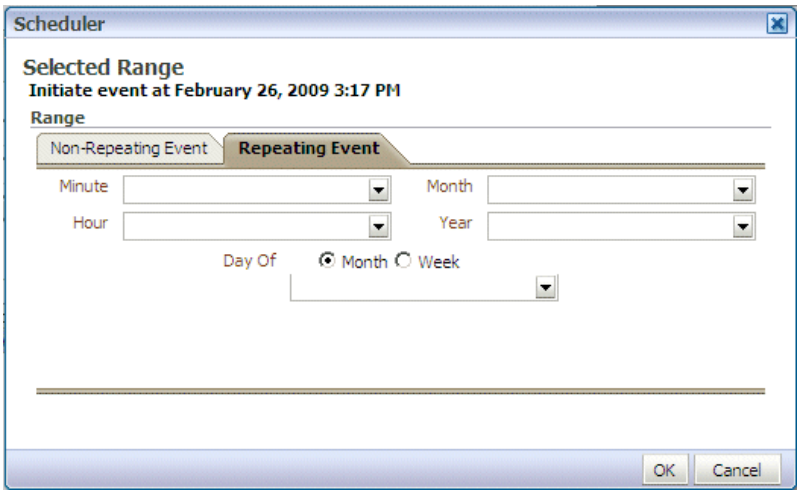
Task 2 Create the Batch

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, enter details on the interval to trigger the event, by specifying the minutes, hour, month, year, and date details. Then click **OK**.

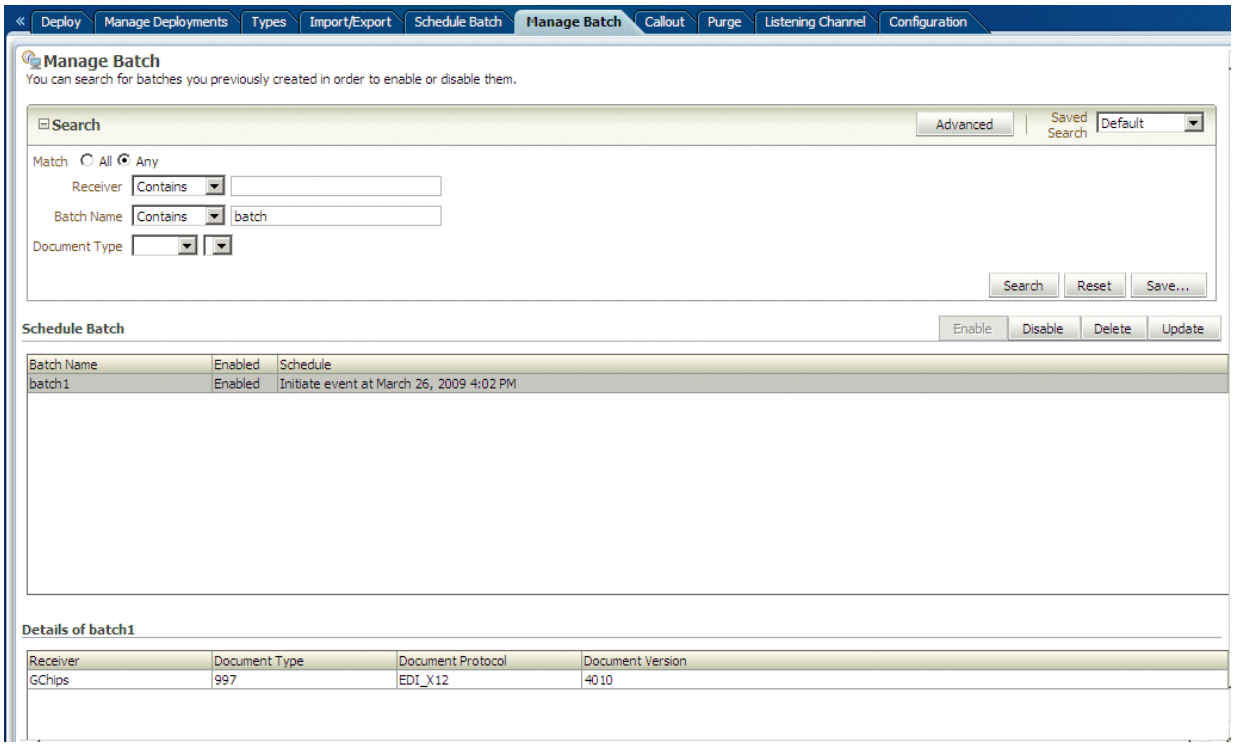


You can see the batches you create on the **Manage Batch** tab.

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.

12

Managing Callouts

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.

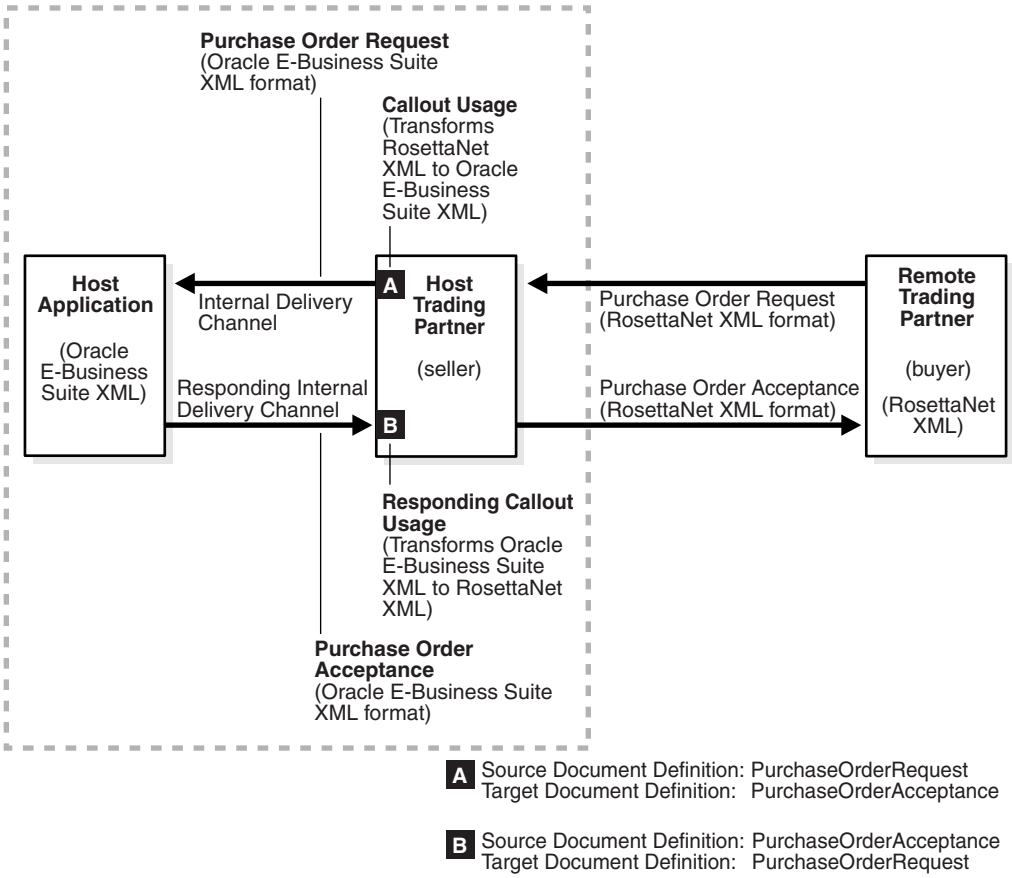
This chapter contains the following topics:

- Introduction to Callouts
- Creating a Callout
- Including a Callout in an Agreement
- Implementing a Callout

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.

Figure 12–1 A Purchase Order Example: Using Callouts for Differently Formatted XML Messages



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 12–2 for where you add parameters and Table 12–2 for a list of parameter attributes.

Creating a Callout Library JAR File

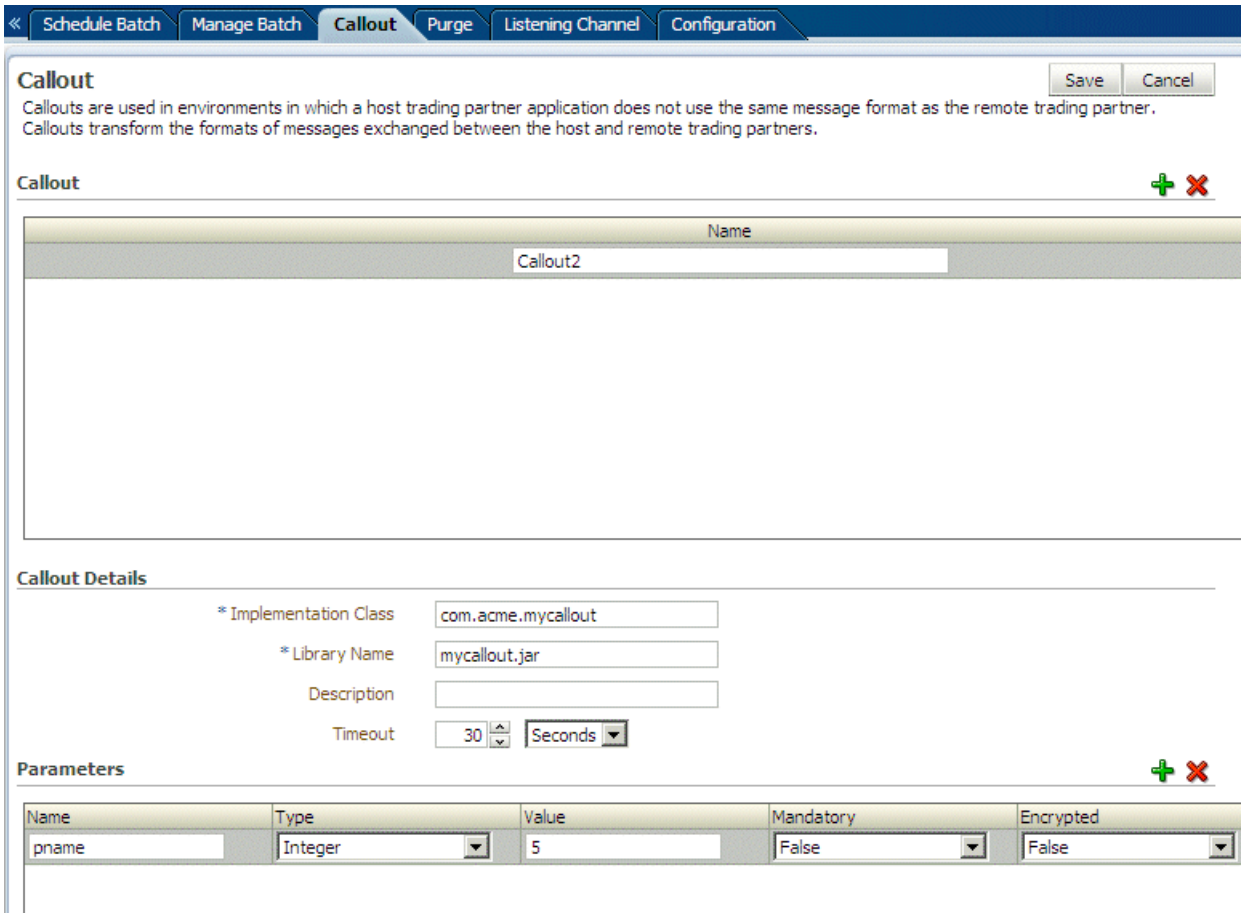
If the callout JAR file provided with Oracle B2B is not sufficient for your needs, you can create your own callout JAR file outside of Oracle B2B, following the standards described in the *Oracle Fusion Middleware B2B Callout Java API Reference*. Use the **Configuration** tab of the **Administration** 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.

Creating a Callout

To create a callout, provide callout details—the implementation class name and library name—and callout parameters, as shown in Figure 12–2.

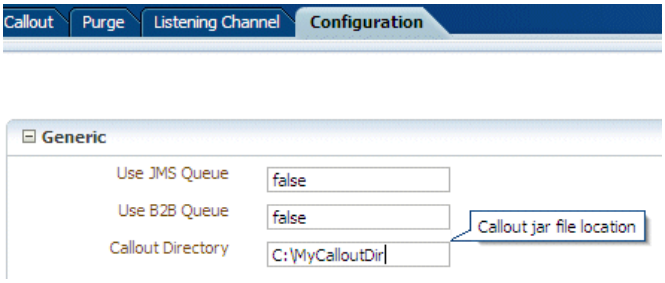
Figure 12–2 Creating a Callout



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

Field	Description
*Implementation Class	Enter the class file name without `.class`. **Note:** Oracle B2B includes a predefined class file named XSLTCalloutImpl that you can use for XML-to-XML transformations.
*Library Name	Enter the JAR file name that has the callout implementation classes. **Note:** If you specify one or more of your own callout JAR files, you must specify the directory location. Use the **Configuration** tab from the **Administration** link. The directory location for the default `b2b.jar` file included with Oracle B2B does not need to be specified. See "Setting Configuration Parameters" on page 15-1 for information on specifying the callout directory for your own callout JAR files. 
Description	Enter a description.
Timeout (seconds)	Enter the time limit in which to process the callout.

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 12–2 Callout Parameter Attributes

Field	Description
Name	Enter a parameter name.
Type	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.
Value	Enter a value. If **Encrypted** is set to **True**, then this value is encrypted.
Mandatory	Select **True** or **False**.
Encrypted	Select **True** or **False**.
Description	Enter an optional description.

After you create a callout, it is available to include in an agreement. See "Including a Callout in an Agreement" on page 12-5 for more information. If you change a callout after it is deployed with an agreement, a server restart is required.

To create a callout:

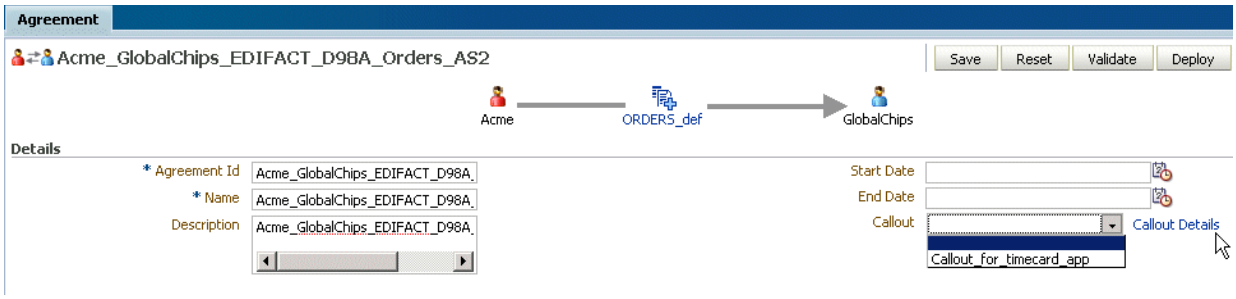
1. Click **Administration**, and then **Callout**.
2. In the **Callout** section, click **Add**.
3. Enter a name for the callout.
4. Enter callout details, as described in Table 12–1.
5. (Optional) Click **Add** in the **Parameters** section.
6. Enter a parameter name and attributes, as described in Table 12–2.
7. Click **Save**.

You can edit the details, parameters, or parameter values at any time, but not the callout name.

Including a Callout in an Agreement

After you create a callout, it is available to include in an agreement, as shown in Figure 12–3.

Figure 12–3 Specifying a Callout in an Agreement

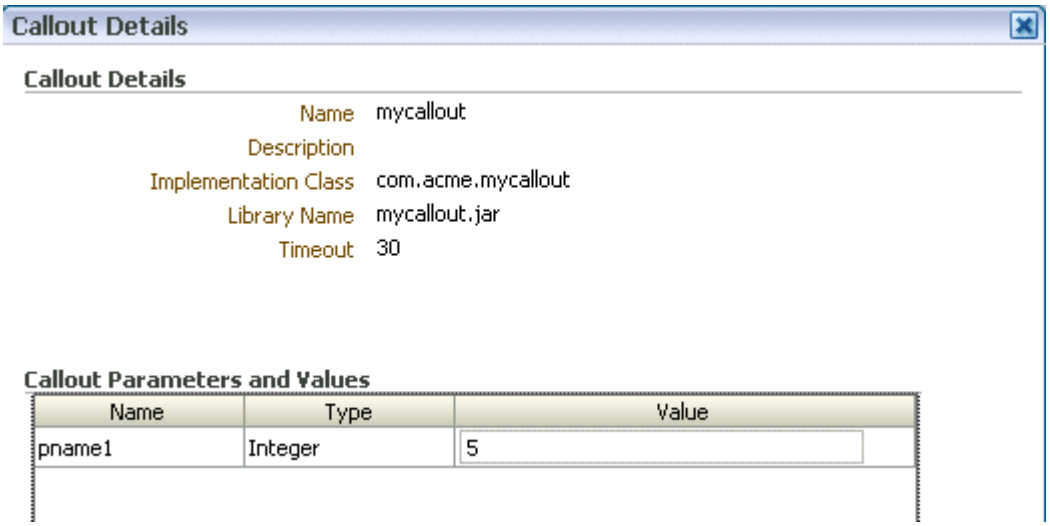


To include a callout in an agreement:

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.



6. Click **OK**.

Implementing a Callout

Example 12–1 shows how an incoming XML document is transformed to another XML document. The directory structure is `oracle.tip.callout`.

Example 12–1 Code Example of an XML-to-XML Transformation

```
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
      // create a callout output 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);
    }
  }
}
```

13

Purging Data

Use the Oracle B2B interface to purge design metadata and instance data.

This chapter contains the following topics:

- Purging Design Metadata and Instance Data

See the following for alternate methods of purging:

- Chapter 18, "B2B Command Line Tools"
- Chapter 19, "Scripts for Archiving and Restoring Data"

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 "Purging Messages" on page 16-2 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:

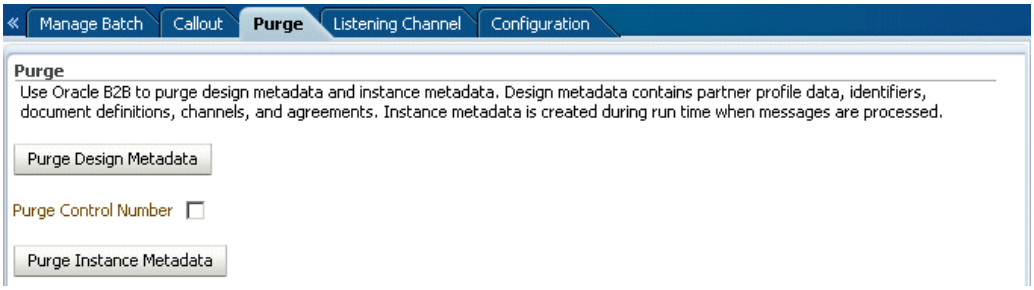
- Managing disk space and improving performance
- Removing repositories on a test system

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 you are purging instance data) Select **Purge Control Number** to reset the sequence.
3. Click **Purge Design Metadata** or **Purge Instance Data**.

   Note: You are purging instance (run-time) data, *not* instance metadata.



   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**.

14

Configuring Listening Channels

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:

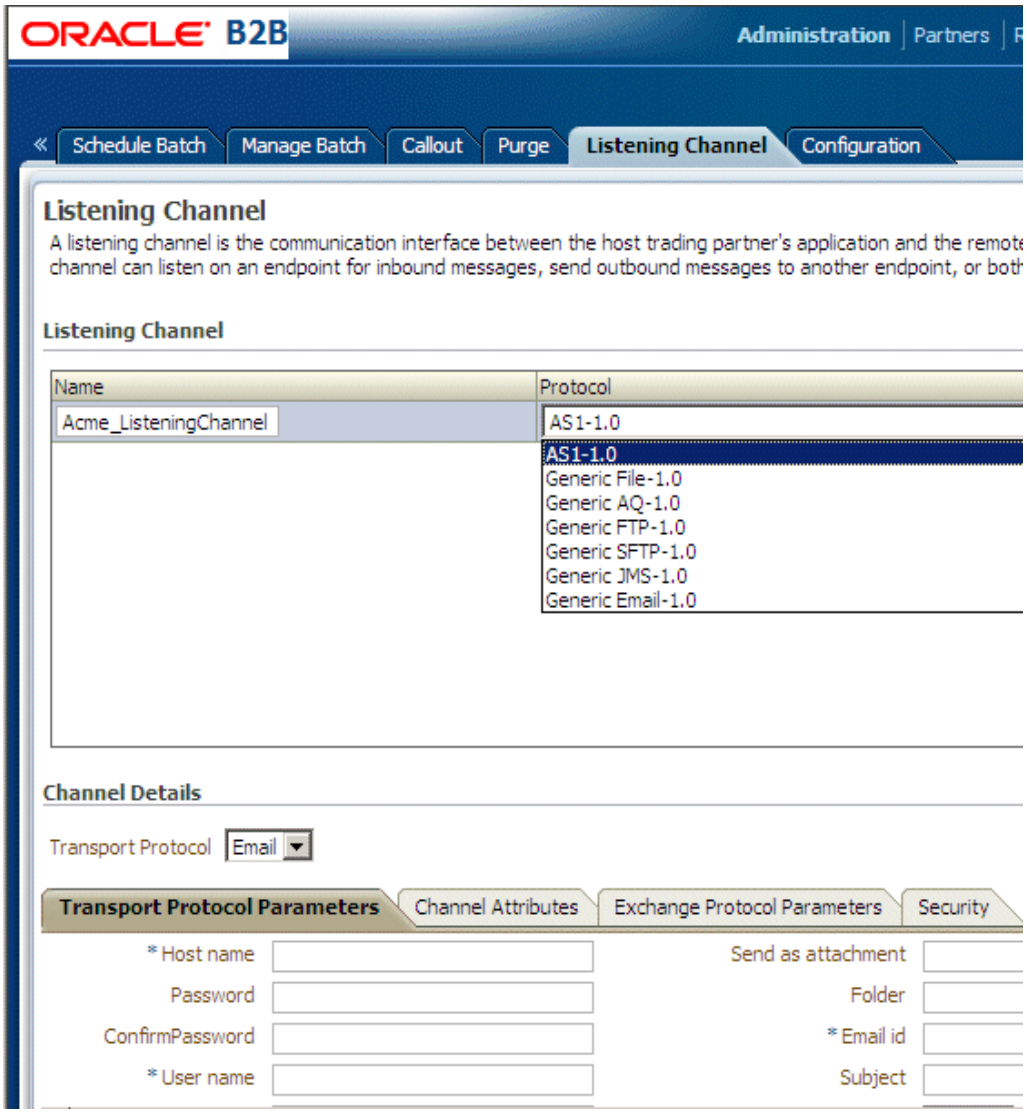
- Adding a Listening Channel and Protocol
- Using Transport Protocols
- Adding Listening Channel Details
- Configuring a Listening Channel

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.

Figure 14–1 Adding a Listening Channel



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 14–1 Listening Channel Protocols

Protocol	Description
AS1-1.0	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. AS1-1.0 is in preview mode for this release.
Generic File, Generic AQ, Generic FTP, Generic SFTP, Generic JMS, Generic Email	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 (acknowledgment mode must be set to None).

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

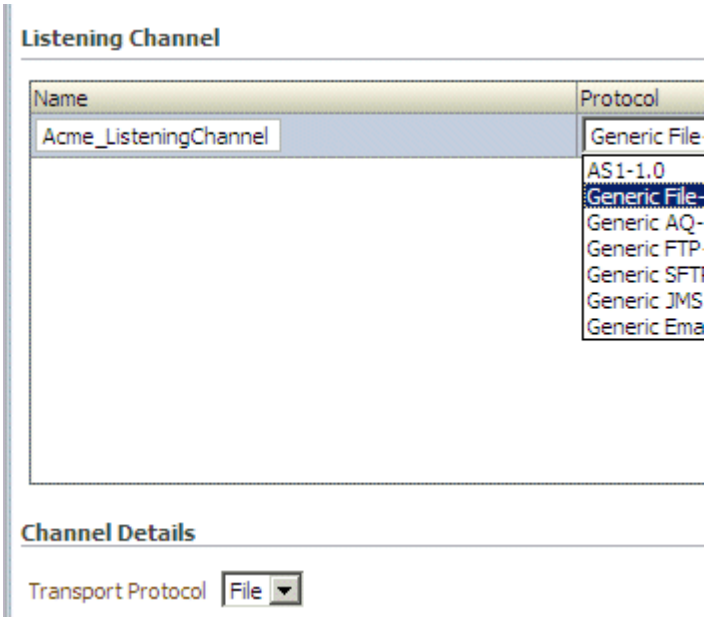


Table 14–2 describes the transport protocols available in Oracle B2B.

Table 14–2 Transport Protocols Available in Oracle B2B

Protocol	Description
Email	Use Email for AS1 and Email listening channels.
File	The File transport enables files to be picked up from a shared file directory.
AQ	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.
FTP	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).
SFTP	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.
JMS	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).

Adding Listening Channel Details

Listening channel details include transport protocol parameters, channel attributes, exchange protocol parameters, and security specifications. Table 14–3 describes these details.

Table 14–3 Listening Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
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 (trading partner address, such as a URI).	-
Channel mask	To enable SSL for FTP, enter one of the following: ▪ `Control`—Encrypts the control channel ▪ `Data`—Encrypts the data channel ▪ `Both`—Encrypts both the data and control channels The default is **None** (no SSL).	FTP (optional)
Cipher suites	Sets of ciphers defined in SSL.	FTP (optional)
Connection factory	The JNDI location or Java class name for the connection factory, as in `jms/b2b/B2BQueueConnectionFactory`.	JMS (optional)
Consumer	The client that receives the message.	AQ (optional)
Content type	The content type of the payload being sent over e-mail. The default content type is `text/plain`; other examples include `application/xml` and `application/edi`. This value is used only for the delivery channel (to send e-mail) and not for the listening channel. On the listening channel side, intelligence is built into the transport adapter to deal with different content types, so no configuration is required.	AS1 (optional) Email (optional)
Control port	Provide a value to change the default FTP port value (21)	FTP (optional)
Data port	For active FTP connections, use this option to configure the static/fixed data port of the FTP server.	FTP (optional)
Data source	The JNDI name of the JDBC data source to access AQ queues.	AQ (optional)
Destination name	The JMS destination name.	JMS (optional)
Email ID	The e-mail address to which messages are delivered (similar to specifying the path for a file channel or queues in AQ or JMS).	AS1 (required) Email (required)
Email Server	Select **IMAP** or **POP3**.	AS1 (required) Email (required)
Encoding	The encoding used in B2B to convert the contents of the inbound files.	FTP (optional)

Filename format	The following filename formats can be used:
	`%FROM_PARTY%`
	`%TO_PARTY%`
	`%DOCTYPE_NAME%`
	`%DOCTYPE_REVISION%`
	`%MSG_ID%`
	`%TIMESTAMP%`
	This filename format can be used for ebMS documents only:
	`%ACTIONNAME%`
	These formats can be used in any combination; for example,
	`%TO_PARTY%_%DOCTYPE_NAME%_%DOCTYPE_REVISION%.dat`
	produces something like `Acme_4010_850.dat`. Any file extension is allowed.
		File (optional)
		FTP (optional)
		SFTP (optional)

Folder	An absolute directory path is recommended.	AS1 (optional) Email (optional)

Table 14–3 (Cont.) Listening Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
Folder name	An absolute directory path is recommended.	File (required) FTP (required)
Host name	The trading partner's transport or e-mail server exchanging messages.	AS1 (required) AQ (optional) FTP (required) SFTP (required) Email (required)
Is Map Payload Alone	Indicates that the payload is sent alone as part of a JMS message of type `javax.jms.MapMessage`	JMS (optional)
Is topic	Select to indicate that JMS is communicating with a topic (not a queue).	JMS (optional)
Message type	Select a JMS messages type: **BYTES**, **TEXT**, or **MAP**.	JMS (optional)
Pass phrase and Confirm pass phrase	If you enter a private key file location, and if the private key file is pass-phrase protected, then enter the pass phrase.	SFTP (optional)
Password and Confirm Password	To use password authentication, provide a key store password, which is used for authentication.	AS1 (optional) AQ (optional) FTP (optional) SFTP (optional) JMS (optional) Email (optional)
Path	The absolute directory path where messages are sent from or received.	SFTP (required)
Polling interval	The time interval in milliseconds during which Oracle B2B polls the server for inbound messages.	AS1 (optional) File (optional) AQ (optional) FTP (optional) SFTP (optional) JMS (optional) Email (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.	AQ (optional) SFTP (required)
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.	SFTP (optional)
Queue name	The AQ queue name.	AQ (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).	AQ (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.	AS1 (optional) Email (optional)
SID	System ID to identify an Oracle database.	AQ (optional)
Subject	The subject header of the e-mail message.	AS1 (optional) Email (optional)
Subscriber ID	The JMS subscriber ID is required if JMS is communicating with a topic.	JMS

Table 14–3 (Cont.) Listening Channel Details and Associated Protocols

Protocol/Parameter	Description	Protocol Used With
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 (required) AQ (optional) FTP (required) SFTP (required) JMS (optional) Email (required)
Use proxy	Select this option if a proxy server is used.	FTP (optional) SFTP (optional)
Channel Attributes	The channel is the communication interface between the host trading partner's host application and its installation.	
Ack Mode	Select **Sync**, **Async**, or **None** for the mode in which the trading partner receives messages. Select **None** for all generic exchanges.	AS1 (optional)
Description	Provide an optional description.	AS1 (optional) File (optional) AQ (optional) FTP (optional) SFTP (optional) JMS (optional) Email (optional)
Enable/Disable Channel	The channel is the communication interface between the host trading partner's host application and its installation.	AS1 (required) File (required) AQ (required) FTP (required) SFTP (required) JMS (required) Email (Required)
Internal	Select this option if the channel is internal to the host trading partner's enterprise. (This feature is disabled for AS1.)	File (optional) AQ (optional) FTP (optional) SFTP (optional) JMS (optional) Email (optional)
Response Mode	Select **Sync**, **Async**, or **None**,	AS1 (required)
Retry Count	The number of times that Oracle B2B retries sending the message.	AS1 (optional) File (optional) AQ (optional) FTP (optional) SFTP (optional) JMS (optional) Email (optional)
Retry Interval	The time interval in seconds during which Oracle B2B attempts to resend the message. A time interval of 2 minutes increments the HH:MM:SS timestamp as follows: If the sent timestamp is 3:42:58, then 42 seconds is incremented by 2 minutes and the retry is sent at 3:44:00. The seconds are dropped in the retry increment. Subsequent retries are at 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	AS1 (optional) File (optional) AQ (optional) FTP (optional) SFTP (optional) JMS (optional) Email (optional)

Table 14–3 (Cont.) Listening Channel Details and Associated Protocols

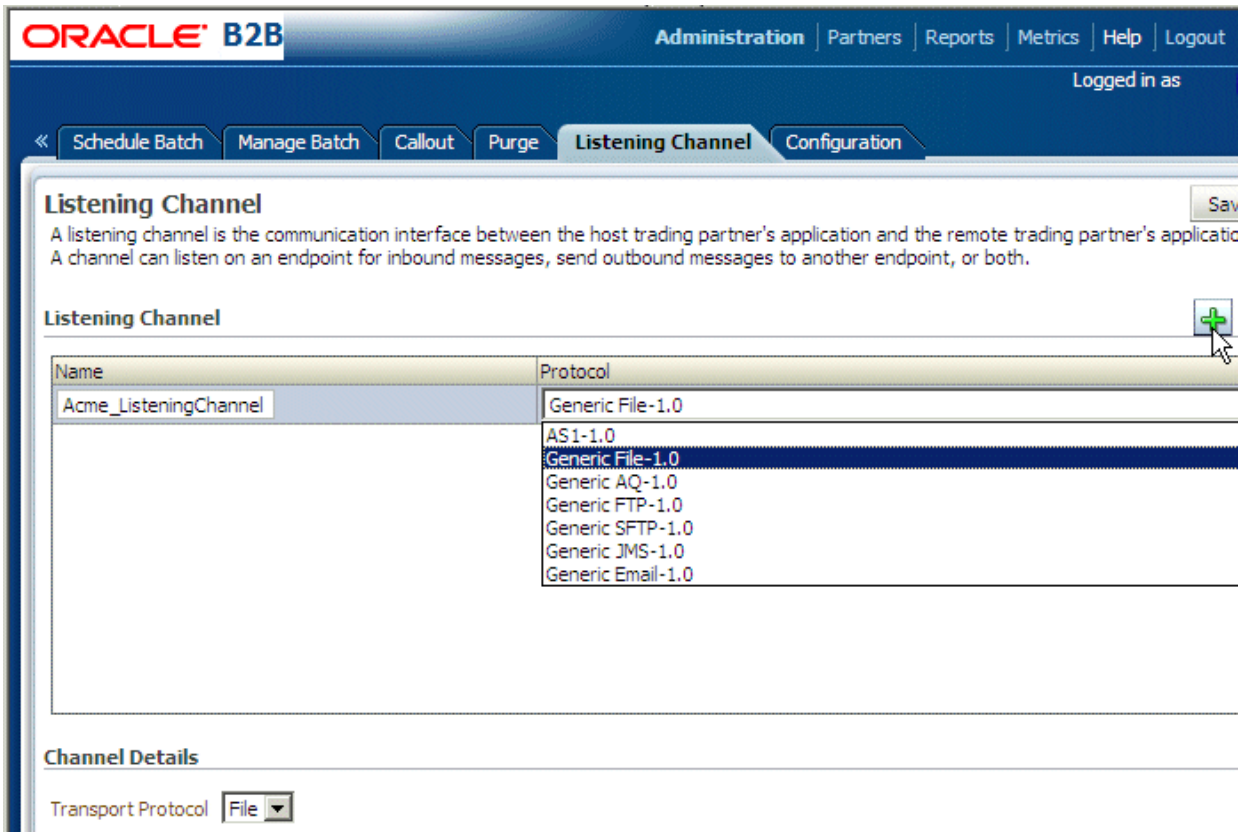
Protocol/Parameter	Description	Protocol Used With
Exchange Protocol Parameters	The exchange protocol defines the headers, acknowledgments, and packaging that puts the headers and payload together (the message exchange mechanism). The exchange protocol also defines signing and compression.	-
Signed and Compressed	Select to enable these options.	AS1 (optional)
Security Parameters	-	-
Ack Signed	Select this option to ensure that the responder acknowledges receipt of the messages; nothing needs to be provided.	AS1
Digital Signature	If **Message Signed** is selected, then select one of the following: SMIME 3.0 with MD5 - RSA SMIME 3.0 with SHA1 - RSA	AS1
Encryption	If **Message Encrypted** is selected, then select one of the following: SMIME 3.0 with DES SMIME 3.0 with 3DES SMIME 3.0 with RC2 - 40 SMIME 3.0 with RC2 - 64 SMIME 3.0 with RC2 - 128	AS1
Message Encrypted	Select this option to enable message encryption. This option requires you to select an encryption schema in the **Encryption** field.	AS1
Message Signed	Select this option to provide one of the digital signatures in the **Digital Signature** field.	AS1

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.



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 within the transport protocol parameters category) 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 14–3 describes the channel attributes (listed in alphabetical order within the channel attributes category) 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 14–3 describes the exchange protocol parameters (listed in alphabetical order within the exchange protocol parameters category) and the protocols to which the attributes apply.
3. Click **Save**.

To add security parameters:

1. Click the **Security Parameters** tab.
2. Provide security parameters, depending on the channel/transport protocols selected.

   Table 14–3 describes the security parameters (listed in alphabetical order within the security parameters category) and the protocols to which the attributes apply.
3. Click **Save**.

15

Configuring B2B System Parameters

Configuration settings that were formerly accessible in `oracle.tip` properties files are now accessible in the Oracle B2B interface on the **Configuration** tab. Settings on the **Configuration** tab override property settings in `b2b-config.xml`. See Appendix B, "Properties of b2b-config.xml."

This chapter contains the following topics:

- Setting Configuration Parameters

Setting Configuration Parameters

Figure 15–1 shows the configuration settings available in the Oracle B2B interface.

Figure 15–1 Configuration Parameters in the Oracle B2B Interface

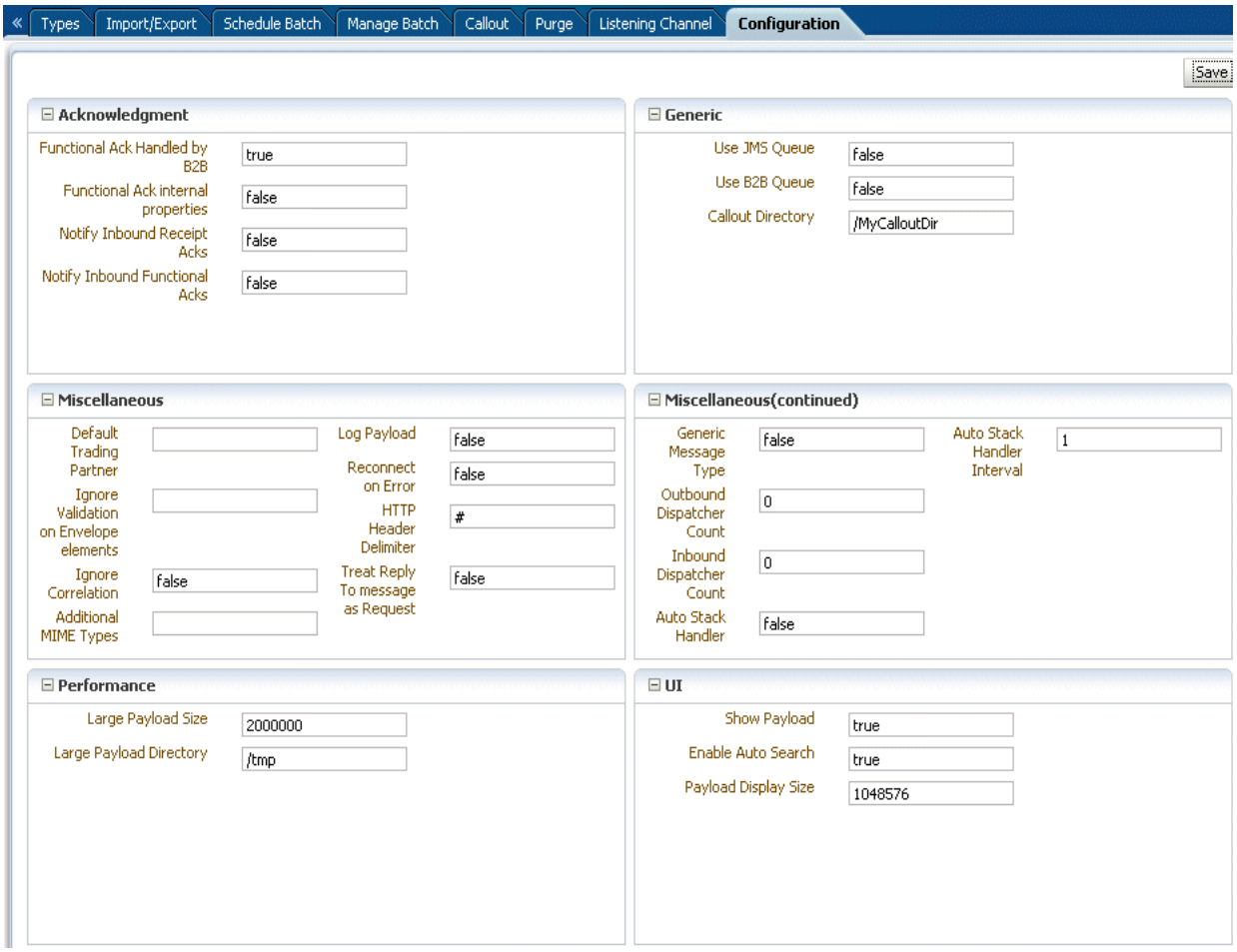


Table 15–1 describes the configuration parameters.

Table 15–1 Configuration Settings

Field	Description
Acknowledgment Settings	-
Functional Ack Handled by B2B	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 "Setting b2b.FAHandleByB2B for EDI EDIFACT and EDI X12" on page B-2 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.
Functional Ack Internal Properties	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.
Notify Inbound Receipt Acks	If set to true, B2B sends an acknowledgment notification to the application when an exchange acknowledgment is received.
Notify Inbound Functional Acks	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.
Generic Settings	-
Use JMS Queue	Set this option to true to use the default JMS queues (B2B_IN_QUEUE and B2B_OUT_QUEUE) as the default internal delivery channel.
Use B2B Queue	Set this option to true to use the default AQ queues (IP_IN_QUEUE and IP_OUT_QUEUE) as the default internal delivery channel.
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 `\`.
Miscellaneous Settings	-
Default Trading Partner	Defaults to this trading partner if trading partner agreement identification fails. Used for HL7 documents.
Ignore Validation on Envelope Elements	When this property is set to true, the validation of look-up parameters is turned off Use this option to provide a list of envelope elements, separated by commas, to be ignored during look-up validation. The possible values are InterchangeSenderID, InterchangeReceiverID, GroupReceiverID, GroupSenderID, TransactionAssociationAssignedCode, InterchangeReceiverQual, InterchangeSenderQual, and InterchangeControlVersion.

Table 15–1 (Cont.) Configuration Settings

Field	Description
Ignore Correlation	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.
Additional MIME Types	Use to specify attachments (additional MIME types) in addition to the default MIME types supported by B2B for ebxml exchanges. By default, B2B supports `application/xml : application/octet-stream : application/EDIFACT : application/EDI-X12 : application/jpg : image/jpeg : application/gzip : application/x-gzip: application/pkcs7-signature`.
Log Payload	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.
Reconnect on Error	If set to true, the AQ adapter retries the enqueue operation when the initial enqueue fails. This parameter is not available in this release.
HTTP Header Delimiter	A delimiter to separate the HTTP headers provided in the **Additional Transport Headers** field for HTTP delivery channel configuration.
Treat Reply to Message as Request	Used in ebMS to indicate that the conversation message is to be considered as a request message.
Miscellaneous (continued)	-
Generic Message Type	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.
Outbound Dispatcher Count	The number of dispatchers used for handling the outbound messages. Used in message sequencing for MLLP. The default value is 0.
Inbound Dispatcher Count	The number of dispatchers used for handling the inbound messages. Used in message sequencing for MLLP. The default value is 0.
Auto Stack Handler	Used in stacking for MLLP. If true, the stack handler processes stacked messages in automatic mode. The default value is false.
Auto Stack Handler Interval	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.
Performance Settings	-
Large Payload Size	Specify a large payload size, in bytes. The default value is 2,000,000 (2MG).
Large Payload Directory	The default directory is `/tmp`. For Windows-based systems, change the directory to an appropriate directory, such as `C:\temp`.
UI Settings	-
Show Payload	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.
Enable Auto Search	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.
Payload Display Size	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.

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**.

Part IV

Reports and Metrics

This part contains the following chapters:

- Chapter 16, "Creating Reports"
- Chapter 17, "Using B2B Metrics"

16

Creating Reports

Oracle B2B reports provide real-time status on the run-time behavior of deployed data.

This chapter contains the following topics:

- Introduction to Reports
- Creating Business Message Reports
- Creating Wire Message Reports
- Creating Application Message Reports
- Creating Error Reports
- Creating Conversation Reports

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 "Creating Business Message Reports" on page 16-2
- Wire messages—See "Creating Wire Message Reports" on page 16-5
- Application messages—See "Creating Application Message Reports" on page 16-7
- Error messages—See "Creating Error Reports" on page 16-10
- Conversation messages—See "Creating Conversation Reports" on page 16-12

> **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.

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 "Adding Trading Partner Users" on page 5-7 for how to assign the Monitor role.

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.

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 needs to 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 needs to be retried.

Resubmitting a wire message, for an outbound message, only tries to redeliver 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 needs to be translated and validated again.

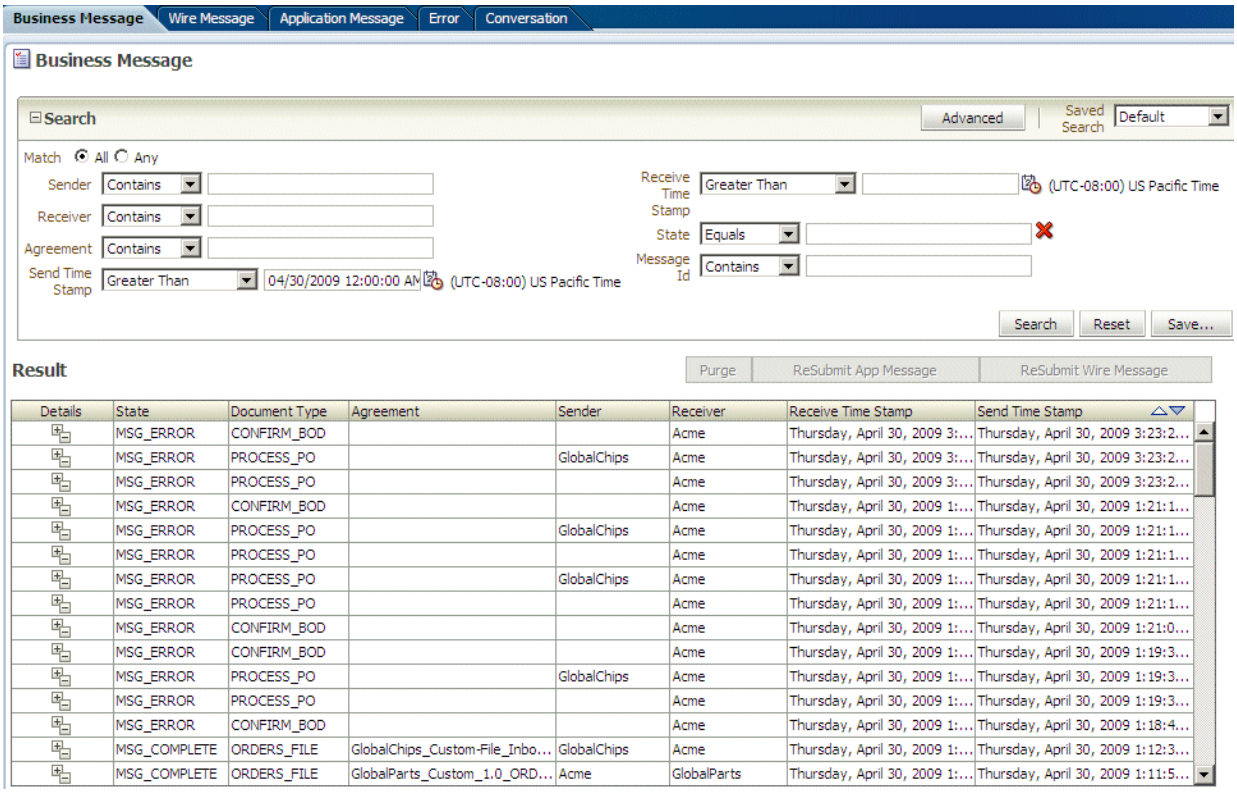
Note: If you resubmit an inbound AS2 synchronous wire message, the MDN is generated, but it is not returned to the sender in synchronous mode. This is because the sender is not the one who is initiating the originating message. In this scenario, the MDN message state is in the `MSG_COMPLETE` state.

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.

Field	Description
Match	Select **All** or **Any**.
Sender	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of a trading partner name.
Receiver	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of a trading partner name.
Agreement	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of a trading partner agreement name.
Send Time Stamp	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.
Receive Time Stamp	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.

Field	Description

State	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`

Message ID	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of a message ID.

3. To add more search fields, click **Advanced** and select from **Add Fields**:

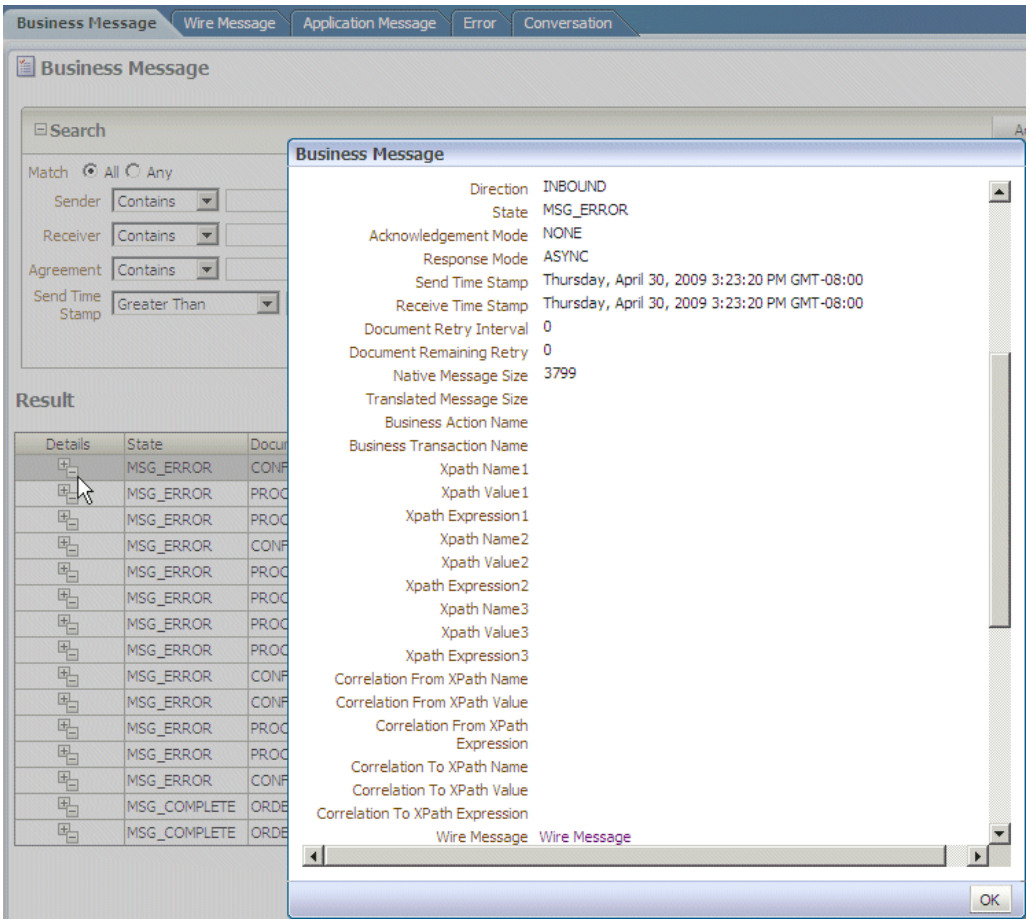
Field	Description
Document Protocol Name	Enter Custom, EDI_EDIFACT, EDI_X2, HL7, OAG, PositionalFlatFile, RosettaNet, or UCCNet. (**Equals** is the only operator.)
Document Type	Select from a previously created document type, for example, 850 for EDI X12. (**Equals** is the only operator.)
Document Protocol 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.

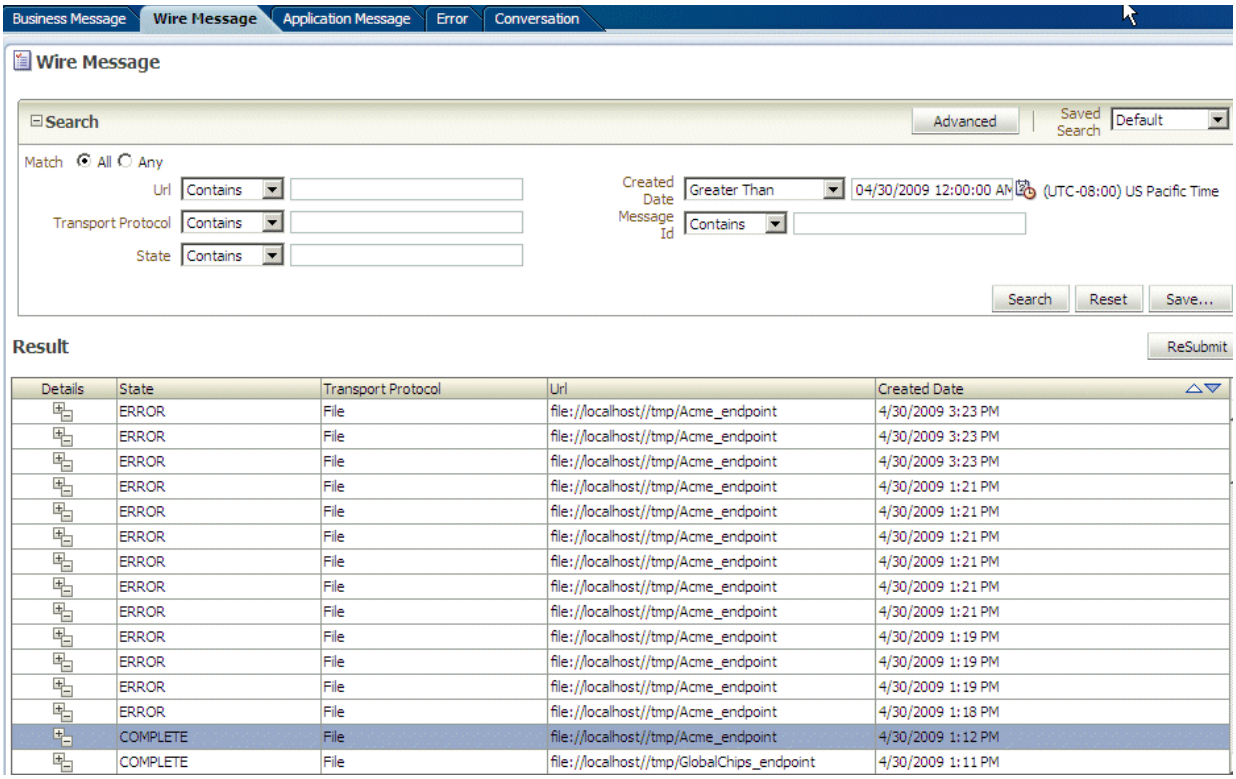


Creating Wire Message Reports

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 16–2 shows a wire message report.

Figure 16–2 Wire Message Report



To create a wire message report:

1. Click **Reports**, and then **Wire Message**.
2. Provide search parameters.

Field	Value
URL	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of the URL.
Transport Protocol	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of the transport protocol.

State	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`

Created Date	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.
Message ID	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of a message ID.

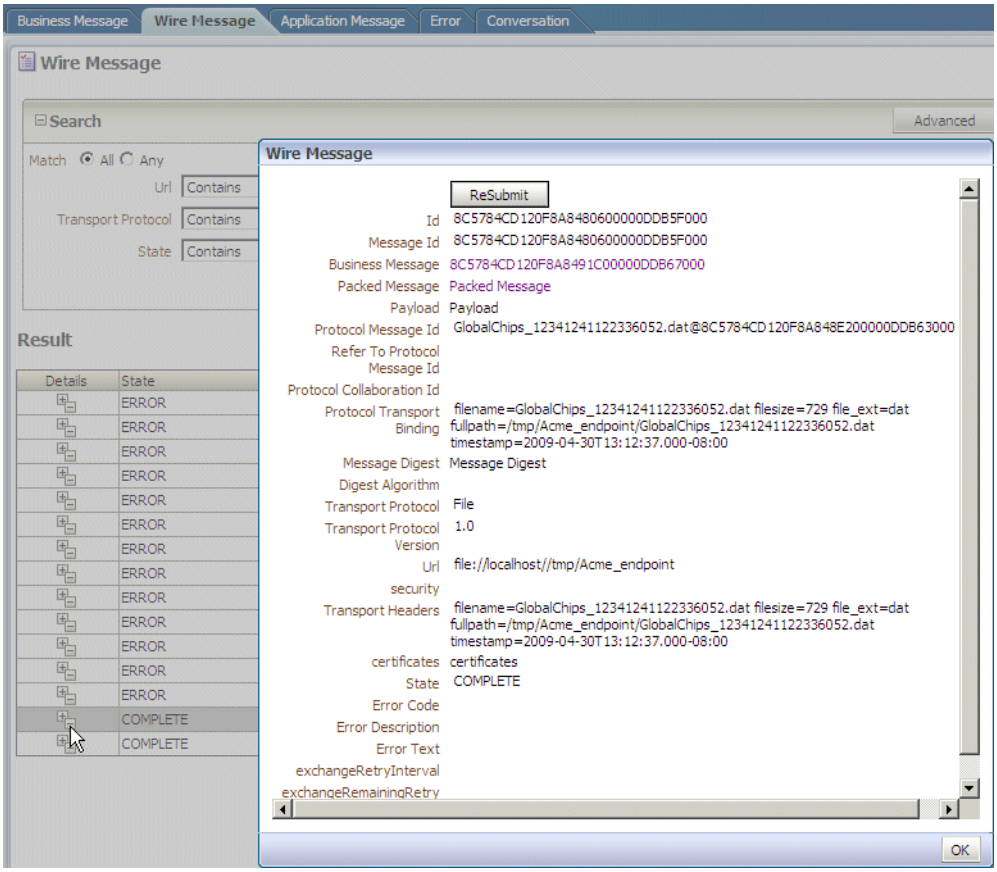
3. To add more search fields, click **Advanced** and select from **Add Fields**:

Field	Description
Document Protocol Name	Select from 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.)

4. Click **Search**.

   View the results, as shown in Figure 16–2.

5. In the **Details** column of the **Results** area, click the icon to see report details.

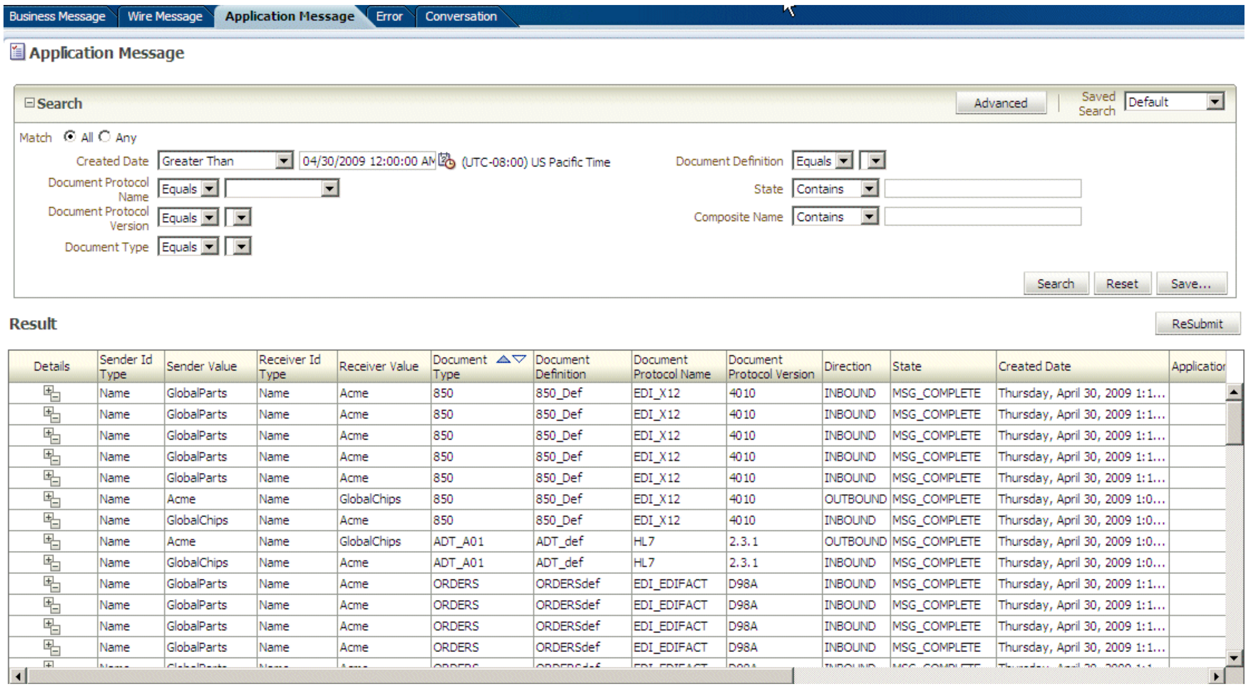


Creating Application Message Reports

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 16–3 shows an application message report.

Figure 16–3 Application Message Report



To create an application message report:

1. Click **Reports**, and then **Application Message**.
2. Provide search parameters.

Field	Description
Match	Select **All** or **Any**.
Created Date	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.
Document Protocol Name	Select from Custom, EDI_EDIFACT, EDI_X2, HL7, OAG, PositionalFlatFile, RosettaNet, or UCCNet. (**Equals** is the only operator.)
Document Protocol Version	Select from a previously created document protocol version. (**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 Definition	Select from a previously created document definition. (**Equals** is the only operator.)

State	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`

Field	Description
Composite Name	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of the SOA composite application name.

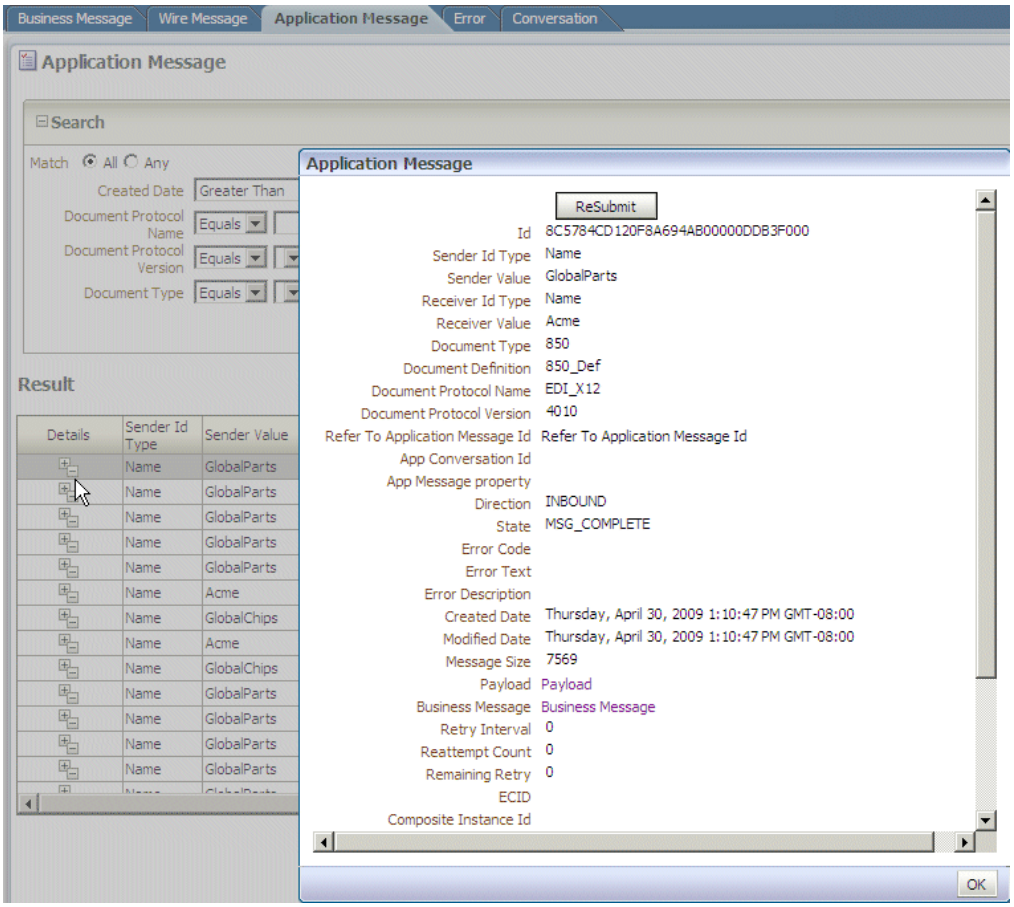
3. To add more search fields, click **Advanced** and select from **Add Fields**:

Field	Description
Application Name	Provide the name of the application.
Composite Version	Provide the version of the SOA composite application in Oracle JDeveloper.
ECID	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide an instance ID.
Sender ID Type	Provide the sender's identifier type, such as Name, DUNS, or MLLP ID.
Service Name	Provide the name of the B2B service binding component.
Receiver ID Type	Provide the receiver's identifier type, such as Name, DUNS, or MLLP ID
Receiver Value	Provide the value of the receiver's identifier type. For example, if DUNS is the Receiver ID Type, provide the DUNS number.
Sender Value	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.
Reference Name	Provide the name of the B2B reference binding component.
Fabric CompositeDn	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of the composite name.

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.

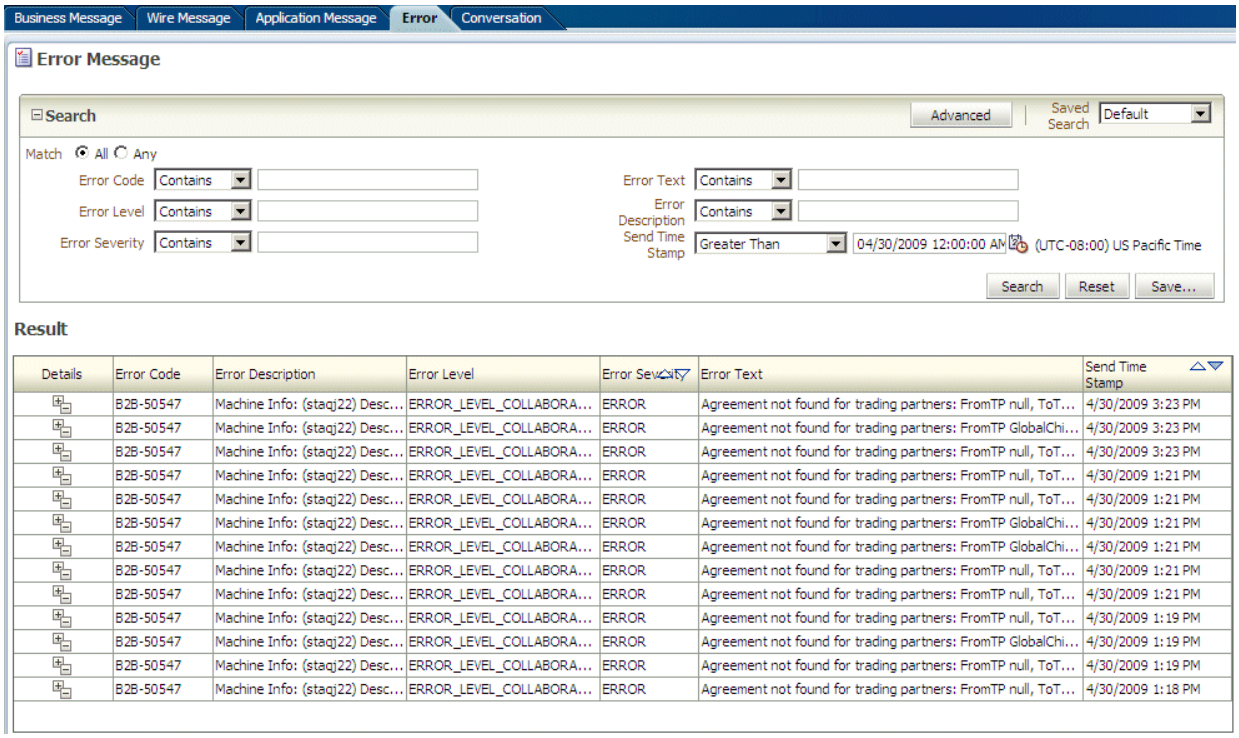


Creating Error Reports

Error status reports provide error message details. These details include the error code, error text, business message identification, message date, and message details.

Figure 16–4 shows an error report.

Figure 16–4 Error Report



To create an error report:

1. Click **Reports**, and then **Error**.
2. Provide search parameters.

Field	Description
Match	Select **All** or **Any**.
Error Code	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of an error code.
Error Level	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of an error level
Error Severity	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of an error severity.
Error Text	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of the error text.
Error Description	Select from **Starts With**, **Equals**, **Contains**, or **Ends With**. Provide all or part of the error description.
Send Time Stamp	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.

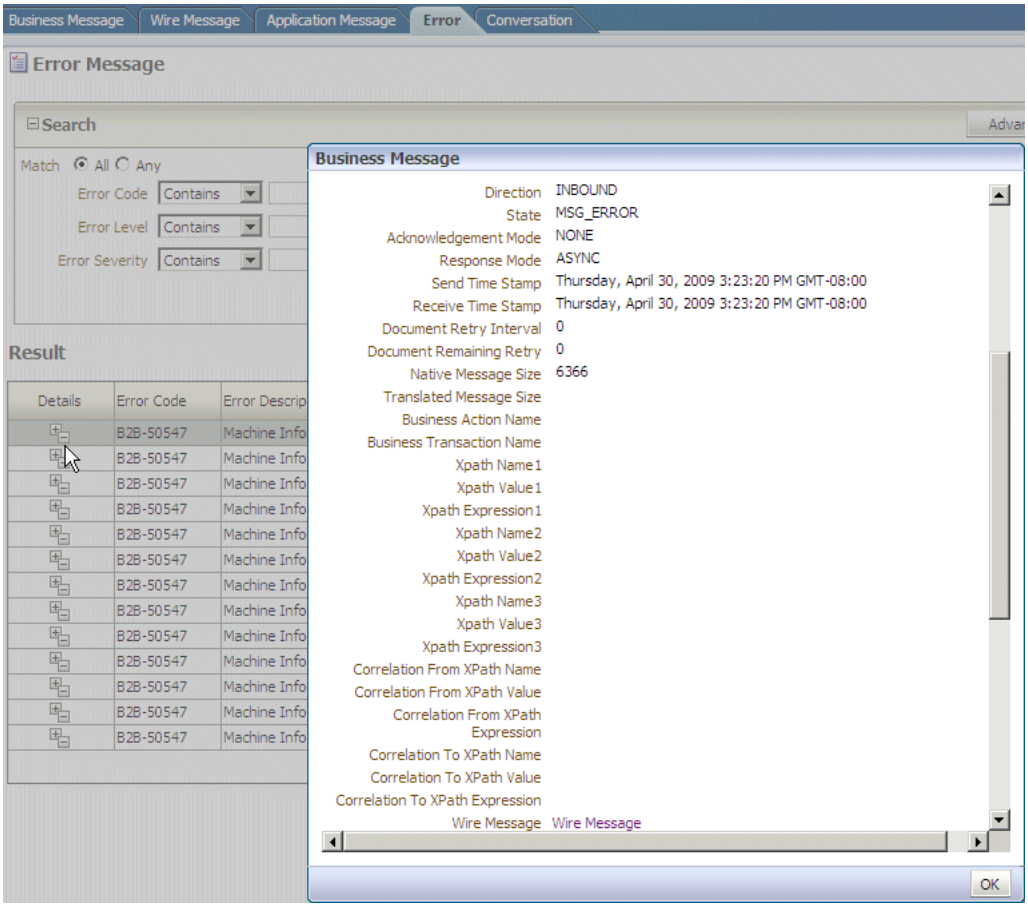
3. To add more search fields, click **Advanced** and select from **Add Fields**:

Field	Description
Document Definition	Select from a previously created document definition. (**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 Protocol Name	Select from Custom, EDI_EDIFACT, EDI_X2, HL7, OAG, PositionalFlatFile, RosettaNet, or UCCNet. (**Equals** is the only operator.)

4. Click **Search**.

   View the results, as shown in Figure 16–4.

5. In the **Details** column of the **Results** area, click the icon to see report details.



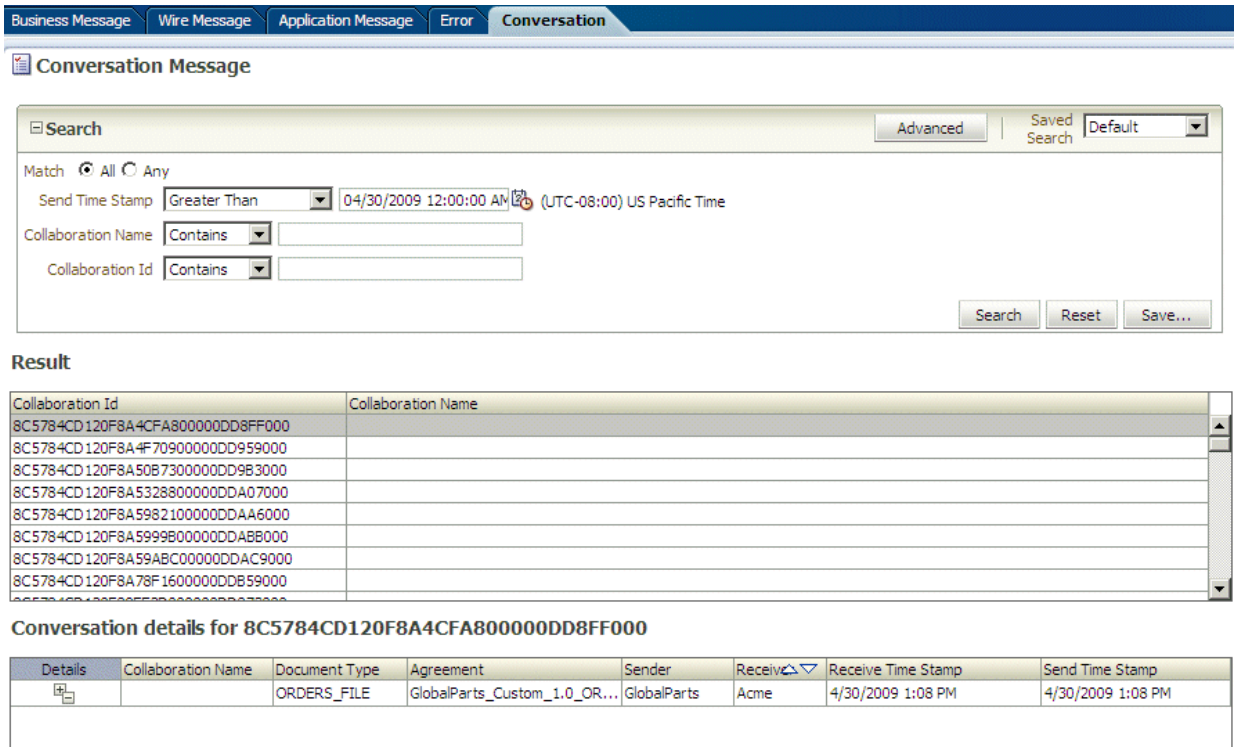
Creating Conversation Reports

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 and 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 16–5 shows a conversation report.

Figure 16–5 Conversation Report



To create a conversation report:

1. Click **Reports**, and then **Conversation**.
2. Provide search parameters.

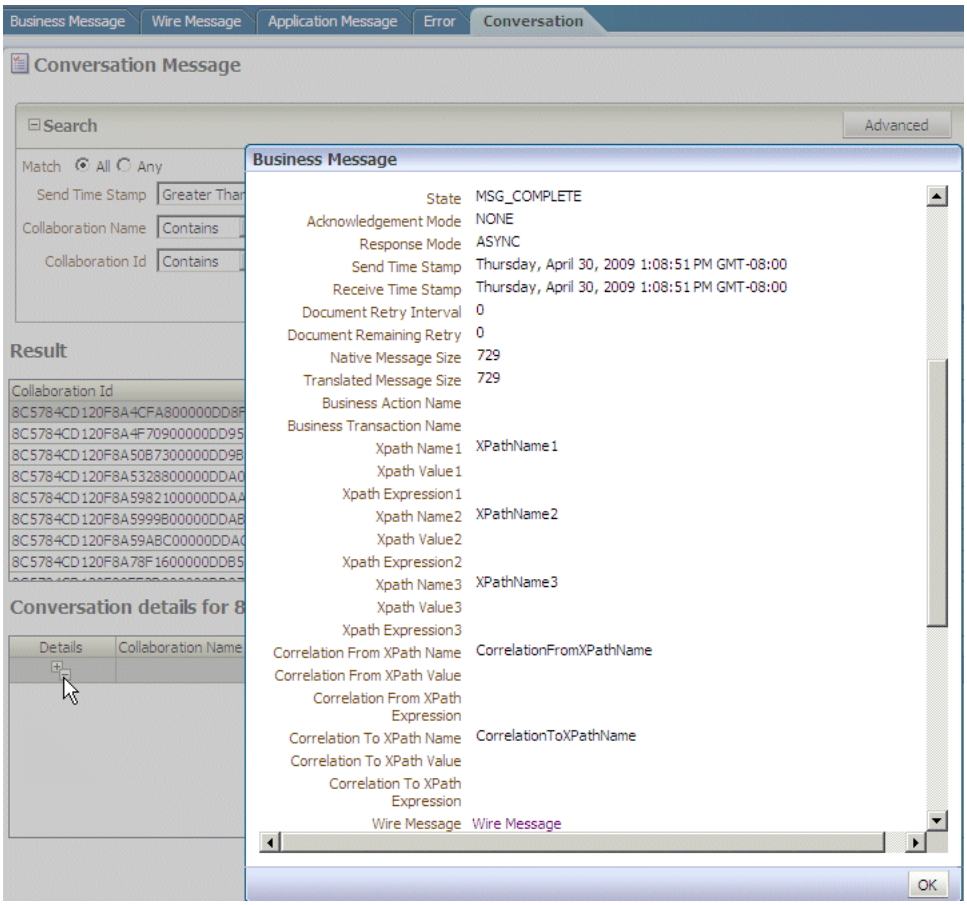
Field	Description
Match	Select **All** or **Any**.
Send Time Stamp	Select from **Less Than**, **Greater Than**, **Greater Than Equals**, **Equals**, or **Less Than Equals**. Provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the **Select Date and Time** icon.
Collaboration Name	Applies to ebMS and RosettaNet documents and is available from header information.
Collaboration ID	Applies to ebMS and RosettaNet documents and is available from header information.

   No additional fields can be added using the **Advanced** search button.

3. Click **Search**.

   View the results, as shown in Figure 16–5.

4. In the **Details** column of the **Results** area, click the icon to see report details.


Business Message
Wire Message
Application Message
Error
Conversation
Conversation Message
Search
Advanced
Match All Any
Send Time Stamp
Collaboration Name
Collaboration Id
Contains
Result
Collaboration Id
Conversation details for 8
Details
Collaboration Name
Business Message
State MSG_COMPLETE
Acknowledgement Mode NONE
Response Mode ASYNC
Send Time Stamp Thursday, April 30, 2009 1:08:51 PM GMT-08:00
Receive Time Stamp Thursday, April 30, 2009 1:08:51 PM GMT-08:00
Document Retry Interval 0
Document Remaining Retry 0
Native Message Size 729
Translated Message Size 729
Business Action Name
Business Transaction Name
Xpath Name1 XPathName1
Xpath Value1
Xpath Expression1
Xpath Name2 XPathName2
Xpath Value2
Xpath Expression2
Xpath Name3 XPathName3
Xpath Value3
Xpath Expression3
Correlation From XPath Name CorrelationFromXPathName
Correlation From XPath Value
Correlation From XPath Expression
Correlation To XPath Name CorrelationToXPathName
Correlation To XPath Value
Correlation To XPath Expression
Wire Message Wire Message
OK

17

Using B2B Metrics

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:

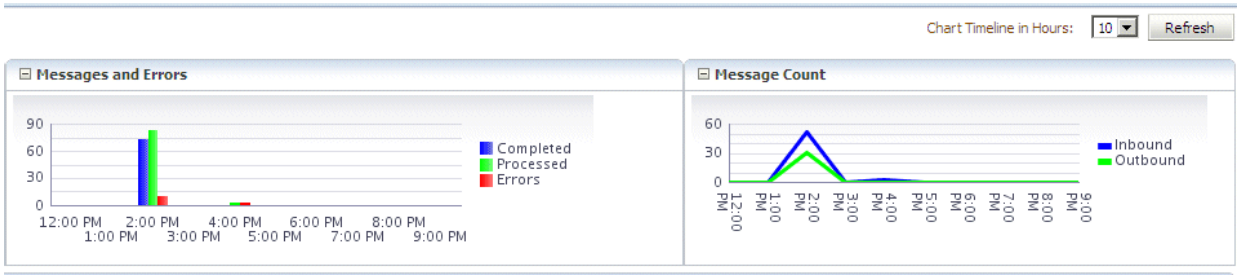
- Introduction to B2B Metrics
- B2B System Metrics
- B2B Partner Metrics

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.

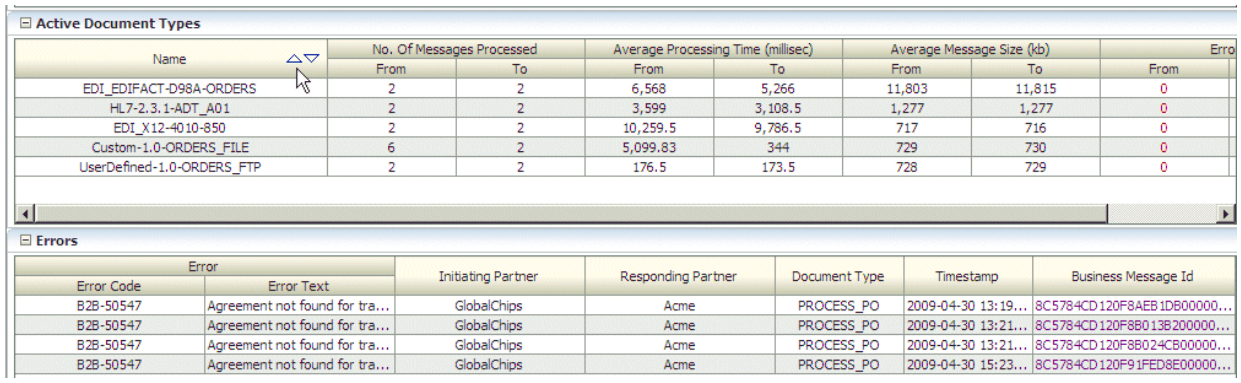
Figure 17–1 The Messages and Errors Chart and Message Count Chart



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 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.

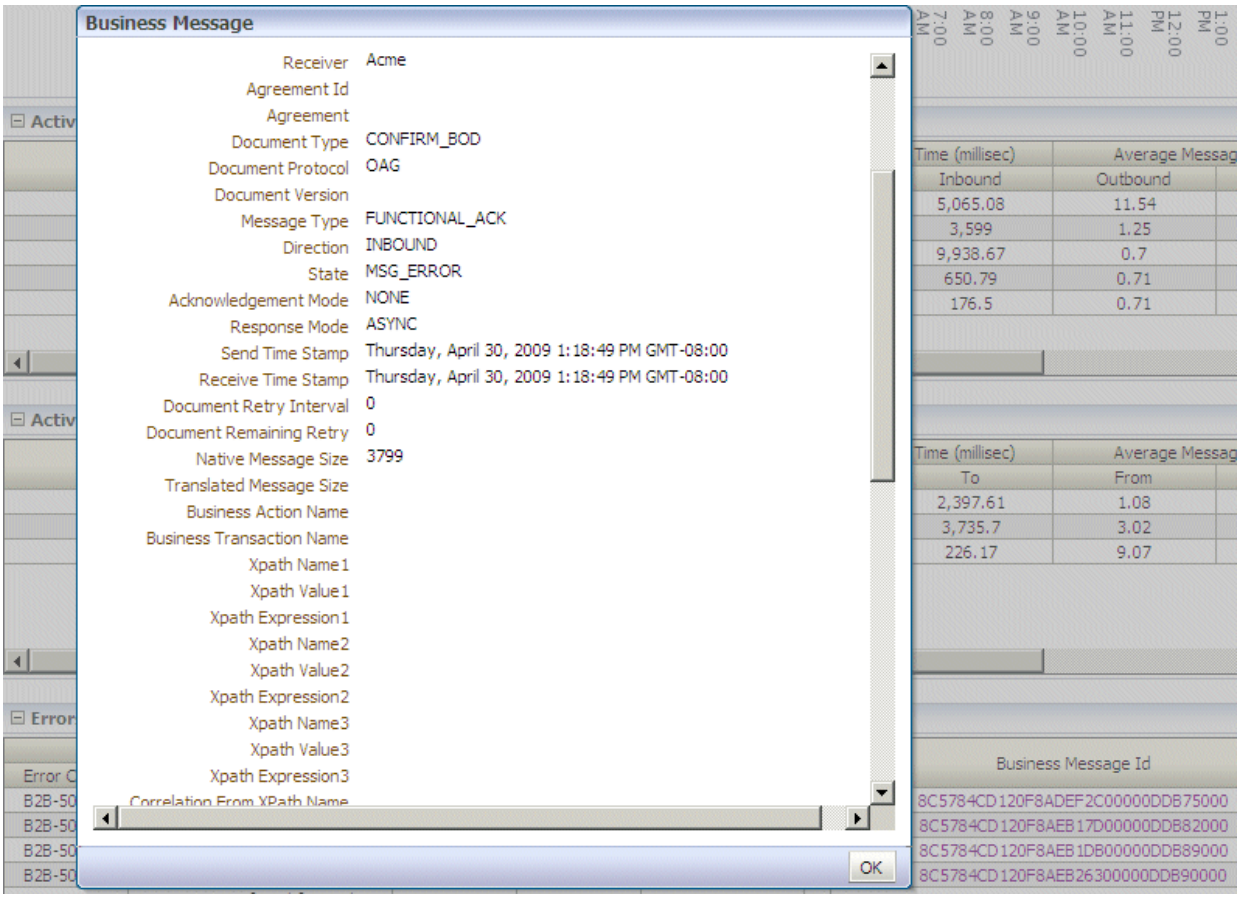
Most fields in the active document types, active trading partners, and errors tables can be sorted in ascending or descending order, as shown in Figure 17–2.

Figure 17–2 Sorting Columns



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.

Figure 17–3 Business Message Details



B2B System Metrics

Figure 17–4 shows system metrics summary data.

Figure 17–4 System Metrics

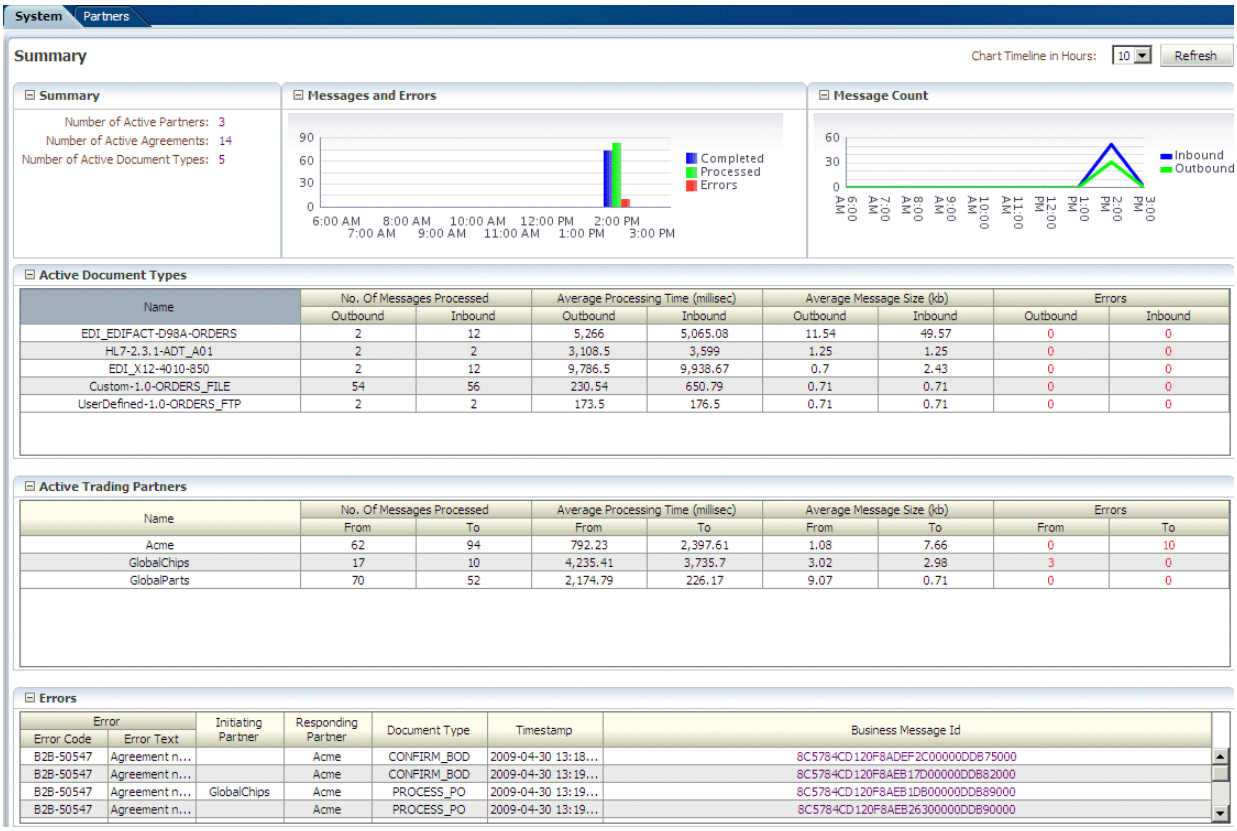


Table 17–1 describes the information on the **System** metrics tab.

Table 17–1 B2B System Metrics

Area	Description
Summary	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.
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.
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.
Active Trading Partners	Active trading partners are partners for which an agreement has been deployed and is in an active state. The host trading partner is included in the list. Messages processed include completed plus errored messages, that is, active messages.
Errors	Error message text is available from the Java resource bundle. The business message IDs link to business message details.

B2B Partner Metrics

Figure 17–5 shows metrics summary data for a selected trading partner.

Figure 17–5 Partner Metrics

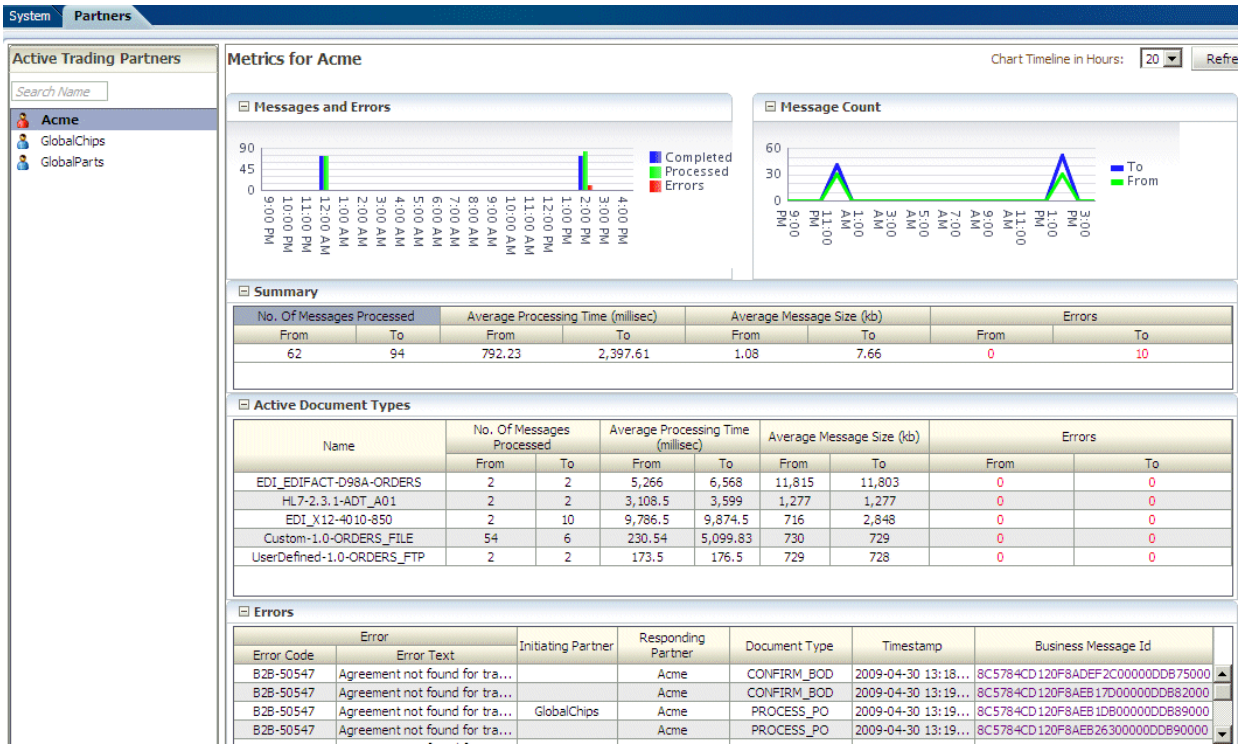


Table 17–2 describes the information on the **Partners** metrics tab.

Table 17–2 B2B Partner Metrics

Area	Description
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 quantified.
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.
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, "Scripts for Archiving and Restoring Data"
- Chapter 20, "Utilities for Enqueuing and Dequeuing"

18

B2B Command Line Tools

B2B command line tools are available for a number of tasks.

Note:

- Command line tools are for administrator use only.
- Self-service APIs are not available in this release.

This chapter contains the following topics:

- Prerequisites for Running the Command Line Tools
- Purging Data
- Importing Data
- Exporting Data
- Deploying Agreements
- Validating B2B Metadata
- CPP/CPA Import
- CPP/CPA Export
- CPP/CPA Templates

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.

Purging Data

Note: Before purging data, exporting or archiving data is recommended.

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 Options for ant -f ant-b2b-util.xml b2bpurge

Option	Description	Domain	Required
`mode`	Specifies purging design-time or run-time data.	`DT` `RT`	No

`msgState`	Deletes messages with the specified message state. Used for run-time data.	`MSG_COMPLETE`
		`MSG_ERROR`
		`MSG_WAIT_TRANSMIT`
		`MSG_WAIT_FA`
		`MSG_WAIT_BATCH`
			No. If `msgstate` is present, then `start` and `end` must be used.

`start`	Deletes messages that are created on or after the specified date. Used for run-time data.	`dd-MMM-yyyy`	No
`end`	Deletes messages that are created on or before the specified date. Used for run-time data.	`dd-MMM-yyyy`	No
`purgecontrolnumber`	Deletes control numbers. Used for run-time data.	`true` `false` (default)	No
`host`	Used for design-time data.		

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
```

Importing Data

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 Options for ant -f ant-b2b-util.xml b2bimport

Option	Description	Domain	Required
`exportfile`	Location of the export (ZIP) file	-	Yes
`overwrite`	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`.	`true` `false` (default)	No
`localfile`	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.	`true` `false` (default)	No

Exporting Data

Exports the entire repository (without policy details) if no other options are specified.

```
ant -f ant-b2b-util.xml b2bexport
```

Table 18–3 Options for ant -f ant-b2b-util.xml b2bexport

Option	Description	Domain	Required
`exportfile`	Location of the ZIP file where the exported data is stored	`/tmp/export.zip` (default)	No
`tpname`	The trading partner name to be exported	Name of the trading partner	No
`tpanames`	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.	Agreement names must be separated by a comma	No
`active`	Exports agreements that have been deployed and are in active state.	`true` `false` (default)	No
`policies`	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. See "What Is Copied When You Import or Export from the Import/Export Tab" on page 10-3 for more information.	`true` `false` (default)	No

Example 18–5 Exports the Trading Partner Acme to /tmp/Acme.zip

```
ant -f ant-b2b-util.xml b2bexport -Dtpname="Acme" -Dexportfile="/tmp/Acme.zip"
```

Example 18–6 Exports an Agreement fro Design-Time with Listening Channel Details to /tmp/acmeGc.zip

```
ant -f ant-b2b-util.xml b2bexport -Dtpanames="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 -Dtpanames="Acme_GC_Agreement1, GC_Acme_
Agreement1" -Dactive=true
```

No listening channels are exported.

Deploying Agreements

Validates and deploys all agreements in the repository. If an agreement is deployed, then it is deployed again. The older version of the agreement is then in an inactive state. This feature is in preview mode for this release.

> **Note:** Validation can be turned off by enabling the property `oracle.tip.b2b.deploy.validation=false` in `b2b-config.xml`, which is found in
>
> *DOMAIN_HOME*`/config/soa-infra/configuration/`
>
> Turning off validation is useful when deploying large numbers of agreements, where you are certain that the data is valid. It requires a SOA Server restart.

```
ant -f ant-b2b-util.xml b2bdeploy
```

Table 18–4 Options for ant -f ant-b2b-util.xml b2bdeploy

Options	Description	Domain	Required
tpanames	One or more names of agreements to be deployed	Agreement names must be separated by a comma	No

Example 18–8 Deploys the Agreements Acme_GC_Agreement1 and GC_Acme_Agreement1

```
ant -f ant-b2b-util.xml b2bdeploy -Dtpanames="Acme_GC_Agreement1,GC_Acme_
Agreement1"
```

Validating B2B Metadata

Validates B2B metadata, including agreements, trading partners, and documents. All agreements are validated if no options are specified. This feature is in preview mode for this release.

```
ant -f ant-b2b-util.xml b2bvalidate
```

Table 18–5 Options for ant -f ant-b2b-util.xml b2bvalidate

Options	Description	Domain	Required
`args`	File names of the trading partner, agreement, or document protocol	File names must be separated by a comma	Yes

Example 18–9 Validates All Agreements

```
ant -f ant-b2b-util.xml b2bvalidate
```

Example 18–10 Validates Agreement tpa_ID1234.xml

```
ant -f ant-b2b-util.xml b2bvalidate -Dargs="tpa_ID1234.xml"
```

Example 18–11 Validates Trading Partner tp_MyCompany.xml and Agreement tpa_ID1234.xml

```
ant -f ant-b2b-util.xml b2bvalidate -Dargs="tp_MyCompany.xml,tpa_ID1234.xml"
```

CPP/CPA Templates

Creates a `cpp_cpa.properties` template file, which is used in the `propfile` option. This feature is in preview mode for this release.

```
ant -f ant-b2b-util.xml b2bcreate-cpaprop
```

Table 18–6 Options for ant -f ant-b2b-util.xml b2bcpaimport

Option	Description	Domain	Required
`propfile`	Property file that stores configuration details for `b2bcpaimport` and `b2bcpaexport`	-	Yes

Example 18–12 Creates a Property File Template That Is Used in the propfile Option

```
ant -f ant-b2b-util.xml b2bcreate-cpaprop
```

CPP/CPA Import

Converts an ebXML standard `cpa.xml` file to an Oracle B2B metadata file, which must then be imported into Oracle B2B. This feature is in preview mode for this release.

```
ant -f ant-b2b-util.xml b2bcpaimport

propfile
```

Table 18–7 Options for ant -f ant-b2b-util.xml b2bcpaimport

Option	Description	Domain	Required
`propfile`	Property file that stores configuration details for `b2bcpaimport` and `b2bcpaexport`	-	Yes

Example 18–13 Converts CPA-Formatted XML to an Oracle B2B ZIP File

```
ant -f ant-b2b-util.xml b2bcpaimport -Dpropfile="/tmp/cpp_cpa.properties"
```

CPP/CPA Export

Converts an Oracle B2B metadata file (data exported from Oracle B2B) to an ebXML standard `cpa.xml` file (a CPA-ready configuration). This feature is in preview mode for this release.

```
ant -f ant-b2b-util.xml b2bcpaexport
```

Table 18–8 Options for ant -f ant-b2b-util.xml b2bcpaimport

Option	Description	Domain	Required
`propfile`	Property file that stores configuration details for `b2bcpaimport` and `b2bcpaexport`	-	Yes

Example 18–14 Converts an Oracle B2B ZIP File to a CPA-Formatted XML File

```
ant -f ant-b2b-util.xml b2bcpaexport -Dpropfile="/tmp/cpp_cpa.properties"
```

Errors During Import

If you get the following broken pipe error, use Oracle WebLogic Server Administration Console to increase Maximum Message Size to 200000000

```
[java] Exception in thread "main" java.lang.Exception: java.rmi.UnmarshalException: Broken pipe;
nested exception is:
[java]  java.net.SocketException: Broken pipe
[java] at
oracle.tip.b2b.utility.B2BCommandLineUtility.upgradeRepository(B2BCommandLineUtility.java:548)
[java]  at oracle.tip.b2b.utility.B2BCommandLineUtility.main(B2BCommandLineUtility.java:601)
[java] Caused by: java.rmi.UnmarshalException: Broken pipe; nested exception is:
[java] java.net.SocketException: Broken pipe
```

19

Scripts for Archiving and Restoring Data

This chapter describes how to archive and restore B2B business messages using SQL scripts. These features are in preview mode for this release.

This chapter contains the following topics:

- Introduction to Archiving and Restoring B2B Business Messages
- Archiving B2B Business Messages
- Restoring B2B Business Messages

See Chapter 10, "Importing and Exporting Data," for information on importing and exporting design-time data.

Introduction to Archiving and Restoring B2B Business Messages

Oracle B2B uses Oracle Data Pump, an Oracle Database 11*g* feature that enables fast bulk data and metadata movement, to archive B2B run-time instance data *in Oracle databases*.[1]

You can specify criteria for archiving (and optionally purging) business messages based on start date, end date, and message state. The targeted business messages are marked with `JOB_ID`, a column in the B2B run-time tables that is used to synchronize archive and purge activity. B2B invokes the Data Pump PL/SQL API using `JOB_ID`. Hence, when you archive business messages, all the associated tables are also archived. Archived business messages can also be restored by using the Data Pump to import the run-time data into Oracle B2B (Oracle Metadata Service repository) and accessing it through B2B reports.

Archiving B2B Business Messages

To archive business messages, set up the archive directory and permissions and then run the archive procedure. The procedure provides an option to purge the archived rows.

To set up the archive directory and permissions:

1. On the computer running the database, create a directory for the archive file. For example,

   ```
   mkdir /tmp/archive
   ```

[1] For non-Oracle databases, external database archiving tools can be used to export and import run-time data.

2. Give permissions to this directory so that the database process can write to it. For example,

```
chmod 777 /tmp/archive
```

3. Log in to the database as `sysdba`.

```
sqlplus /as sysdba
```

4. Set up `B2B_EXPORT_DIR`.

```
SQL> create or replace B2B_EXPORT_DIR as '/tmp/archive'
```

5. Grant the SOA schema user (for example, `b2b_soainfra`) permission for the export.

```
SQL> grant read, write on directory B2B_EXPORT_DIR to b2b_soainfra;
SQL> grant exp_full_database to b2b_soainfra;
```

To archive, with an option to purge:

Set up the archive directory and permissions before using the following PL/SQL API.

1. Log in as the SOA schema user.

```
$ sqlplus b2b_soainfra/password
```

2. Execute the archive procedure, for example,

```
SQL> exec b2b_archive_procedure('21-JAN-2008','28-JAN-2008','MSG_
COMPLETE','JAN.dmp','N');
```

The signature of the procedure is

```
b2b_archive_procedure(fromDate, toDate, messageState, fileName, shouldPurge);
```

Table 19–1 lists the parameters for the `b2b_archive_procedure` API.

Table 19–1 b2b_archive_procedure Parameters

Parameter	Example	Description
`fromDate`	21-JAN-2008	Starting date for archival, `DD-MON-YYYY`
`toDate`	28-JAN-2008	Ending date for archival, `DD-MON-YYYY`
`messageState`	MSG_ COMPLETE	State of the business message. The `MSG_COMPLETE` state is typically archived. Other possible states are `MSG_ INVALID, MSG_CONTINUE_PROCESS, MSG_COLLAB_ WAIT, MSG_PROCESS_ACK, MSG_SEND_ACK, MSG_ WAIT_ACK, MSG_ERROR, MSG_WAIT_TRANSMIT, MSG_SEND_EXP, MSG_PROCESS_EXP, MSG_ABORTED, MSG_TRANSMITFAILED, MSG_WAIT_FA, MSG_SEND_ FA, MSG_WAIT_BATCH`
`fileName`	JAN.dmp	Name of the archive file to be created by the database. Ensure that a file with this name does not exist in the archive directory.
`should_purge`	N	`Y` removes the archived rows. The default is `N`.

Restoring B2B Business Messages

To restore business messages, set up the import directory and permissions and then run the restore procedure.

To set up the import directory and permissions:

1. On the PC running the database, create a directory for the import file.

   ```
   mkdir /tmp/import
   ```

2. Give permissions to this directory so that the database process can read from it.

   ```
   chmod 777 /tmp/import
   ```

3. Log in to the database as `sysdba`.

   ```
   sqlplus /as sysdba
   ```

4. Set up `B2B_IMPORT_DIR`.

   ```
   SQL> create or replace B2B_IMPORT_DIR as '/tmp/import'
   ```

5. Grant the SOA schema user (`b2b_soainfra`) permission for the export.

   ```
   SQL> grant read, write on directory B2B_IMPORT_DIR to b2b_soainfra;
   SQL> grant imp_full_database to b2b_soainfra;
   ```

To restore business messages:

Set up the import directory and permissions before using the following PL/SQL API.

1. Log in as the SOA infra schema user.

   ```
   $ sqlplus soa_infra_user/password
   ```

2. Execute the import procedure, for example

   ```
   SQL> exec b2b_restore_procedure('JAN.dmp');
   ```

   The signature of the procedure is

   ```
   b2b_restore_procedure(fileName)
   ```

   Use the **Reports** tab to search for and display the imported data.

20

Utilities for Enqueuing and Dequeuing

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.

This chapter contains the following topics:

- AQ Enqueue and Dequeue Utilities
- JMS Enqueue and Dequeue Utilities

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.

AQ Enqueue

Table 20–1 lists the Java AQ enqueue utility properties.

Table 20–1 IPEnqueue Properties

Name	Description
queue	The outbound AQ queue name. If unspecified, the Java enqueue utility uses the default outbound queue `IP_OUT_QUEUE`.
replyToMsgID	The message ID to which the sending message is replying, typically used for the response message type.
from	Trading partner that sends the message
to	Trading partner that receives the message
doctypeName	Document type name for the message
doctypeRevision	Document protocol revision for the message
payload	Payload file name
attachment	Attachment file name
url	The database URL format is `jdbc:oracle:thin:@`*`host:port:sid`*
user	The database user
password	The database password

Table 20–1 (Cont.) IPEnqueue Properties

Name	Description
eventName	Action name
msgID	Message ID (optional). B2B generates its own message ID if it is not provided as part of an enqueue.
msgType	Provide an optional message type: ■ Request = 1 (default) ■ Response = 2 ■ Functional Ack = 9

Example: `ipenqueue.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.

AQ Dequeue

To dequeue messages, use the `IPDequeue` utility.

Table 20–2 lists the Java AQ dequeue utility properties.

Table 20–2 IPDequeue Properties

Name	Description
queue	The inbound AQ queue name. If unspecified, the Java dequeue utility uses the default inbound queue `IP_IN_QUEUE`.
count	The number of messages to dequeue. If unspecified, only one message is dequeued.
output	Output file name
url	The database URL format is `jdbc:oracle:thin:@host:port:sid`
user	The database user
password	The database password

Example: `ipdequeue.properties:`

```
queue           =
count           = 1
output          = t1.trc
url             = jdbc:oracle:thin:@host:1521:sid
user            = user1
password        = password
```

JMS Enqueue and Dequeue Utilities

You can enqueue to and dequeue from a JMS destination (queue or topic) using utilities.

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 20–3 lists the properties that can be configured in the file.

Table 20–3 JMS Enqueue Properties

Name	Description
destination	JNDI name of queue or topic to send message to
cf	JNDI name of connection factory to use
factory	Factory provider class
isTopic	Indicator for topic (optional)
url	The database URL format is `jdbc:oracle:thin:@host:port:sid`
user	The database user
password	The database password
from	From party
to	To party
eventName	Action name
doctypeName	Document type name
doctypeRevision	Document type revision
payload	Payload file path
attachment	Attachment file path
msgID	Message ID (optional). B2B generates its own message ID if it is not provided as part of an enqueue.
replyToMsgID	Reply to message (optional)
msgType	Message type; the default is `Request` (optional).

Example 20–1 shows the sample `jms_enqueue.properties` file.

Example 20–1 Sample jms_enqueue.properties File

```
####### Destination Details        #######
destination = jms/b2b/B2B_IN_QUEUE
```

```
cf = jms/b2b/B2BQueueConnectionFactory

####### Server and Factory Details #######
factory=weblogic.jndi.WLInitialContextFactory
url=t3://stacz36:8001/
#user=<uncomment and provide you username>
#password=<uncomment and proivde 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.

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 20–4 are set as part of the `javax.jms.Message` delivered to the destination that B2B listens on.

Table 20–4 How AQ/JMS Properties Are Translated for Custom Utilities

AQ/JMS Utilities	Translated Value—For Custom Utilities	JMS Message
from	FROM_PARTY	Sent as a `string` type message property
to	TO_PARTY	Sent as a `string` type message property
doctypeName	DOCTYPE_NAME	Sent as a `string` type message property
doctypeRevision	DOCTYPE_REVISION	Sent as a `string` type message property
eventName	ACTION_NAME	Sent as a `string` type message property
msgID	MSG_ID	Sent as a `string` type message property
replyToMsgID	INREPLYTO_MSG_ID	Sent as a `string` type message property
msgType	MSG_TYPE	Sent as a `string` type message property
attachment	ATTACHMENT	Sent as a `string` type message property
payload	-	Sent as the message body

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 sample documentation for how to run these utilities.

Example 20–2 shows the sample JMS dequeue properties file.

Example 20–2 Sample jms_dequeue.properties File

```
####### Destination Details#######
destination = jms/b2b/B2B_IN_QUEUE
cf = jms/b2b/B2BQueueConnectionFactory
count=1

####### Server and Factory Details #######
factory=weblogic.jndi.WLInitialContextFactory
url=t3://stacz36:8001/
#user=<uncomment and provide your username>
#password=<uncomment and provide your password if required>
```

Part VI

Appendixes

This part contains the following appendixes:

- Appendix A, "Performance Tuning and Large Payloads"
- Appendix B, "Properties of b2b-config.xml"
- Appendix C, "Back-End Applications Interface"
- Appendix D, "Exception Handling"

A

Performance Tuning and Large Payloads

This appendix contains the following topics:

- Settings for Performance Tuning
  - Memory Arguments
  - Heap Size Settings
  - MDS Cache Size
  - Number of Threads
  - Stuck Thread Max Time
  - Tablespace
  - JTA Settings
- Handling Large Payloads
  - Introduction to Large Payload Support
  - Large Payloads and 32-Bit Windows PCs

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/need of the resource.

When using the large payload settings, the internal delivery channel must be the default channel or a JMS queue.

Changes to `b2b-config.xml` require a server restart. 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.

Memory Arguments

Memory arguments are captured in `DOMAIN_HOME/bin/setSOADomainEnv.sh`. Memory tuning applies to Oracle JRocket or SUN JVM.

For Oracle JRocket

```
export JAVA_VENDOR Oracle
DEFAULT_MEM_ARGS="-Xms1024m -Xmx1024m"
```

For Sun JVM

```
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"
```

`-Xms` and `-Xmx` can be increased up to 2 GB based on memory availability.

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.

MDS Cache Size

To set the Metadata Service (MDS) instance cache size, add the following property and value to `DOMAIN_HOME/config/soa-infra/configuration/b2b-config.xml`.

```
<property>
 <name>b2b.mdsCache</name>
 <value>200000</value>
 <comment>MDS Instance cache size </comment>
</property>
```

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.

Number of Threads

Changing the value of `threadCount` 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 `sleepTime` property puts a thread to sleep after message processing. A setting between 10 and 1000 (milliseconds) is recommended.

Set these values in *`DOMAIN_HOME`*`/config/soa-infra/configuration/b2b-config.xml` as follows:

```
<property>
  <name>b2b.inboundProcess.threadCount</name>
  <value>5</value>
  <comment></comment>
</property>
<property>
  <name>b2b.inboundProcess.sleepTime</name>
  <value>10</value>
  <comment></comment>
</property>
<property>
```

```
  <name>b2b.outboundProcess.threadCount</name>
  <value>5</value>
  <comment></comment>
</property>
<property>
  <name>b2b.outboundProcess.sleepTime</name>
  <value>10</value>
  <comment></comment>
</property>
<property>
  <name>b2b.defaultProcess.threadCount</name>
  <value>5</value>
  <comment></comment>
</property>
<property>
  <name>b2b.defaultProcess.sleepTime</name>
  <value>10</value>
  <comment></comment>
</property>
```

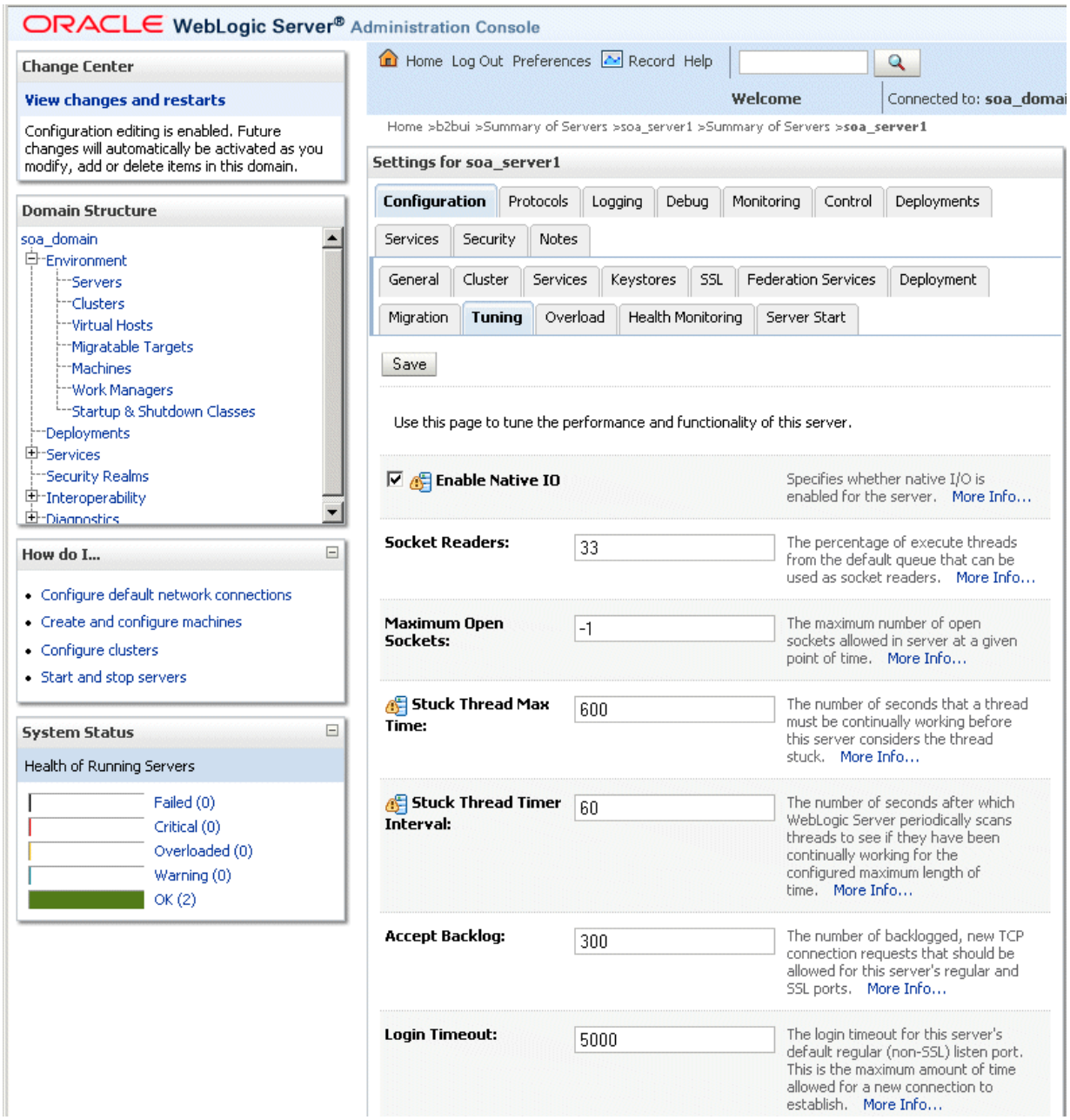
Stuck Thread Max Time

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



Tablespace

If you store more than an a 150 MG configuration, extend or add a data file to increase tablespace size as follows:

```
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;
```

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, a setting of 350 seconds is sufficient.

Handling Large Payloads

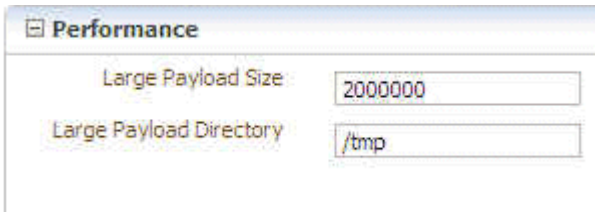
Oracle B2B can handle large payloads through the SOA Infrastructure and JMS internal queues.

Introduction to Large Payload Support

Inbound Setup

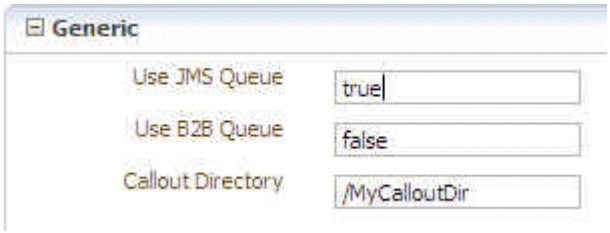
Figure A–2 shows the properties to set for inbound cases. Go to **Administration** > **Configuration** > **Performance**.

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** to true, as shown in Figure A–3. Go to **Administration** > **Configuration** > **Generic**.

Figure A–3 Use JMS Queue



With **Use JMS Queue** 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.

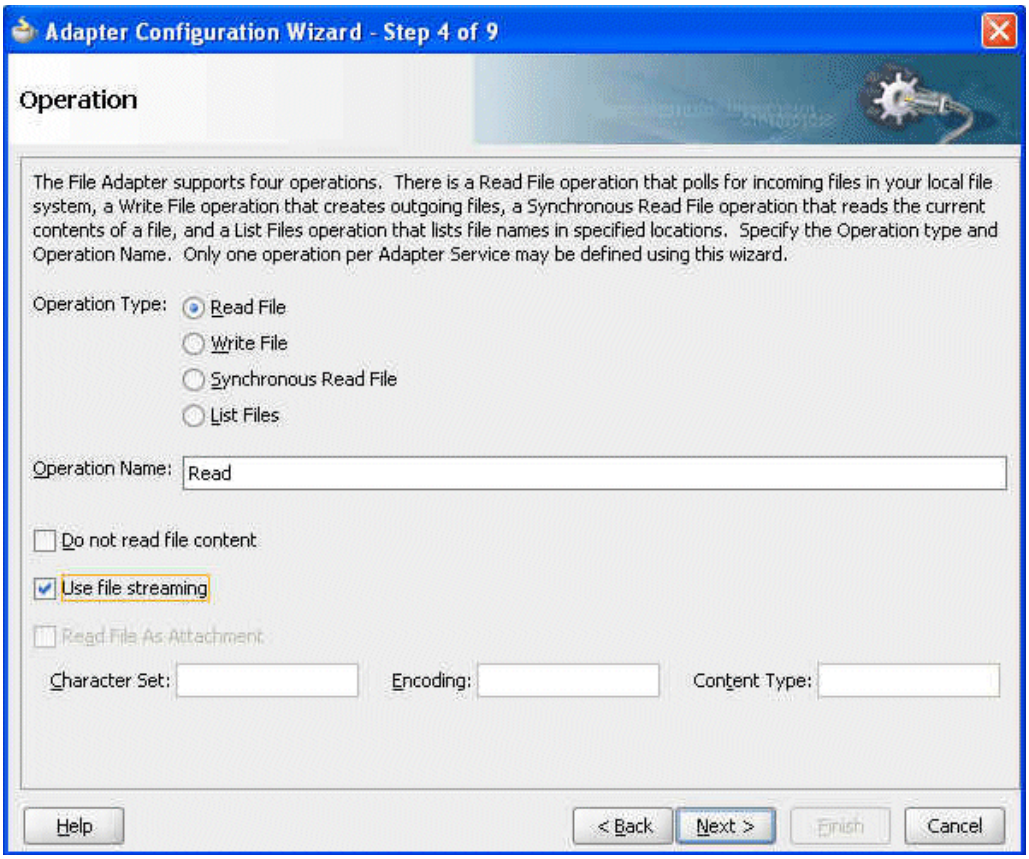
Figure A–4 Large Payload Directory

Notes

1. If you are doing large payload testing, set the logPayload property on the **Administration** > **Configuration** tab to false.
2. If you are doing large payload testing, set showpayload to false to avoid listing the payload in reports.
3. Increase the maximum heap size by using -Xmx2048m.
4. 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
   ```

5. 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
6. If Oracle B2B is used alone (without the SOA Infrastructure), the JTA timeout can be set in `b2b-config.xml` by using the advanced property `oracle.tip.b2b.jtaTimeout`.
7. For an outbound SOA composite, always select the **Use file streaming** option for the File Adapter.



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.

B

Properties of b2b-config.xml

Most B2B properties are set on the **Configuration** tab of the Oracle B2B interface. **Configuration** tab settings override properties set in `b2b-config.xml`. See Chapter 15, "Configuring B2B System Parameters."

This appendix contains the following topics:

- Turning off Validation During Deployment
- MDS Cache Size
- Number of Threads
- Setting Up File, FTP, or Email in an HA Environment
- Setting Internal Properties for a Functional Acknowledgment
- Setting b2b.FAHandleByB2B for EDI EDIFACT and EDI X12
- Setting the b2b.outboundOneErrorAllError Parameter

> **Note:** Changes to `b2b-config.xml` require a server restart.

Turning off Validation During Deployment

You can turn off validation during deployment by setting the property `b2b.deploy.validation=false` in `b2b-config.xml`. This is useful when deploying a large number of agreements where you are certain that the data is valid. Restarting the SOA Server is required.

MDS Cache Size

See "MDS Cache Size" on page A-2 for how to set the Metadata Service (MDS) instance cache size in `b2b-config.xml`.

Number of Threads

See "Number of Threads" on page A-2 for how to set the number of threads in `b2b-config.xml`.

Setting Up File, FTP, or Email in an HA Environment

To set up File, FTP, or Email transports in an HA environment, specify a unique name for each instance in `b2b.HAInstance`. If you use `#ServerName#` for the value, B2B retrieves the WebLogic Server name as the HAInstanceName.

Setting Internal Properties for a Functional Acknowledgment

To ensure that the ISA segment elements (1 - 4) of a 997 message generated by B2B are identical to the received 850, or any other transaction message, set the property `FAInternalProperties` to `true`.

Setting b2b.FAHandleByB2B for EDI EDIFACT and EDI X12

When the `b2b.FAHandleByB2B` property in `b2b-config.xml` 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, 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 `FAHandledByB2B` is true).

Setting the b2b.outboundOneErrorAllError Parameter

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. 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, all outbound batch messages with errors are flagged.

To set the OneErrorAllError parameter:

Use either of the following methods:

- In `b2b-config.xml`, add the following property:

```
<property>
  <name>b2b.outboundOneErrorAllError</name>
  <value>true</value>
  <comment>error all outbound batch messages on one error</comment>
</property>
```

- In Oracle Enterprise Manager Fusion Middleware Control, add the property to the b2b MBean by using the System MBean Browser. See Section 24.2, "Configuring B2B Operations" in *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite* for how to access the System MBean Browser.

C

Back-End Applications Interface

This appendix contains the following topics:

- Mapping B2B IP_MESSAGE_TYPE to SCA Normalized Message Properties

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 C–1 B2B IP_MESSAGE_TYPE to AS11 SCA Normalized Message Property Mapping

AQ (IP_MESSAGE_TYPE)	SCA	JMS
MSG_ID	b2b.messageId	MSG_ID
INREPLYTO_MSG_ID	b2b.replyToMessageId	INREPLYTO_MSG_ID
FROM_PARTY	b2b.fromTradingPartnerId	FROM_PARTY
-	b2b.fromTradingPartnerIdType	-
TO_PARTY	b2b.toTradingPartnerId	TO_PARTY
-	b2b.toTradingPartnerIdType	-
ACTION_NAME	-	ACTION_NAME
DOCTYPE_NAME	b2b.documentTypeName	DOCTYPE_NAME
DOCTYPE_REVISION	b2b.documentProtocolVersion	DOCTYPE_REVISION
-	b2b.documentProtocolName	-
-	b2b.documentDefinitionName	-
MSG_TYPE	b2b.messageType	MSG_TYPE
-	b2b.conversationId	-
PAYLOAD	body	-
ATTACHMENT	-	-

D

Exception Handling

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:

- Inbound Messages
- Outbound Messages
- Inbound Exception Handling Scenarios
- Exception Payload Definition

Inbound Messages

This section describes the following inbound message types:

- Request or Response Messages
- Acknowledgment Messages
- Exception Messages

Request or Response Messages

For an incoming request or response message that results in an exception, the following actions occur:

- An exception message is sent to the application.

  The exception message is enqueued to `B2B_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.

Acknowledgment Messages

For an incoming acknowledgment message that results in an exception, the following actions occur:

- An exception message is sent to the application.

  The exception message is enqueued to B2B_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.

Exception Messages

For an incoming exception message, the following actions occur:

- 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 B2B_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.

B2B errors cannot be delivered on other queues that you may configure (for example, an AQ or JMS queue).

Failures with 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.

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. The exception message is enqueued to B2B_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 `B2B_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.
- The error reports are updated to show only business messages; a business message is always created in the inbound and outbound directions.

Inbound Exception Handling Scenarios

Table D–1 describes inbound exception handling scenarios.

Table D–1 Inbound Exception Handling Scenarios

If an exception occurs because. . .	Then Oracle B2B does . . .
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

Table D–1 (Cont.) Inbound Exception Handling Scenarios

If an exception occurs because. . .	Then Oracle B2B does . . .
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 request messages only.
- No exception is sent back to the trading partner for response, acknowledgment, and functional acknowledgment messages.

Exception Payload Definition

The exception payload, `ipException.xsd`, is defined as follows:

```
<?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>
```

Index

A

Administration page, 2-5
Administrator role, 5-7
agreements
  active, 8-3
    exporting, 8-3
  creating, 6-2
  definition, 6-1
  deleting, 6-6
  deploying, 2-12, 6-5
  deployment states, 8-1
  functional acknowledgements, 6-4
  redeploying, 6-6
  trading partners, 2-11
  translation, 6-4
  validation, 6-4
Analyzer, 3-11
Applicability Statement 1 (AS1)
  channel, 5-11
  exchange protocol, 1-4
  identifier, 9-2
Applicability Statement 2 (AS2)
  exchange protocol, 5-11
  identifier, 9-2
application message reports, 16-7
AQ
  channel attributes, 5-17
  transport protocol parameters, 5-14
archiving B2B business messages, 19-1
AS1
  See Applicability Statement 1 (AS1) exchange protocol
AS2
  See Applicability Statement 2 (AS2) exchange protocol
Auto Create Agreement, 5-29

B

B2B IP_MESSAGE_TYPE, C-1
B2B transaction, creating, 2-8
b2b-config.xml file, 1-8, A-1, A-2, B-1, B-2
b2bcpaexport command line tool, 18-6
b2bcpaimport command line tool, 18-5
b2bcreate-cpaprop command line tool, 18-5
b2bdeploy command line tool, 18-4
b2bexport command line tool, 18-3
b2b.FAHandleByB2B property, B-2
B2BHeader.sendException flag, D-1, D-2
b2bimport command line tool, 18-2
b2b.outboundOneErrorAllError parameter, B-2
b2bpurge command line tool, 18-2
B2BUser, 10-3
b2bvalidate command line tool, 18-4
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 messages
  archiving, 19-1
  restoring, 19-2
business-to-business e-commerce
  comparison to traditional transactions, 1-2
  description, 1-1

C

Callout Directory parameter, 12-4
callouts
  code example, 12-6
  creating, 12-3
  creating a library JAR file, 12-3
  defining callout .jar file library location, 12-4
  definition, 12-1
  differently formatted XML messages, 12-2
  implementation class, 12-4
  implementing, 12-6
  including in an agreement, 12-5
  library name, 12-4
  parameter attributes, 12-4
  predefined XSLTCalloutImpl class file, 12-4
channel attributes
  AQ, 5-17
  configuring, 5-16
  email, 5-16
  file, 5-17
  FTP, 5-17
  JMS, 5-17

MLLP, 5-17
SFTP, 5-17
channel details, 5-21
CLIENT-CERT authentication method, 1-6
cloning trading partners, 5-2
command line tools
b2bcpaexport, 18-6
b2bcpaimport, 18-5
b2bcreate-cpaprop, 18-5
b2bdeploy, 18-4
b2bexport, 18-3
b2bimport, 18-2
b2bpurge, 18-2
b2bvalidate, 18-4
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, 18-5
templates, creating, 18-5
custom contact information types, creating, 9-4
custom document protocols
configuring XPath expressions, 7-5
definition, 7-2
document definition parameters, 7-3
document protocol parameter types, 7-31
document type parameters, 7-2
document version parameters, 7-2
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, 8-3
inactive, 8-2
purged, 8-2
retired, 8-2
deployed agreements
managing, 8-1
metrics, 2-7
searching, 8-2
deploying agreements, 18-4
dequeuing
See Java dequeue command
design metadata, 13-1
design-time repository
description, 10-1
exported file, 10-4
exporting, 10-2
importing, 10-2
what is imported or exported, 10-3
document definitions
adding, 5-9
changing after importing metadata, 7-33
creating, 2-10, 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 protocol parameter types, 7-31
document protocols, 1-4
custom, 7-2
definition, 7-1
EDI, 3-1
EDI EDIFACT, 7-7
EDI X12, 4-2, 7-12
EDIEL, 1-3, 1-4
HL7, 1-2, 1-8, 3-1, 7-16
introduction, 4-1
managing, 2-5
NCPDP Telecom, 1-3, 1-4, 3-1
OAG, 1-2, 1-3, 1-4, 7-20
ParserSchema, 3-1
Partner Interface Process (PIP), 7-24
Positional Flat File Document Protocol, 7-22
proprietary transactions, 7-2
RosettaNet, 1-2, 1-3, 1-8, 3-1, 7-24
supported, 1-2
UCCnet, 1-2, 1-3, 1-4, 7-29
Document Routing IDs, 7-33
DUNS, 9-2

E

ebMS Identifier, 9-4
ebMS-1.0, 1-3, 5-11, 5-21
ebMS-2.0, 5-11, 5-21
ebXML messaging service (ebMS) parameters, 7-2
EDI
document guidelines, 3-1
Group ID, 9-2
Group ID Qualifier, 9-2
Interchange ID, 9-2
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 protocol parameter types, 7-31
document protocols, 7-7
document type parameters, 7-9
document version parameters, 7-7
Oracle B2B support, 1-3
samples, 1-8
setting FAHandleByB2B property, B-2
traditional transaction comparison, 1-2
UNA, 7-8

EDI X12
definition, 7-12
document definition parameters, 7-14
document hierarchy, 4-2
document protocol parameter types, 7-31
document type parameters, 7-14
document version parameters, 7-12
Oracle B2B support, 1-3
samples, 1-8
setting FAHandleByB2B property, B-2
traditional transaction comparison, 1-2
EDIEL
Oracle B2B support, 1-3
preview mode, 1-4
Email
AS1-1.0 exchange protocol parameters, 5-19
channel attributes, 5-16
transport protocol parameters, 5-13
enqueue, JMS JCA adapter, 20-4
enqueuing
See Java enqueue command
enqueuing and dequeuing, 20-1
error message reports, creating, 16-10
errorDescription, D-1
errorText, D-1
exception handling
description, D-1
inbound exception handling scenarios, D-3
inbound messages, D-1
acknowledgement messages, D-2
document identification failure, D-4
document processing failure, D-4
ebMS, AS1, and AS2 failures, D-2
exception messages, D-2
exchange identification failure, D-3
exchange not supported, D-3
message decoding failure, D-3
message duplicated, D-3
message unpacking failure, D-3
request or response messages, D-1
scenarios, D-3
trading partner agreement failure, D-4
outbound messages, D-2
payload definition, D-4
exchange protocol parameters
configuring, 5-17
email, AS1-1.0, 5-19
HTTP, AS2-1.1, 5-18
HTTP, ebMS-1.0, 5-18
HTTP, ebMS-2.0, 5-18
TCP, MLLP-1.0, 5-18
exporting
active agreement, 8-3
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

FAInternalProperties property, B-2
file channel attributes, 5-17
file transport protocol parameters, 5-14
FTP
channel attributes, 5-17
transport protocol parameters, 5-15
Functional Acknowledgment, setting internal properties, B-2

G

Generic AQ-1.0, 1-3, 5-11
Generic Email-1.0, 1-3, 5-12
Generic File-1.0, 1-3, 5-11
Generic FTP-1.0, 1-3, 5-11
Generic HTTP-1.0, 1-3, 5-12
Generic Identifier, 9-3
Generic JMS-1.0, 1-3, 5-12
Generic SFTP-1.0, 1-3, 5-11

H

HA environment
email transport, B-1
File transport, B-1
FTP transport, B-1
HAInstanceName, B-1
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, 7-16
document definition parameters, 7-19
document guidelines, 3-1
document protocol parameter types, 7-31
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, 9-3
Message Facility ID, 9-3
Message Facility Universal ID, 9-4
Message Facility Universal ID Type, 9-4
negative acknowledgment messages, 7-20
Oracle B2B support, 1-3
samples, 1-8
traditional transaction comparison, 1-2
Host Administrator role, 1-5
Host Monitor role, 1-5

host trading partner, 5-1
agreements, 6-3
key store information, 5-6
HTTP
AS2-1.1 exchange protocol parameters, 5-18
ebMS-1.0 exchange protocol parameters, 5-18
ebMS-2.0 exchange protocol parameters, 5-18
transport protocol parameters, 5-13

I

Identity Store, creating a new user, 5-7
import errors, 18-6
importing
CPP/CPA files, 18-5
data, 18-2
description, 10-1
design-time repository from file, 10-1
instance data, 2-6, 13-1
instance metadata, 13-1
ipException.xsd file, D-2
enqueued exception, D-1, D-2, D-3
exception payload definition, D-4

J

Java dequeue command, 20-1
Java enqueue command, 20-1
JMS
channel attributes, 5-17
dequeue properties, 20-4
enqueue and dequeue, 20-3
enqueue properties, 20-3
JCA adapter, 20-4
transport protocol parameters, 5-15

L

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, 14-8
adding details, 14-4
adding exchange protocol parameters, 14-9
adding security parameters, 14-9
adding transport protocol parameters, 14-8
configuring, 14-7
definition, 14-1
protocols, 14-2
logging in, user interface tool, 2-2
logPayload property, A-6

M

message exchange protocol
AS1, 1-4
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
messages, purging, 16-2
Metadata Service (MDS)
instance cache size, B-1
managing metadata, 1-5
repository, 1-5
metrics
introduction, 17-1
summary data, 17-3
Metrics page, 2-7
MIME, 1-2
Minimum Lower Layer Protocol (MLLP)
channel attributes, 5-17
delivery channels, 5-20
description, 5-11
ID, 9-4
message exchange protocol, 1-3
transport protocol parameters, 5-14
Monitor role, 5-7, 16-1

N

NCPDP Telecom
document guidelines, 3-1
Oracle B2B support, 1-3
preview mode, 1-4
number of threads, B-1

O

OAM-SSO protected URLs, 1-6
Open Applications Group (OAG)
document definition parameters, 7-21
document protocol parameter types, 7-31
document type parameters, 7-20
document version parameters, 7-20
introduction, 7-20
Oracle B2B support, 1-3
preview mode, 1-4
traditional transaction comparison, 1-2
Oracle B2B
adding a channel, 5-12
adding a user, 5-8
Administration page, 2-5
as a binding component, 1-6
available channels, 5-10
channel attributes, 5-16
channel details, 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, 1-6
in JDeveloper environment, 2-13
introduction, 1-1
logging in, 2-2
metadata, 1-5
metrics, 17-1
Metrics page, 2-7
partner metrics, 17-3
Partners page, 2-5
port information, 2-3
process flow, 3-1
Reports page, 2-6
resubmitting messages, 16-2
samples, 1-8
security features, 1-5
supported protocols, 1-2
trading partners, 5-1
transport protocols, 14-3
user interface, 2-4
user interface, accessing, 2-2
Oracle B2B Document Editor, 2-1, 2-9
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, 1-7
Oracle Business Activity Monitoring (BAM), 1-7
Oracle Data Pump, 19-1
Oracle Enterprise Manager Fusion Middleware Control, 1-8
Oracle JDeveloper, 2-1
Oracle Mediator, 1-7
Oracle Metadata Service repository, 19-1
Oracle Platform Security Services, 1-5
Oracle SOA Suite overview, 1-6
Oracle WebLogic Server Administration Console
administering Oracle B2B, 1-8
Security Realms, 5-7

P

ParameterValue, 10-3
ParserSchema, document guidelines, 3-1
Partner Interface Process (PIP)
document definition parameters, 7-26
document type parameters, 7-25
document version parameters, 7-25
introduction, 7-24
partner metrics, 17-3
Partners page, 2-5
performance tuning, 1-8
description, A-1
heap size settings, A-2
JTA settings, A-5
large payloads, A-5
MDS cache size, A-2
memory arguments, A-1
number of threads, A-2
Stuck Thread Max Time, A-3
tablespace, A-4
positional flat files
document definition parameters, 7-22
document protocol parameter types, 7-32
document type parameters, 7-22
document version parameters, 7-22
introduction, 7-22
predefined identifier types, 9-2
preview mode, 1-4
process flow, overview, 2-8
proprietary transactions, 7-2
protocols
Custom, 7-2
document, 7-1
EDI EDIFACT, 7-7
EDI X12, 7-12
HL7, 7-16
OAG, 7-20
positional flat file, 7-22
RosettaNet, 7-24
UCCnet, 7-29
purging
data, 18-2
design metadata, 13-1
messages, 16-2

R

reference, 1-6
Remote Administrator role, 1-5
Remote Monitor role, 1-5
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, 2-6
restoring B2B business messages, 19-2
resubmitting messages, 16-2
roles
Administrator role, 5-7
Host Administrator, 1-5
Host Monitor, 1-5
Monitor role, 5-7
Remote Administrator, 1-5
Remote Monitor, 1-5
RosettaNet
Business Directory, 7-28
definition, 7-24
dictionaries, 7-24
document guidelines, 3-1
document protocol parameter types, 7-32

document type definition (DTD), 7-24
Message Guideline specification, 7-28
Oracle B2B support, 1-3
Partner Interface Process (PIP), 7-24
samples, 1-8
Technical Directory, 7-28
traditional transaction comparison, 1-2
validation, 7-28
RosettaNet-01.10, 1-3, 5-11, 5-21
RosettaNet-V02.00, 1-3, 5-11, 5-21
run-time
data, 2-6
metadata, 13-1

S

sample SOA implementation, 1-6
SCA normalized message properties, C-1
security
configuring, 5-19
features, 1-5
Security Realms, 5-7
SegmentCount macro, 3-9
service, 1-6
setSOADomain.sh script, A-2
SFTP
channel attributes, 5-17
transport protocol parameters, 5-15
SMIME, 1-2
SOA composite application, 1-6
SOA Server, 1-8
SOAP, 1-2
supported protocols, 1-2
system parameters, setting, 15-1

T

TCP, MLLP-1.0 exchange protocol parameters, 5-18
trading partner agreements
creating, 2-11, 6-1, 6-2
definition, 6-1
deleting, 6-6
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, 2-5, 2-11
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 protocol parameters
AQ, 5-14
configuring, 5-13
email, 5-13
file, 5-14
FTP, 5-15
HTTP, 5-13
JMS, 5-15
MLLP, 5-14
SFTP, 5-15
transport protocols
available in Oracle B2B, 14-3
using, 14-3

U

UCCnet
definition, 7-29
document definition parameters, 7-29
document protocol parameter types, 7-32
document type parameters, 7-29
document version parameters, 7-29
Oracle B2B support, 1-3
preview mode, 1-4
traditional transaction comparison, 1-2
UNA, 7-8
user interface
accessing, 2-2
main page, 2-3
users
accessing user interface tool, 2-2
admin user name, 2-2

V

validating B2B metadata, 18-4
validation, turning off during deployment, B-1

X

XMLDSig, 1-2
XMLEncrypt, 1-2
XMLSchema, document guidelines, 3-1
XSLTCalloutImpl class file, 12-4