Oracle® Fusion Middleware

Administrator's Guide for Oracle Business Data Synchronization
Server

11*g* Release 1 (11.1.1.7.0)

E12319-03

March 2013

Documentation for developers that describes how to install,
configure, and administer connectors for Oracle Business
Data Synchronization Server (BDSS) that enable
synchronized interactions between Personal Information
Management (PIM) data stores.

ORACLE®

Oracle Fusion Middleware Administrator's Guide for Oracle Business Data Synchronization Server 11*g* Release 1 (11.1.1.7.0)

E12319-03

Primary Author: John Bassett

Contributors: Travis Dutchover, Dennis Ferry, Larry Harding, Owen Wessling

Contents

3 Installing the BDSS Components

Part II   Basic Administration

4 Getting Started with Administering BDSS

5   Configuring the BDSS Hub Services

6   Configuring the BDSS Connector for Microsoft Exchange 2007

7 Configuring Security for BDSS

8 Mapping Connector Fields to Hub Fields

9 Synchronizing the Calendar Domain

Part III   Appendixes

A BDSS Configuration Operations

B   Tables in the BDSS Data Stores

C   Connector API

D   Custom Field XSLTs and XSDs

E   Troubleshooting BDSS

x

Preface

Welcome to the *Administrator's Guide for Oracle Business Data Synchronization Server*.

Audience

This document is intended for administrators, application and process designers.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents:

- *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite*
- *Oracle Fusion Middleware Securing Web Services for Oracle WebLogic Server*

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 the text or 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

This part presents introductory and conceptual information about Oracle Business Data Synchronization Server (BDSS) and contains the following chapters:

1

Introduction to Oracle Business Data Synchronization Server

This chapter provides a general description of Oracle Business Data Synchronization Server (BDSS)

This chapter includes the following topics:

- What Is Oracle Business Data Synchronization Server?
- Using BDSS
- Configuring the Exchange 2007 Connector to Test Domain Synchronization

1.1 What Is Oracle Business Data Synchronization Server?

Oracle Business Data Synchronization Server (BDSS) is a synchronization service that enables server data stores to exchange and compare data. BDSS focuses primarily on interactions between Personal Information Management (PIM) data stores, such as Microsoft Exchange. For example, when a user changes a record for a particular **domain** of a **connector** (a connector domain), such as a *ToDo* item on one PIM server, BDSS automatically synchronizes that change to a corresponding *ToDo* item on another PIM server. Figure 1–1 illustrates an example of a BDSS installation, one that includes Microsoft Exchange interacting with other types of PIM data stores.

> **Note:** Although BDSS emphasizes the use of PIM data stores, you can extend BDSS for other types of data stores.

Figure 1–1   Oracle Business Data Synchronization Server
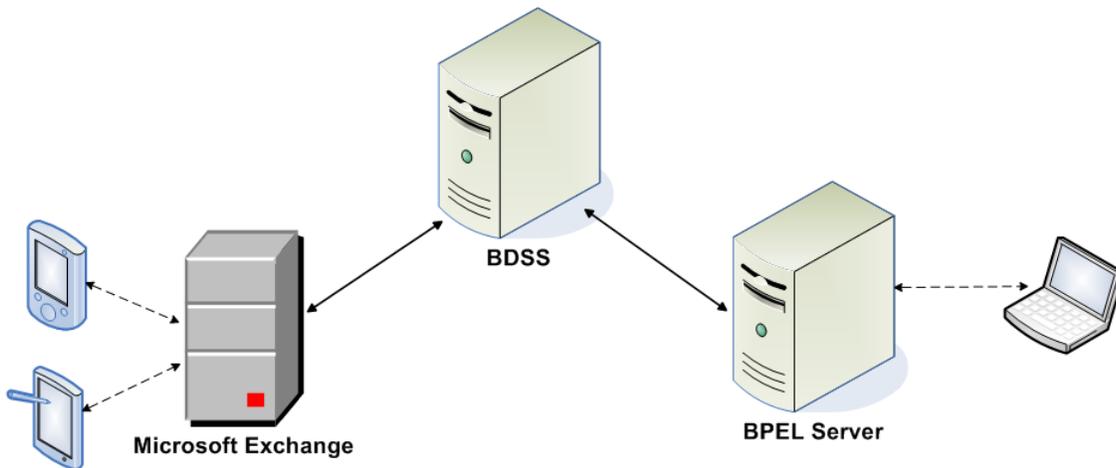
1.2 Using BDSS

This section provides usage scenarios for synchronizing tasks and contacts.

> **Note:**   BDSS is intended as a background process. As such, users are not required to perform any explicit action for synchronization. Users are aware of BDSS only when they notice that the records they create and modify in one system propagate to other systems.

1.2.1 Creating a Task

An Exchange 2007 user creates a new task through the Microsoft Outlook client. After the user saves the task, Outlook synchronizes the task to the Exchange server hosting the user's mailbox. Later, when the user opens her account on another PIM server, the user notices that the task she created using the Outlook client appears in the *ToDo* list. When the user updates this task, BDSS updates the task in the Microsoft Exchange data store so that the user's Outlook client displays the updated task.

1.2.2 Storing Contact Information

When an Exchange 2007 user creates a new contact, BDSS synchronizes the contact record, writing it to the user's Outlook Contacts lists. When the user changes the spelling of a contact in Outlook, BDSS writes the change to the user's account on other PIM servers. After the record is written to this PIM server, all users that appear on the access list have the name change propagated to their Outlook Contacts list.

1.3 Configuring the Exchange 2007 Connector to Test Domain Synchronization

You can test the Hub's synchronization of the Task, Contact, and Calendar domains using the Exchange 2007 Connector, because it provides synchronization for multiple Hub domains. You can configure the Exchange 2007 Connector to present itself as any type of PIM server to the Hub, because server type is irrelevant to the Hub. If you configure a second Exchange 2007 Connector using the same PIM server type, the Hub can then synchronize between two different Exchange users as if it were synchronizing between two users of different PIM servers.

To test Hub synchronization:

1. Create a second connector. Differentiate this connector by giving a name such as *Exchange2007_2*.

2. Create a PIM server instance and a **connector instance** for this connector.

3. Configure the PIM Transport Implementation Class and Extract Response Batch Size parameters for each Exchange 2007 Connector whereby the value for the Profile is the name of the connector as it appears in the NAME column of the CONNECTORS table. For example, suppose you have deployed a single Exchange 2007 Connector and want to use the single instance to synchronize two Exchange 2007 users that are mapped to a single **Hub user**. Further, assume that Exchange2007 and Exchange2007_2 are in the NAME column of the CONNECTORS table and both have a SERVER_TYPE_CD corresponding to the Exchange2007 LOV value.

4. Add the following four rows to the PROFILES table using the `setProfileParameter` of the ProfileRuntimeInfo MBean. For more information, see Section 4.8, "Managing Profiles."

Table 1–1   Adding Profiles to Test Synchronization

Profile	Section	Parameter	Value
Exchange2007	Hub Transport	Pim Transport Implementation Class	oracle.bdss.connectors.exchange2007
			pimtransport.PimTransport Adapter
Exchange2007	Hub Transport	Extract Response Batch Size	100000
Exchange2007_2	Hub Transport	Pim Transport Implementation Class	oracle.bdss.connectors.exchange2007.pimtransport.PimTransportAdapter
Exchange2007_2	Hub Transport	Extract Response Batch Size	100000

5. Create a **connector user** for each of the connectors.

> **Note:** You must use different Exchange users. That is, you cannot map *john@exchange.com* to *john@exchange.com*.

2

Understanding the BDSS Architecture

This chapter provides a high-level description of the Oracle Business Data
Synchronization Server (BDSS) architecture

This chapter includes the following topics:

- Overview of the Oracle Business Data Synchronization Server Architecture
- Overview of Connectors

2.1 Overview of the Oracle Business Data Synchronization Server Architecture

Oracle Business Data Synchronization Server (BDSS) utilizes a hub-and-spoke
architecture that enables synchronization between PIM servers. The hub provides the
core synchronization functions and is PIM-server agnostic, enabling connections for
any type of PIM server. Connectors, which send data to and retrieve data from the PIM
servers, comprise the spokes of the system. Figure 2–1 illustrates an example of this
architecture, one that results in fewer connectors between PIM servers. Even in
multi-server topologies, this simpler synchronization topology addresses the potential
issue of data feedback looping between the PIM servers.

Figure 2–1   An Example of the BDSS Hub-and-Spoke Architecture
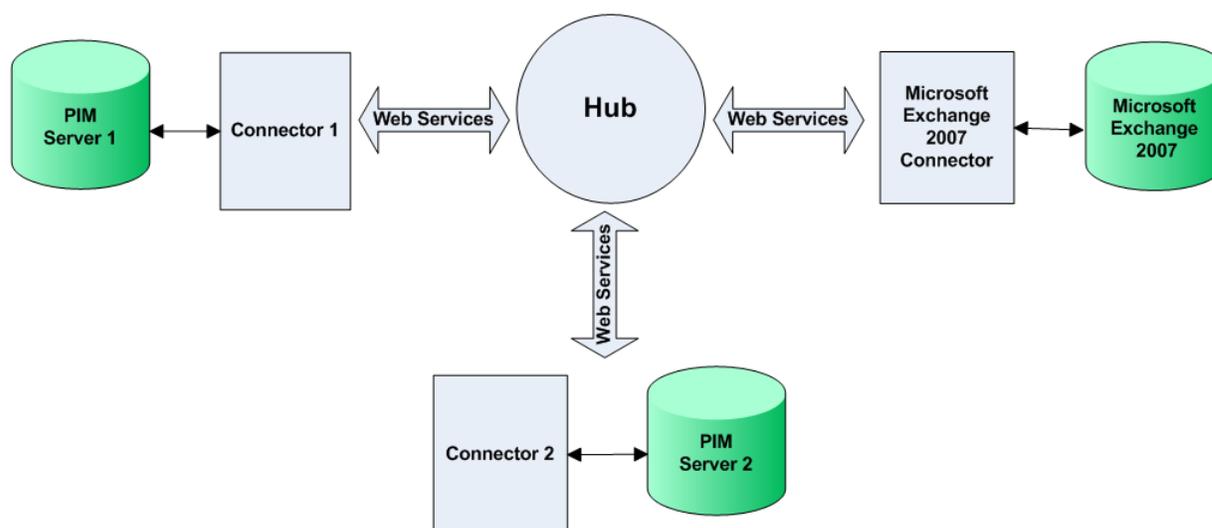
> **Note:** For Oracle Fusion Middleware 11*g* release 1, BDSS ships with
> support for Microsoft Exchange.

2.1.1 BDSS Components

BDSS consists of the following two main components:

- Hub

- Connectors

2.1.1.1 Hub

The Hub, which orchestrates the synchronization of data for the system, includes the
following subsystems (illustrated in Figure 2–2):

- Dispatcher

- Engine

Figure 2–2   The Subsystems of the Hub
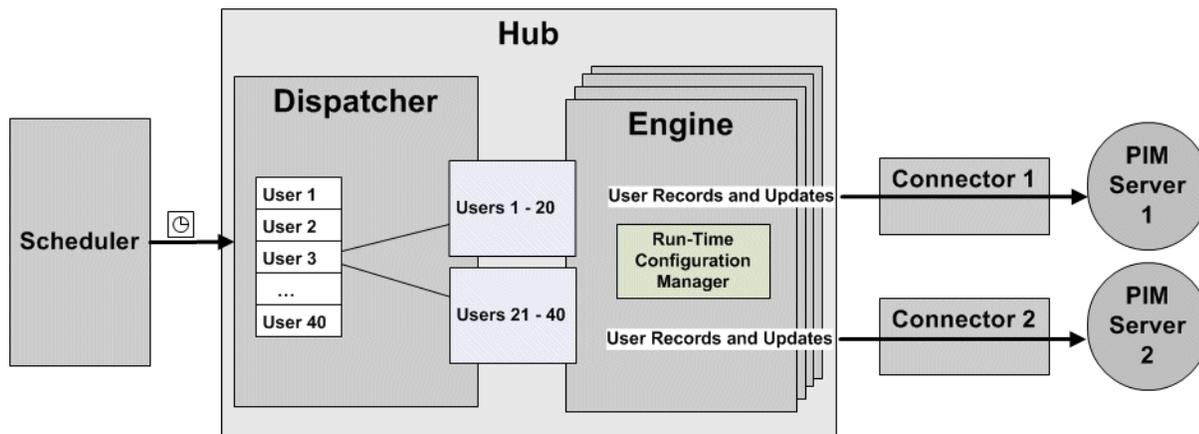
Dispatcher

The Dispatcher first reads the set of users that it must send to the Engine and verifies
that users on the list are available for synchronization. After it is called, the Dispatcher
process divides the entire set of users for synchronization into batches and calls the
Engine component. This subdivision of the user list allows for improved scalability of
the system, as several Engine components can be made available. Dispatcher calls can
be balanced across the available Engine components as well.

> **Note:** The Dispatcher does not send users to the Engine when those
> users are currently undergoing synchronization.

Scheduler

An external scheduler runs the Dispatcher at set time intervals. BDSS does not ship
with a scheduler. You can use any scheduling service, such as the Windows scheduler
service or the Oracle Enterprise Manager scheduler.

> **Note:** Although Oracle suggests a default interval of 5 minutes, you can select a scheduling interval that provides performance and scalability to suit the environment.

Engine

The Engine receives the list of users from the Dispatcher and synchronizes each of these users at the same time. The Engine contacts each of the connectors to receive from the PIM servers a set of records that have changed for the user. The extraction sets are combined to form a complete set of changes to be pushed back to the PIM servers.

> **Note:** The Engine only notifies a PIM server about records that have changed on other PIM servers. It does not inform the PIM server of changes identified from its own extraction set.

A conflict can arise if both PIM servers change a record during the same synchronization cycle. The Engine determines which changes to keep and which to discard based on the relative priority of each connector domain involved. (You can configure the priority assigned to each connector domain.) After the Engine determines which records comprise the result set, it pushes them to the PIM servers and then tracks the user's success.

The Hub itself does not read the data that it receives from the connectors. Instead, it reads the ID tag for each record and builds a record map that correlates the records of the different PIM servers. The Hub passes the data to the connectors.

The Engine also includes the Run-Time Configuration Manager, which stores and retrieves the data required for synchronizing users. This data includes component configuration settings, user information, mappings, user record information, and information about PIM server instances.

2.1.1.2 Connectors

Connectors standardize communication between the Hub and the PIM server that they service. Connectors update the data for connector domains, such as a contact or a task. During a synchronization session, the connectors retrieve changed data for a user from their respective PIM servers and transform it into the Hub format. They then pass the transformed data to the Hub. The connectors also transform the data from the Hub into the format appropriate to the PIM server. The connectors use Web services to both retrieve user records and return them to the Engine. For more information, see Section 2.2, "Overview of Connectors."

For Oracle Fusion Middleware 11*g* release 1, BDSS ships with Oracle BDSS Connector for Microsoft Exchange 2007 (the Exchange 2007 Connector). The Exchange 2007 Connector uses the Exchange 2007 Exchange Web Service (EWS) when it interacts with a user's Microsoft Exchange mailbox. The Exchange 2007 Connector supports synchronization of Calendar and Task data.

> **Note:** Using the BDSS Connector API, you can create connectors that synchronize any domain, such as *service requests* or *opportunities*. For more information, see Appendix C, "Connector API."

2.1.2 How BDSS Synchronizes Users

BDSS works on a set of users that are configured through MBeans (managed beans) by mapping a user login from each of the end systems to a central representation of the user called a **Hub user**. The Dispatcher can be triggered either by the schedule, or when a connector sends an event notification to the Hub when a user has changed a record. Scheduler-based synchronization synchronizes all users at each interval, even if some users have no changes. Event-based synchronization is more efficient, as users only synchronize when there is data to be synchronized.

2.1.2.1 Synchronization Options

You can set the following synchronization options for each user on each connector domain. For example, if synchronization is enabled for task items for a user, then BDSS automatically synchronizes the task for that user.

Inbound

The inbound direction limits the exchange of data for an entity (such as a domain, user, or a connector) to only pushing records and record updates to the PIM server. Changes to the records on the PIM server are not propagated to other systems.

Outbound

Outbound direction enables the connector to extract Create, Update, and Delete record event data from the PIM server. The outbound direction limits the synchronization for an entity by only allowing changes to be collected from a PIM server. No records or changes are pushed to a PIM server.

None

No synchronization of data occurs.

Full

Full synchronization enables bidirectional (that is, both inbound and outbound) synchronization.

You can configure the synchronization direction for any connector, domain, and user. For more information, see Section 4.1.2, "Specifying the Synchronization Direction."

2.1.2.2 Scheduler-Based Synchronization

A typical flow demonstrating a synchronization session between a BPEL server and the Microsoft Exchange server is as follows:

1. The Scheduler triggers the Dispatcher.

2. The Dispatcher searches for Hub users who are available for synchronization, meaning those users who, despite having been enabled for synchronization, are not currently synchronizing.

3. The Dispatcher chunks these users into batches.

4. The Dispatcher sends the batches to the Engine.

5. The Engine starts a synchronization session for each user.

6. The synchronization session performs the following operations:

   a. Determines the appropriate PIM servers for the user.

   b. Determines the domains (such as Tasks or Contacts) for the user.

    c. Sends an extract message to the connector instance for each connector. This extract message lists the user domains for which record changes are requested.

7. The connectors return the extract response to the Engine. Each extract response contains information describing new, updated, and deleted records for a particular connector domain.

8. The Engine merges the extract response records, resolving conflicts as needed.

9. The Engine sends the appropriate record changes to each PIM server.

10. The user's synchronization process terminates.

See also Section C.1.9, "Synchronization Session."

2.1.2.3 Tracking Record Changes Using Synchronization State

The Engine tracks the user's domain records that have been exported successfully from a PIM server using connector-provided data known as the synchronization state. Synchronization state is monitored at the connector domain level. For more information about synchronization state, see Appendix C, "Connector API."

2.1.3 Event-Based Synchronization

Event-based synchronization reduces the amount of processing required of the Hub. It also reduces network and database usage by eliminating unneeded user synchronization sessions. Using event-based synchronization, the Hub synchronizes users only when record changes occur. If there are no changes for a user, then the Hub does not synchronize the user.

User changes that affect the domains do not trigger synchronization sessions; as in standard synchronization (described in Section 2.1.2), the scheduler dictates the frequency of the synchronization sessions. The event-based synchronization process is initiated when a PIM server notifies its connector that a user's records have changed. The connector then notifies the Hub and sets an event flag that identifies the user as having changed records on the PIM server. When the Scheduler triggers a new synchronization cycle, the Dispatcher batches this user to the Engine to undergo synchronization.

BDSS works with connectors and PIM servers that can flag users for event-based synchronization and with PIM servers that cannot flag users for event-based synchronization. In systems where all of the connectors and PIM servers can flag users for event-based synchronization, the Dispatcher batches users with change flags. When synchronization is started for theses users, the Hub synchronizes them as it would in a scheduled synchronization session (that is, the Hub requests changes from the connector, since changes may have occurred to a user's PIM data after the Hub received the connector's initial notification). The Engine does not process users or send an extract message to the connector during the synchronization cycle. During these synchronization sessions, the synchronization direction is Inbound Only for the unflagged user. In systems having connectors that cannot flag events, the Hub synchronizes all of the users that belong to these connectors for event-based synchronization.

If a BDSS connector cannot support sending event notifications, then the Hub cannot support the event-based method to initiate user synchronization sessions. For example, if changes were made in a system where a connector was unable to detect changes, the Hub could not synchronize the records for a user unless a change to a record was made in a system that has a connector that supports event notification for that same user.

Each time BDSS starts (or restarts), it sets change flags for users who synchronize on all of the domains that are set for either Full or Inbound synchronization. These flags enable connectors to create subscriptions for the user. A connector creates separate subscriptions for each domain if it can to do so.

> **Note:** You enable event-based synchronization by setting the SYNC_ EVENTS_ENABLED_FLG, SUPPORTS_EVENTS_FLG, and USER_ EVENT_FLG flags in the CONNECTORS table.

The Event-Based Synchronization Flow

1. The Dispatcher retrieves Hub users who are available for synchronization.

2. The Dispatcher chunks these users into batches.

3. The Dispatcher sends these users to the Engine.

4. The Engine starts a synchronization session for each user. The synchronization session performs the following:

   a. Determines the appropriate PIM servers for user.

   b. Determines if there are changes made to the user's domains (such as Tasks or Contacts).

   c. Sends an extract message to the connector instance for each connector. This extract message lists the user domains for which record changes are requested.

   d. The connectors return the extract response to the Engine.

   e. The Engine processes the extract responses and then pushes data to the connectors.

5. The user's synchronization process terminates. The Hub resets the change flag for the connector user. If the connector does not support events, then the Hub leaves the change flags in the state that causes synchronization.

2.2 Overview of Connectors

The connectors use Web services to communicate user records to the Hub. They implement the BDSS Connector API as described in Appendix C, "Connector API," which details the BDSS Connector Interface, Connector Run-Time Interface, and Engine Callback Interface. A connector consists of the following three components:

- Hub Transport
- Transformer
- PIM Transport

> **Note:** Although you do not have to implement these three components, Oracle recommends their implementation as a best practice.

2.2.1 Hub Transport

The Hub Transport implements the connector interface and performs data exchange between the Engine and the connector.

2.2.2 Transformer

The Transformer translates data into both the Hub schema and the schema of the PIM servers. You can define how the Hub formats the data and freely revise it to meet customer needs. Using Oracle JDeveloper you can create the XSLTs (XML Transformations) that map the XSDs (XML Schema Definitions) of the Hub and PIM server. See also Chapter 8, "Mapping Connector Fields to Hub Fields."

> **Note:** You can create and revise the Hub schema, but you cannot change the connector schema. The connector schema is created by the developers of the connector and must remain fixed.

2.2.3 PIM Transport

The PIM Transport uses the PIM API interfaces and performs data exchange between the connector and the PIM server. As part of the data exchange, the PIM Transport translates data between the PIM XML schema representation and the PIM server representation.

2.2.4 How the Connector Components Manage Domain Data During Synchronization

Synchronization between the Hub and connectors is bidirectional by default, but can be configured to restrict the synchronization direction to either inbound-only or outbound-only.

2.2.4.1 Inbound Synchronization

The components interact as follows upon receiving data from the PIM server:

1. The PIM Transport collects a record set for a domain using a native PIM server API. This data is in the format specific to the PIM Server.

2. The PIM Transport converts each record to a PIM XML format that conforms to the PIM XSD defined for the domain to which the record belongs.

3. The PIM Transport passes each PIM XML record to the Transformer, which performs the following actions to translate the PIM XML record into the Hub XML record:

   a. Determines the synchronization direction (inbound) to apply the correct XLST document (PIM to Hub). See also Section 5.4, "Creating Connector Configuration Profiles."

   b. Invokes an API that applies the XSLT document to each record. This XLST contains the schema definitions for both the Hub and the PIM server.

   c. Returns the Hub XML record to the Hub Transport.

4. The Hub Transport calls an API to build a SOAP (Simple Object Access Protocol) message containing the Hub XML record set.

5. The Hub Transport then calls an API to push the SOAP message to the Hub.

2.2.4.2 Outbound Synchronization

Outbound synchronization begins with the Hub sending a SOAP message containing a single record in Hub XML format to the connector. The connector then transforms this XML record to an appropriate format required by the target PIM server.

The components interact as follows upon receiving data from the PIM server:

1. The Hub Transport receives the SOAP message containing the Hub XML record.

2. The Hub Transport then passes the Hub XML set to the Transformer.

3. The Transformer performs the following actions to translate the record from Hub XML to the appropriate PIM XML:

    a. Identifies the synchronization direction (outbound) to apply the appropriate XSLT to the Hub XML record (Hub to PIM).

    b. Calls an API to apply the XSLT.

    c. Passes the converted record to the Hub Transport.

4. The Hub Transport passes the PIM XML record to the PIM Transport.

5. The PIM Transport converts the PIM XML into a format needed by the native PIM API used to push the record to the PIM server.

6. The PIM Transport sends the PIM API-formatted record to the PIM server.

2.2.5 Error Messages

Errors can be integrated into a BPEL task flow to alert about administrators data errors that occur during synchronization. For example, if a user inadvertently creates a record with a character in the description that a third-party PIM Server cannot accept, then that PIM Server rejects the record during the next synchronization session. As a result, BDSS raises a BPEL event with the record summary and the error. The administrator sets up a BPEL process that checks the error code and then sends an e-mail telling the user to remove the unsupported character from the record.

An event is raised whenever BDSS logs a Critical, Error, or Warning message to its log files. BDSS can also raise a BPEL event to a BPEL server or web service.

3

Installing the BDSS Components

This chapter describes installing Oracle Business Data Synchronization Server (BDSS).

This chapter includes the following topics:

- Overview of Installation
- Deploying the Hub Components
- Overview of Installing and Configuring BDSS
- Creating a Domain
- Extending the Domain with the BDSS Templates
- Adding Security Credentials
- Setting SSL
- Enabling Data Transformation and Filtering for Exchange 2007 Connector
- Configuring the Exchange 2007 Connector
- Configuring the Hub
- Configuring BDSS Server Instances
- Configuring Connector Instances
- Adding Users
- Postinstallation Configuration

3.1 Overview of Installation

BDSS consists of the following independently installed components, each with their own installation prerequisites (listed in Table 3–1):

- The BDSS Hub components
- The BDSS connectors:
    - Oracle BDSS Connector for Microsoft Exchange 2007
    - The Oracle BPEL Task Connector

The companion CD includes `BDSS.zip`. This file contains the installer file, `bdss_template.jar`, the template file that creates the data source and queues. In addition, it contains the EAR files for the Oracle BDSS Connector For Exchange 2007 (the Exchange 2007 Connector), the Oracle BPEL Task Connector, the Engine (`BDSSHub.ear`), and the Dispatcher (`BDSSDispatcher.ear`). `BDSS.zip` also includes XSL transformation files and SQL scripts.

Table 3–1   Installation Checklist

Component	Required Software or Hardware	Preinstallation Tasks	Postinstallation Tasks
Dispatcher and Engine	Oracle WebLogic Server with the following components installed: ■ Oracle Enterprise Manager ■ At least one instance of Integrated WebLogic Server or a group of Integrated WebLogic Server instances onto which you can deploy the Hub components. You must install WebLogic Server using a user account with administrative privileges so that you have access to the preferences store on the system. ■ A database that supports Oracle EclipseLink for storage of configuration and mapping information. You must also have a database schema and user called *BDSS*. Oracle recommends Oracle Database 11*g*.	Configure a domain for the Oracle WebLogic Application Server.	Configure the Dispatcher (Section 3.14.1). Configure the Engine (Section 3.14.2).
Oracle BDSS Connector for Microsoft Exchange 2007 (The Exchange 2007 Connector)	See the requirements for the Dispatcher and Engine.	See the tasks for the Dispatcher and Engine. Installing Oracle Enterprise Manager provides you with the Oracle JRF (Java Required Files) 11.1.1.0 template.	Configure the Oracle BDSS Connector for Microsoft Exchange 2007 (Chapter 6).

3.1.1 Prerequisites

This section describes the minimum hardware and software requirements for deployment that would enable synchronization between different PIM servers. Figure 3–1 is an example of such a scenario.

> **Note:**   Hardware requirements for a deployment vary depending on the complexity and scale of the deployment.

The minimum set of hardware for the BDSS Hub and Dispatcher is as follows:

■ The Oracle Enterprise Manager

■ Oracle WebLogic Application Server 11*g* (the minimum for the Oracle BPEL Task Connector) for the Hub components and the Oracle BPEL Task Connector deployment.

■ A server for the data store with the Oracle Database 11*g*. For very small deployments, this database may reside on the same server as the other components.

This environment is sufficient for installing the Exchange 2007 Connector.

> **Note:** Although installing most of the components on separate application servers results in improved scalability support, you can install all of them on the same application server. However, installation on a single application server affects performance and port settings.

Figure 3–1   Basic Installation of BDSS
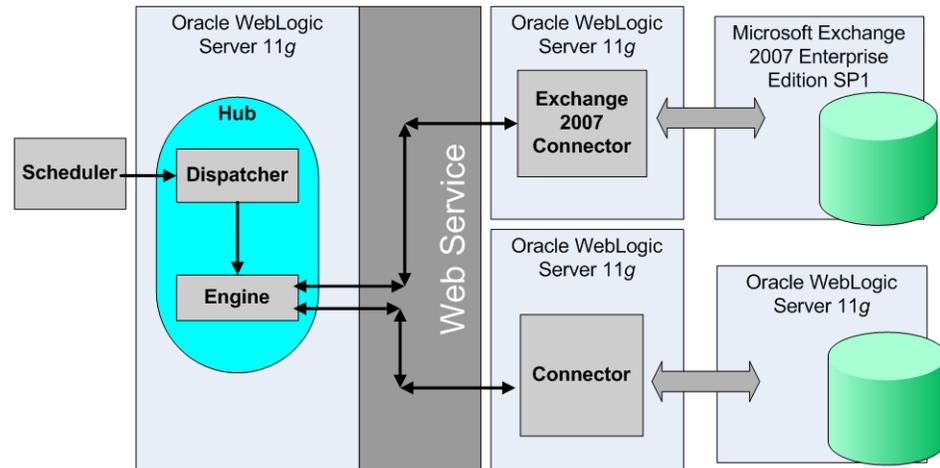
3.2 Deploying the Hub Components

The Dispatcher, Engine, and Exchange 2007 connector are packaged in `bdss_template.jar`. You can deploy the Dispatcher and the Engine on the same application server or on separate application servers. Because the Dispatcher communicates with the Engine through a remote MBean interface, it can be deployed to a separate application server if needed.

3.2.1 Prerequisites for Deploying the Hub Components

The required software for BDSS is as follows:

- Oracle WebLogic Application Server 11*g* (or later) with a domain to which you can deploy the components.

- A database that supports Oracle EclipseLink for storage of configuration and mapping information. Oracle recommends Oracle Database 11*g*.

3.3 Overview of Installing and Configuring BDSS

Tasks include:

1. Creating a Domain

2. Extending the Domain with the BDSS Templates

3. Adding Security Credentials

4. Setting SSL

5. Enabling Data Transformation and Filtering for Exchange 2007 Connector

6. Configuring the Exchange 2007 Connector

7. Configuring the Hub

8. Configuring BDSS Server Instances

9. Configuring Connector Instances

10. Adding Users

3.4 Creating a Domain

To create a domain using Fusion Middleware Configuration Wizard:

1. In the Welcome screen, select **Create a new WebLogic domain** and then click **Next**. The Select Domain Source screen appears.

2. Select **Oracle Enterprise Manager** and then click **Next**. You use Oracle Enterprise Manager to configure the security credentials used by the Exchange 2007 Connector (described in Section 3.6, "Adding Security Credentials"). You can also configure the Hub and Dispatcher using Oracle Enterprise Manager. For more information, see Chapter 5, "Configuring the BDSS Hub Services."

> **Note:** Select **Oracle JRF -11.1.1.0[jdeveloper]** if it is not selected by default. See also Section 3.4.1.1, "Installing the JRF Extension."

3. In the Configure Administrator User Name and Password screen, enter the user name and password and then click **Next**. The Configure Server Start Mode and JDK screen appears.

4. If needed, change the JDK; otherwise, navigate to the Create WebLogic Domain screen by clicking **Next**.

5. In the Create WebLogic Domain screen, enter a domain name and then click **Create**.

6. When the progress indicator completes, click **Done**.

3.4.1 Installing Java Required Files

You deploy the Exchange 2007 Connector on the Oracle WebLogic Server domain. You can include this WebLogic Server extension either as part of the initial installation of Oracle WebLogic Server (described in Section 3.4.1.1) or as an extension to a domain of an existing installation (described in Section 3.4.1.2).

3.4.1.1 Installing the JRF Extension

If the **Oracle JRF - 11.1.1.0 [jdeveloper]** option is available when installing the server, ensure that it is selected. Otherwise, extend a domain following the steps described in Section 3.4.1.2.

3.4.1.2 Extending a Domain for JRF

You can install JRF using the WebLogic Configuration Manager if there is an existing WebLogic Server installation and a domain that has not been extended for JRF.

To extend a domain to include JRF:

1. Start the Configuration Wizard using the Windows Start menu as follows:

   a. Select **Start**.

   b. Choose **Programs**.

    c. Choose *Your WebLogic Server Install*.

    d. Choose **WebLogic Server 10gR3**.

    e. Choose **Configuration Wizard**. The Configuration Wizard appears.

> **Note:** Italicized text indicates content that may differ, depending on environment.

2. In the Welcome screen, select **Extend an existing WebLogic domain** and the click **Next**.

3. Select the WebLogic Server domain and then click **Next**. The Select Extension Source screen appears.

4. Select **Extend my domain using an existing extension template**.

5. Using the browse function of the Template location field, navigate to, and then select `<domain root>\jdeveloper\common\templates\applications\jrf_template_ 11.1.1.jar`.

Figure 3–2 Selecting the JRF Template
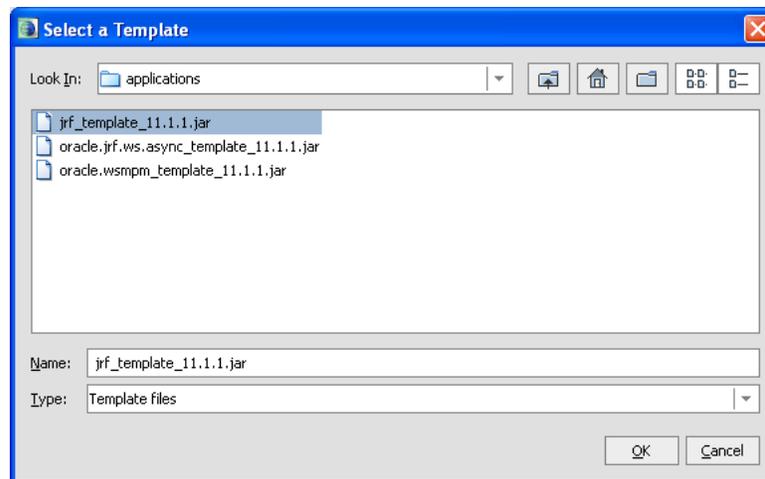
6. Click **OK**.

7. Complete the Configuration Wizard by accepting the default selections on the subsequent pages.

3.5 Extending the Domain with the BDSS Templates

The BDSS template (`bdss_template.jar`) creates a queue on Integrated WebLogic Server and the data source used by the Run-Time Configuration Manager. It can also deploy the BDSS Hub, Dispatcher and Exchange 2007 Connector applications.

You install these BDSS components by using the Fusion Middleware Configuration Wizard to extend a domain with BDSS templates.

> **Note:** These instructions assume that you have created a domain.

To install the components:

1. Start Configuration Wizard and then select **Extend an existing WebLogic Domain** in the Welcome screen (Figure 3–3). Click **Next**. The Select a WebLogic Domain Directory screen appears.

Figure 3–3   Extending the WebLogic Server Domain
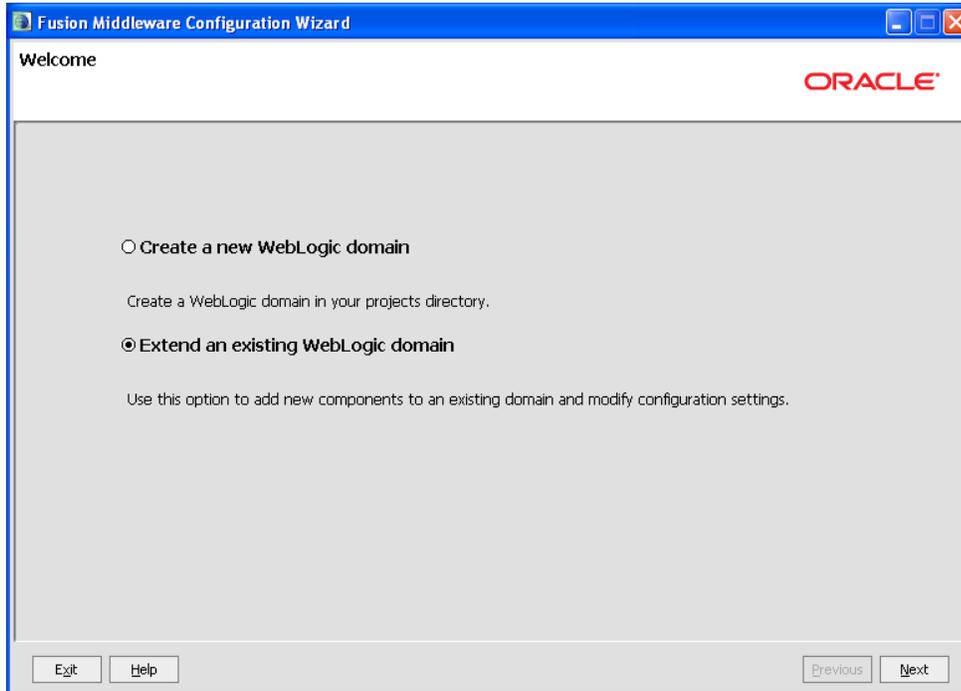
2. Select any domain that has JMS installed and then click **Next**. The Select Extension Source screen appears.

3. Select **Extend my domain using an existing extension template**.

4. Using the Select a Template dialog, locate bdss_template.jar (Figure 3–4) and then click **OK**.

Figure 3–4   Selecting bdss_template.jar
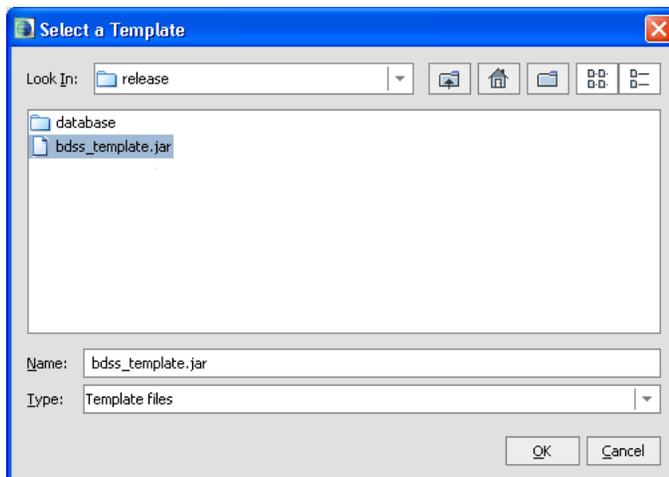
5. In the Select Extension Source dialog, select **BDSS template.jar** and then click **Next**.

Figure 3–5   Selecting bdss_template.jar
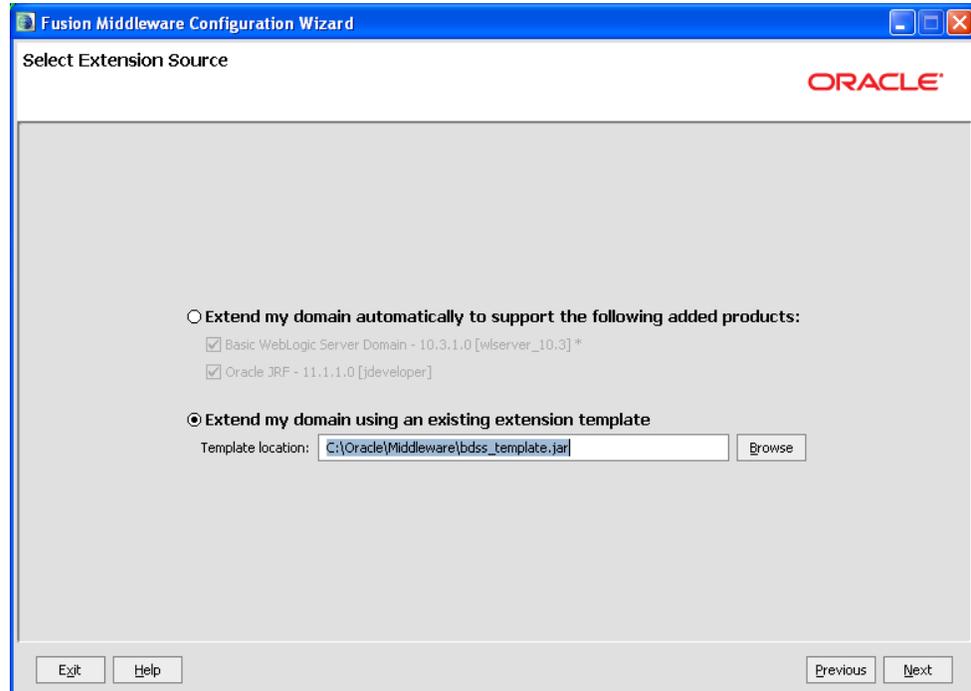
6. In the Configure JDBC DataSource page (Figure 3–6), enter the connection information. The user name must be *bdss*. Click **Next**. The Test JDBC Data sources screen appears.

Figure 3–6   Configuring the JDBC Data Source
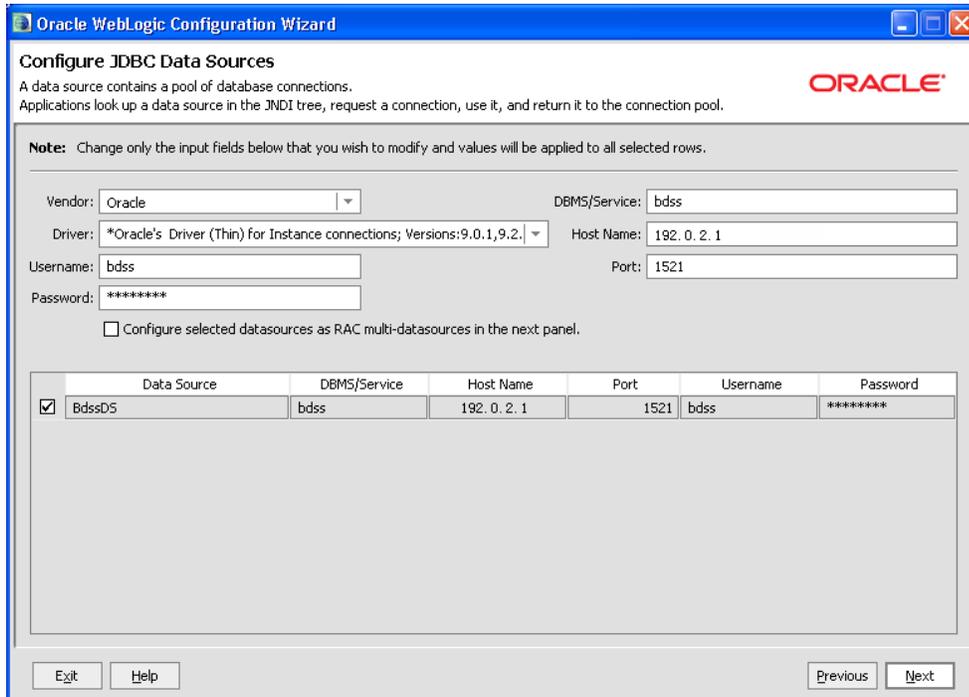
7. Click **Test Connections**. Click **Next** after Configuration Wizard generates a *Test Successful!* message (Figure 3–7). The Run Database Scripts screen appears.

Figure 3–7   Testing the JDBC Connections
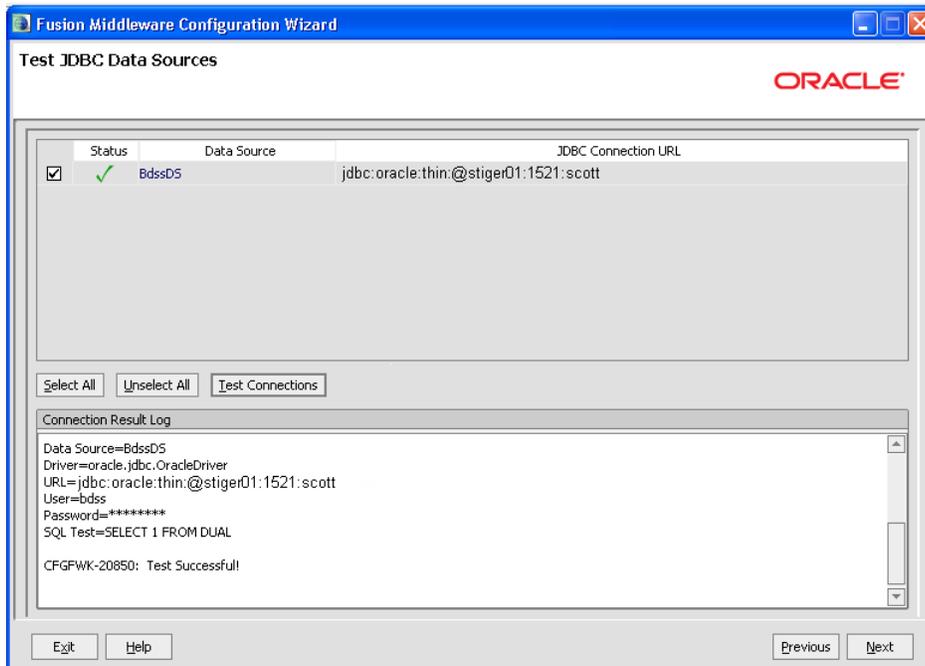
8. In the Run Database Scripts page (Figure 3–8), click **Run Scripts**.The `CreateBDSSDataModel.sql` script creates the database tables and triggers and the `BDSSSeedData.sql` script populates the database with sample data for the domains.

Figure 3–8   Creating the Database Tables and Seeding Data
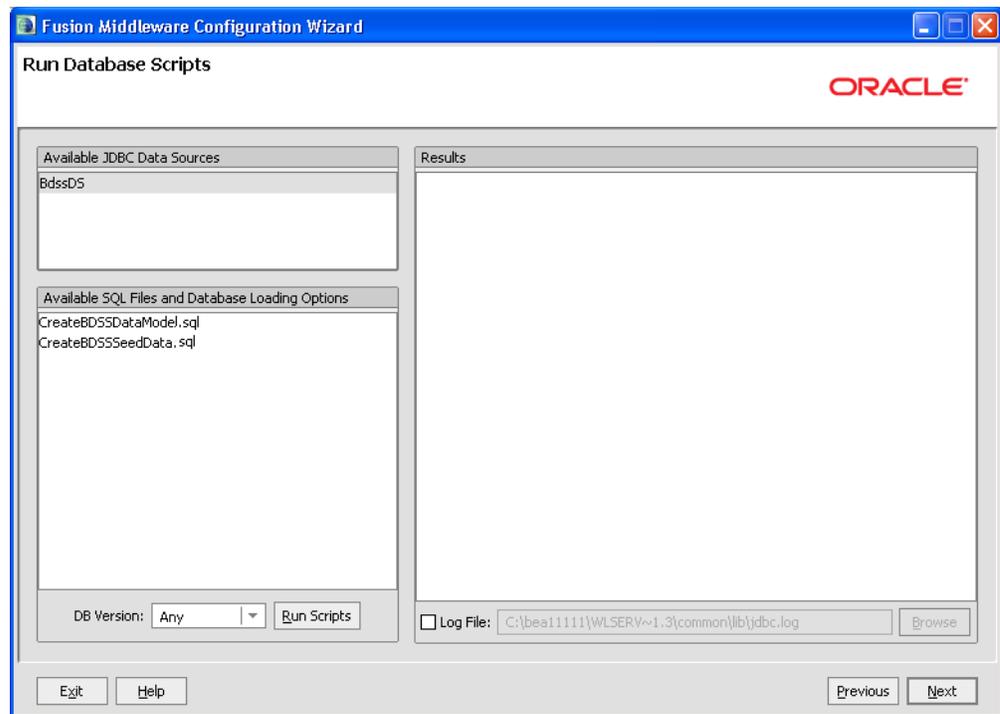
9. Continue to the Target Deployments to Clusters and Servers screen (Figure 3–9).

> **Note:**   The Dispatcher, the Hub, and the Exchange 2007 Connector
> are deployed by default. If needed, select **Deployments and Services**
> in the Select Optional Configuration screen that follows the Run
> Database Scripts screen. After you select this option, you then select
> the applications to deploy. This optional configuration enables
> scalability by allowing you to deploy each component on a different
> WebLogic Server.

Figure 3–9   Selecting BDSS Components
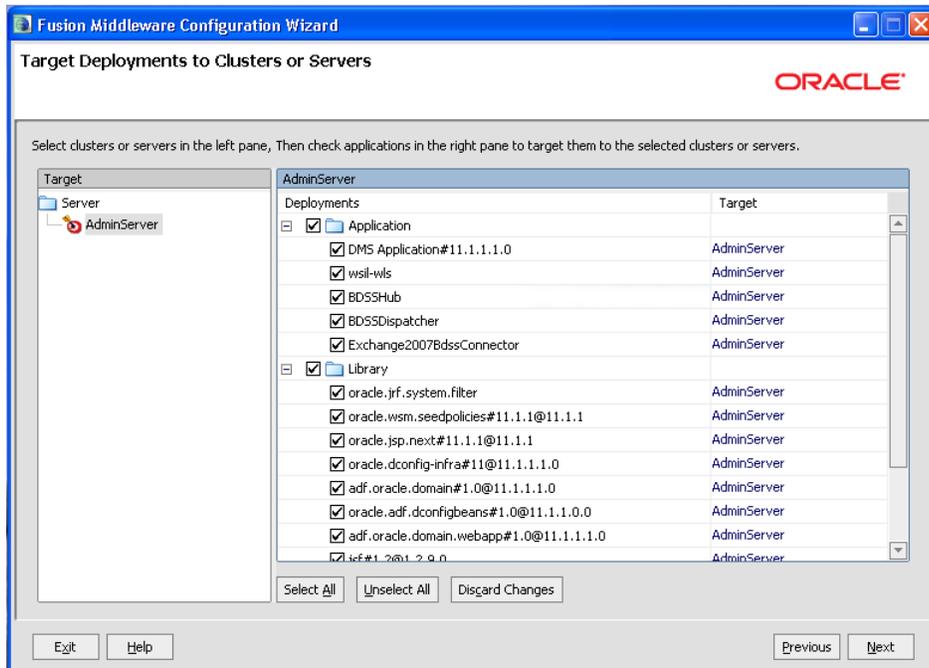
10. Select the BDSS components to deploy and then click **Next**. By default, the wizard deploys the Dispatcher, the Hub, and the Exchange 2007 Connector. The Configuration Summary screen appears (Figure 3–10).

Figure 3–10   The Configuration Summary Screen
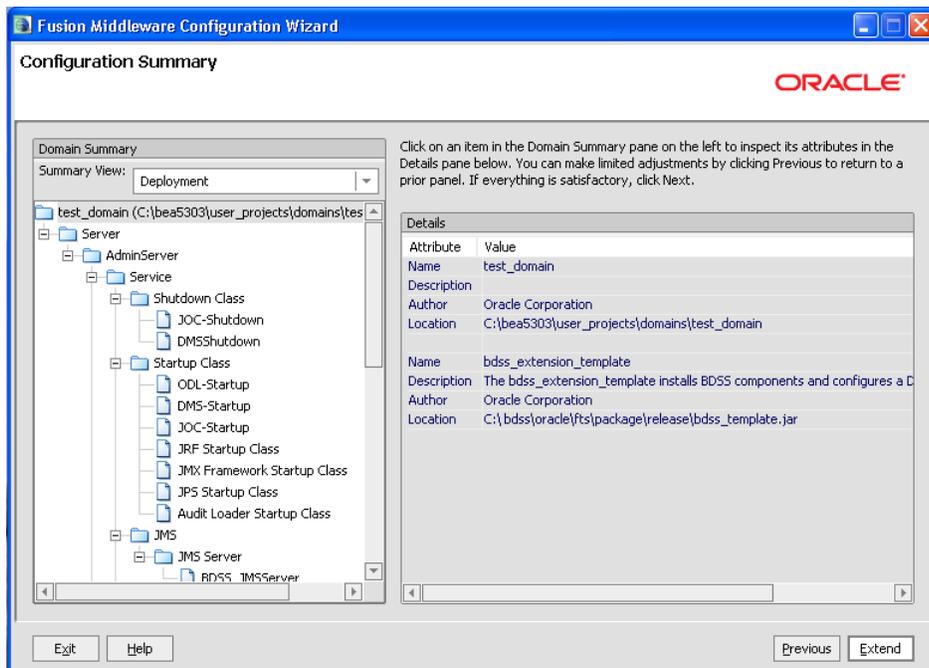
11. Review the installed components. Click **Previous** to return to the Target Deployments to Clusters or Services and change your selection.

12. Click **Next**. The Configuration Manager creates the domain. When the progress indicator on the Creating Domain screen completes (Figure 3–11). Click **Done**.

Figure 3–11   The Create Domain Screen
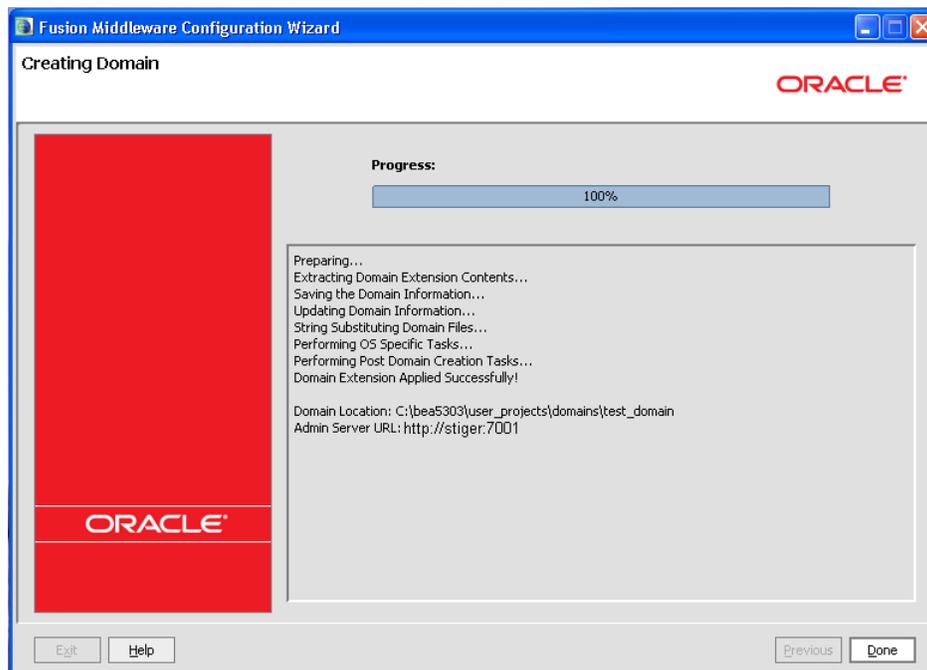
3.6 Adding Security Credentials

The Exchange 2007 Connector requires the three sets of credentials listed in Table 3–2.

Table 3–2   Credentials Required by Exchange 2007 Connector

JPS Map Name	JPS Key	User Name and Password
exchange2007pimtransport	activeDirectoryAdminCredential	Contains the user name and password used by the Exchange 2007 Connector when it queries the Active Directory
exchange2007pimtransport	autoDiscoveryAdminCredential	Contains the user name and password used by the Exchange 2007 Connector when it queries the Microsoft Autodiscover service
exchange2007pimtransport	exchangeWebServiceAdminCredential	Contains the user name and password used by the Exchange 2007 Connector when it calls the Exchange Web Service to access user mailboxes

If the Windows domain is required, it can be prepended to the user name and a backslash ("\") is used to separate the domain and user name as in *domain\username*.

Add the credentials listed in Table 3–2 using Fusion Middleware Control, as illustrated in Figure 3–12.

Figure 3–12   Creating Credentials in Oracle Enterprise Manager

The following steps describe how you enter in the credentials using Oracle Enterprise Manager.

1. Log in to Oracle Enterprise Manager.

2. Expand the **WebLogic Domain** node in the navigation pane and then right-click your domain.

3. From the context menu, select **Security** and then **Credentials**. The Security Page appears.

4. Create the exchange2007pimtransport credential map by clicking **Create Map**. The Create Map dialog appears.

5. Enter exchange2007pimtransport in the Map Name field and then click **OK**.

6. For each JPS key listed in Table 3–2, perform the following:

   a. Click **Create Key**. The Create Key dialog appears.

   b. Select exchange2007pimtransport from the Select Map list.

   c. Enter the JPS key. For example, enter *activeDirectoryAdminCredential*.

   d. In the User Name field, enter in the user name (ensuring to prefix the domain if needed)

   e. In the Password field, enter the password.

7. Click **OK**.

> **Note:**   You can access the System MBean Browser from the context menu that appears when you right-click a domain in the Navigation pane. Using the System MBean browser, you can invoke some operations on the JPS MBean, but not all of them. The steps described in Section 3.6 are the correct method for entering credentials. You can also use the System MBean browser to manage the BDSS MBeans, which you use to configure BDSS Hub components and server and connector instances.

3.7 Setting SSL

If the configuration of the Exchange 2007 servers require secure communications, then you must configure the Exchange 2007 Connector to support SSL to enable it to communicate securely with the Microsoft Autodiscovery and Exchange Web Service (EWS). The Certificate Authority (CA) that issued and signed the certificate bound to the EWS server must be installed as a trusted CA in the trust store used by the connector. By default, the application deployed to WebLogic Server use a trust store called DemoTrust.jks, which is located in the %WL_HOME%\lib directory.

To configure SSL, you must first obtain the X.509 certificate bound to the IIS Web Server hosting the Exchange Web Services and then update `DemoTrust.jks` with this certificate.

> **Note:** The following instructions assume a deployment that uses the default trust store, `DemoTrust.jks`. If you have created a different trust store and have configured WebLogic Server to use it instead of the default trust store, this instructions will still apply, but you must define the name and password using the Java keytool utility.

To install the certificate:

1. Obtain the X.509 certificate bound to the IIS Web Server hosting the Exchange Web Services and save it as a file on the file system on the same box the connector is being deployed on. For the purposes of this document, the file is saved at `c:\temp\exchangecacert.cer`.

2. Log in to the Exchange 2007 server.

3. Start IIS Manager.

4. Right click the Exchange virtual directory (typically at **Web Sites** > **Default Web Site** > **Exchange**) and select **Properties**.

5. Navigate to the Directory Security.

6. Click **View Certificate**.

7. Click the Details tab and then select **Copy to File**.

8. In the wizard, select the defaults on each page until you are prompted for the file name. Specify the name and location for the file. For example, enter *c:\temp\exchangecacert.cert*.

To update `DemoTrust.jks` with the certificate:

1. Use the Command Prompt to change the current directory to the location of `DemoTrust.jks`. For example, enter the following command:

   ```
   cd c:\WL_HOME\wlserver_10.3\server\lib
   ```

2. Use the Java keytool utility to import the certificate.

   For example, navigate to location of the keytool utility, such as `C:\Program Files\Java\jdk1.6.0_25\bin`, and then enter the following commands:

   ```
   keytool -importcert
           -trustcacerts -file C:\temp\exchangecacert.cert
           -keystore DemoTrust.jks
           -storepass DemoTrustKeyStorePassPhrase -alias XXX
   ```

> **Note:** If you use a trust store other than `DemoTrust.jks`, specify the appropriate name and password in the `-keystore` and `-storepass` options, respectively.

Refer to Java SE Technical Documentation (http://download.oracle.com/javase/index.html) for information on certificate files and how to use the keytool utility.

3.8 Enabling Data Transformation and Filtering for Exchange 2007 Connector

The Exchange 2007 Connector uses XSD and XSLT files to transform records to and from the Hub XML and PIM XML forms and XML files to filter unwanted records. These files must be located on a database file system and the location must be configured in BDSS. The seed data script provides you with a default location of `c:/fusion_sync <root>`. For example, the seed data script populates the database with such strings as `c:/fusion_sync/Exchange2007/....`

To set up the various XSLT and XSD files on the file system and configure the XSL and XSD files in BDSS, you can copy the `transform` and `filter` directories from where you extracted `BDSS.zip` to your preferred location. For example, the directory structure may be in the following form:

`<MY DIRECTORY>/exchange2007/task/transform/exchange2007task.xsd`

To set up the files:

1. Copy the Exchange 2007 XSD and XSL files to a local directory having the following hierarchy:

   - `<root>/exchange`—This directory should contain a contact and task subdirectory and the `exchange2007typelibrary.xsd` file.

   - `<root>/exchange/contact/transform`—This directory should contain the following three files:

     ```
     exchange2007contact.xsd
     exchange2007contacttohubcontact.xsl
     hubcontacttoexchange2007contact.xsl
     ```

   - `<root>/exchange/task/transform`—This directory should contain the following three files:

     ```
     exchange2007task.xsd
     exchange2007tasktohubtask.xsl
     hubtasktoexchange2007task.xsl
     ```

   - `<root>/exchange/calendar/transform`—This directory should contain the following three files.

     ```
     pimicalbody.xsd
     exchange2007calendartohubcalendar.xsl
     hubcalendartoexchange2007calendar.xsl
     ```

   - `<root>/exchange/calendar/transform/intermediatetranslation`—the directory should contain the following six files

     ```
     calendaritemtype.xsd
     calendaritemtypetoveventasxml.xsl
     timezoneregistry.xml
     timezoneregistryschema.xsd
     veventasxml.xsd
     veventasxmltocalendaritemtype.xsl
     ```

2. Copy the Exchange 2007 XML filter files to a local directory having the following hierarchy:

   - `<root>/exchange`—This directory should contain a contact a task subdirectory and the `exchange2007typelibrary.xsd` file.

- `<root>/exchange/contact/filter`—This directory should contain `contactfilter.xml` file.

- `<root>/exchange/task/filter`—This directory should contain the `taskfilter.xml` file.

- `<root>/exchange/calendar/filter`—This directory should contain the `calendarfilter.xml` file.

3. Copy the Hub XSD and XSL files to a local directory having the following hierarchy:

   a. `<root>/hub`—This directory should contain a contact and task subdirectory and the `HubTypeLibrary.xsd` file.

   b. `<root>/hub/contact/transform`—This directory should contain the following eight files:

   ```
   hubaddress.xsd
   hubcontact.xsd
   hubcountry.xsd
   hubemailaddress.xsd
   hubnameprefix.xsd
   hubnamesuffix.xsd
   hubphone.xsd
   hubstate.xsd
   ```

   c. `<root>/hub/task/transform`—This directory should contain the `HubTask.xsd` file.

   d. `<root>/hub/calendar/transform`— This directory should contain the `HubCalendar.xsd` file.

3.9 Configuring the Exchange 2007 Connector

This section describes the following tasks.

- Configuring the Exchange 2007 Connector for Record Transformation

- Configuring the Exchange 2007 Connector to the Active Directory Service

- Configuring Exchange 2007 Connector to Use the Auto Discovery Service

These tasks involve only the parameters that you must configure. Table 3–3 lists all of the configuration options for the Exchange 2007 Connector, both mandatory and optional. Although the environment dictates which of these parameters you define, you must define the following in all circumstances:

- Active Directory Server

- Auto Discovery Service Connection Point (SCP)

- Event Receiver

This configuration resides in the PROFILES table. You configure the Exchange 2007 Connector using the BDSS MBeans that are deployed to a JMX console. You can also configure the Exchange 2007 Connector using such tools as JDeveloper or SQL Developer. See also Chapter 6, "Configuring the BDSS Connector for Microsoft Exchange 2007."

Note: You must restart the BDSS Hub if you use a console other than Oracle Enterprise Manager or JConsole.

Table 3–3   Exchange 2007 Configuration Parameters

Profile	Section	Parameter	Default Value(s)	Description
Exchange2007	Hub Transport	Pim Transport Implementation Class	`oracle.bdss.connectors.exchange2007.pimtransport.PimTransportAdapter`	The Hub Transport generic component uses this value to initialize the Exchange PIM Transport
Exchange2007	Hub Transport	Extract Response Batch Size	100000	Set this value to a sufficiently large number so that no user synchronized byBDSS has records that exceeds this value
Exchange2007	Task	Pim Xsd	A path	The URI to the `exchange2007task.xsd` file
Exchange2007	Task	PimToHubUrl	A path	The URI to the `Exchange2007TaskToHubTask.xsl`
Exchange2007	Task	HubToPimUrl	A path	The URI to the `HubTaskToExchange2007Task.xsl`
Exchange2007	Task	Domain Class Type	Task Type	The Exchange Web Service interface class that is used when synchronizing the domain
Exchange2007	Task	Exchange Domain Batch Size	512	The maximum number of records downloaded from Exchange when extract records from the domain target. Valid values are 1 through 512.
Exchange2007	Task	Folder Class	IPF.Task	The folder class for the domain
Exchange2007	Task	Body Type	Text	How the Body field for the domain is synchronized. Valid values are `Text` or `HTML`
Exchange2007	Task	Conflict Resolution Type	AlwaysOverwrite	Determines how Exchange handles a conflict type. Valid values are `AutoResolve`, `AlwaysOverwrite`, and `NeverOverwrite`
Exchange2007	Task	Disposal Type	MoveToDeletedItems	Determines how Exchange handles deleted records. Valid values are `HardDelete`, `SoftDelete`, and `MoveToDeletedItems`
Exchange2007	Task	Pim Domain id	IPM.Task	The PIM domain identifier, used as the message class property for each synchronizing record

Table 3–3   (Cont.) Exchange 2007 Configuration Parameters

Profile	Section	Parameter	Default Value(s)	Description
Exchange2007	Contact	Exchange Domain Batch Size	512	The maximum number of records downloaded from Exchange when extracting records from the domain target. Valid values are 1 to 512, inclusive.
Exchange2007	Contact	Folder Class	IPF.Contact	The folder class for the domain
Exchange2007	Contact	Pim Domain Id	IPM.Contact	The PIM domain identifier, used as the message class property of each record synchronized
Exchange2007	Contact	Pim Xsd	A path	The URI to the `exchange2007contact.xsd` file
Exchange2007	Contact	PimToHubUrl	A path	The URI to the `Exchange2007ContactToHubContact.xsl`
Exchange2007	Contact	HubToPimUrl	A path	The URI to the `HubContactToExchange2007Contact.xsl`
Exchange2007	Calendar	Time Zone Xsd	A path	The URI to `timezoneregistryschema.xsd`
Exchange2007	Calendar	VEvent Xsd	A path	The URI to `veventasxml.xsd`
Exchange2007	Calendar	PimToHubUrl	A path	The URI to `exchange2007calendartohubcalendar.xsl`
Exchange2007	Calendar	Pim Xsd	A path	The URI to `pimicalbody.xsd`
Exchange2007	Calendar	Calendar Item Type Xsd	A path	The URI to `calendaritemtype.xsd`
Exchange2007	Calendar	Time Zone Xml	A path	The URI to `timezoneregistry.xml`
Exchange2007	Calendar	HubToPimUrl	A path	The URI to `hubcalendartoexchange2007calendar.xsl`
Exchange2007	Calendar	CalendarItemToVEvent Xsl	A path	The URI to `calendaritemtypetoveventasxml.xsl`
Exchange2007	Calendar	VEventToCalendarItem Xsl	A path	The URI to `veventasxmltocalendaritemtype.xsl`
Exchange2007	Calendar	Body Type	`Text`	How the Body field for the domain is synchronized. Valid values are `Text` or `HTML`.

Table 3–3   (Cont.) Exchange 2007 Configuration Parameters

Profile	Section	Parameter	Default Value(s)	Description
Exchange2007	Calendar	Conflict Resolution Type	`AlwaysOverwrite`	Determines how Exchange handles a conflict type. Valid values are `AutoResolve`, `AlwaysOverwrite`, and `NeverOverwrite`.
Exchange2007	Calendar	Disposal Type	`MovetoDetailItems`	Determines how Exchange handles deleted records. Valid values are `HardDelete`, `SoftDelete`, and `MoveToDeletedItems`.
Exchange2007	Calendar	Domain Class Type	`CalendarItemType`	The Exchange Web Service interface class that is used when synchronizing a domain
Exchange2007	Calendar	Exchange Domain Batch Size	512	The maximum number of records downloaded from Exchange when extracting records from the domain target. Valid values are 1 to 512, inclusive.
Exchange2007	Calendar	Fanning Limit	30	The number of records created when a recurring calendar record cannot be natively represented in Exchange. Valid values are 1 to 100, inclusive.
Exchange2007	Calendar	Folder Class	IPF.Appointment	The folder class for the domain
Exchange2007	Calendar	Pim Domain id	IPM.Appointment	The PIM domain identifier, used as the message class property of each record synchronized.
Exchange2007	Active Directory	ServerN (where N is a ordinal >= 0)	An LDAP URL	The LDAP URL used by the connector to obtain Service Connection Point URLs from Active Directory when discovering a user's Exchange account information. You can add multiple entries. The ordinal specifies a preference order (that is, Server0 is tried before Server1, and so on). The connector tries each URL until the user is discovered or each entry has been tried. This is a required parameter. See also Chapter 6.7, "Configuring the Microsoft Exchange 2007 Connector to Use Active Directory."

Table 3–3   (Cont.) Exchange 2007 Configuration Parameters

Profile	Section	Parameter	Default Value(s)	Description
Exchange2007	Active Directory	Referral	follow	Specifies how the LDAP provider handles referrals. Values are *follow*, *throw* or *defer*. The default value is *follow*.
Exchange2007	Active Directory	Max Query Result Size	20	Max number of results to return per Active Directory query. The default value is 20.
Exchange2007	Active Directory	Max Query Time	30000	The time (in milliseconds) to wait for an Active Directory query to return. The default value is 30 seconds (30000 milliseconds).
Exchange2007	Active Directory	Read Timeout	5000	The time (in milliseconds) to wait for a response before timing out. The default value is 5000.
Exchange2007	Active Directory	Connect Timeout	5000	The time (in milliseconds) to establish a connection to Active Directory before timing out. The default value is 5000.
Exchange2007	Active Directory	Use SSL	False	Whether secure LDAP is used. The default is `true`. However, in the development environment, set the value to `false` because the Domain Controller is not enabled for secure LDAP.
Exchange2007	Active Directory	Using Client Certificates	False	Indicates if the client provider uses LDAP certificates
Exchange2007	Active Directory	Client Certificate Path	False	Set to `true` if Use SSL is set to true and the Domain Controller requires a client certificate. The default value is `false`.

Table 3–3   (Cont.) Exchange 2007 Configuration Parameters

Profile	Section	Parameter	Default Value(s)	Description
Exchange2007	Auto Discovery	SCP*N* where *N* is an ordinal >= 0	<HTTPS URL>	The URL to a Service Connection Point (SCP) that the connector should use when discovering a user. The ordinal specifies a preference order (the connector tries *SCP0* is tried before *SCP1*, and so on). The connector tries each URL until the user is discovered or each entry has been tried. This is a required parameter. See also Chapter 6.9, "Configuring the Exchange 2007 Connector to Use the Microsoft Autodiscover Service."
Exchange2007	Auto Discovery	Remove Domain From Urls	True or False	Directs the connector to remove the domain information from a discovered SCP URLs obtained from Active Directory. For the development environment, set the value to *true* because DNS prevents the URL from being resolved when the URL contains a domain (that is, `http://server1.example.com/autodiscover/autodiscover.xml` does not work.) When the option set to true, the URL becomes `http://server1/autodiscover/autodiscover.xml`.
Exchange2007	Auto Discovery	Read Timeout	5000	The time (in milliseconds) to wait for a response before timing out. The default value is 5000.
Exchange2007	Auto Discovery	Connect Timeout	5000	The time (in milliseconds) to establish a connection to Auto Discover server before timing out. The default is 5000.
Exchange2007	Auto Discovery	Use SSL	True	Whether secure HTTP (HTTPS) is used. The default is *true*. HTTPS may be required for the Auto Discover service if attempts to configure the server to allow HTTP fail.

Table 3–3   (Cont.) Exchange 2007 Configuration Parameters

Profile	Section	Parameter	Default Value(s)	Description
Exchange2007	Auto Discovery	Client Certificate Path	Defaults to empty string.	Set this value to the location of the PEM (Privacy Enhance Mail security certificate) file containing the client certificate that is used if Auto Discover requires client certificates
Exchange2007	Auto Discovery	Client Certificate Key Path	Defaults to empty string.	Set this value to the location of the PEM (Privacy Enhance Mail security certificate) file containing the client certificate private key
Exchange2007	Auto Discovery	Port	443	Defaults to 443, the typical HTTPS secure port. Set the value to the HTTPS port if Use SSL is set to `true`. If false, this value should be the HTTP port. The value is only used when attempting default SCP URLs (which are attempted when all SCP URLS have been tried (those in the SCPN configuration and those obtained from AD per the Server*n* configuration).
Exchange2007	Event Receiver	CHANGE_ME	The value should be the URL of the deployed event receiver for the connector.	The name of an Exchange 2007 connector as it appears in the NAME Column of the CONNECTORS table. This is a required parameter.
Exchange2007	Event Receiver	Status Frequency	1	The frequency, in minutes, that Exchange provides a notification for each subscription. This is a required parameter.
Exchange2007	Event Receiver	Subscription Status Factor	3	The event receiver performs a check on all subscriptions to verify that the Exchange server has not terminated any subscriptions. This parameter sets frequency in which this check occurs. The frequency in which this check occurs is the value set for this parameter multiplied by the value set for the Status Frequency parameter. For example, if you set the Status Frequency parameter to 1 minute, and then set the Subscription Status Factory to 3, then the event receiver performs the subscription check every 3 minutes. This is a required parameter.

Configuring Parameters to Use URIs for Transformation Files

When configuring such parameters as Pim Xsd, PimToHubUrl, HubToPimUrl, and Calendar Item Type xsd, to use URIs for the transformation files, do not enter the URI in the format of `file:///.`/. Instead, enter the URI in the following format:

```
C:/directory1/directory2.../file.NNN
```

For example, if `c:/fusion_sync` is the `<root>` directory for transformer files, enter the following URI:

```
c:/fusion_sync/transform/exchange/task/exchange2007task.xsd
```

3.9.1 Configuring the Exchange 2007 Connector for Record Transformation

Using the Oracle Enterprise System MBean Browser, you update parameters of the Hub Transport section to use the same name as the Exchange 2007 Connector. For more information on these parameters, see Section 6–1, " Parameters of the Hub Transport Section." You must also update the Pim XSD, PimToHubUrl, and HubToPimUrl parameters of the Task section. See also Section 6.3, "Configuring Record Transformation for the Exchange 2007 Connector."

3.9.2 Configuring the Exchange 2007 Connector to the Active Directory Service

You must configure the ServerN and UseSSL parameters of the Exchange Connector's Active Directory profile. The other parameters for this profile are optional and use appropriate defaults if you do not configure them. For more information, see Section 6.7, "Configuring the Microsoft Exchange 2007 Connector to Use Active Directory."

3.9.3 Configuring Exchange 2007 Connector to Use the Auto Discovery Service

The Exchange 2007 Connector queries the Microsoft Autodiscover service to discover information about users' Exchange accounts. The SCPN parameter of the Auto Discovery profile is a mandatory parameter, as it provides the connector with the list of Service Connection Point URLs that the connector queries to locate the Auto Discovery service. For more information, see Section 6.9, "Configuring the Exchange 2007 Connector to Use the Microsoft Autodiscover Service."

3.10 Configuring the Hub

The attributes of the Engine are exposed through the EngineSettings MBean and the attributes for the Dispatcher are exposed through the DispatcherSettings MBean that are described in Chapter 5, "Configuring the BDSS Hub Services."

To configure the Dispatcher, you must set the URL used by the Dispatcher to connect to the DispatcherHub Web service on the server that hosts the Hub. You configure this URL through the HubEndPointURL parameter of the DispatcherSettings MBean. For more information, see Section 5.2, "Configuring the Dispatcher."

For the Engine, you must configure the EngineSettings MBean's EngineEndpointURL attribute, which provides the URL used by the Engine to enable the connectors to extract requests and data update request responses to the Engine. In addition, you must set the URL for the RunTimeLibraryURL attribute. The Engine sends this URL to the connectors to enable them to communicate with the Connector Run-Time Library. For information on how to set these attributes, see Section 5.3, "Configuring the Engine."

3.11 Configuring BDSS Server Instances

Use the ServerDomainConfigMBean's `updatePIMServerInstance` operation to add the name of the Connector to a BDSS instance as described in Section 4.3.1, "Adding and Updating BDSS Instances."

3.12 Configuring Connector Instances

You must configure the instances of the connectors and the PIM servers with the URLs of the connector web service. To add this URL to a **connector instance**, use the Connector Configuration MBean's `updateConnectorInstance` operation as described in Section 4.4.1, "Adding and Updating Connector Instances."

3.13 Adding Users

Use the `addConnectorUser` operation of the User Configuration MBean to create users. Create one user for each configured connector per **Hub user**. For more information, see Section 4.7.2, "Creating Connector Users."

3.14 Postinstallation Configuration

To enable the proper functioning of the components, perform the following postinstallation tasks:

- Configuring the Dispatcher
- Configuring the Engine
- Configuring PIM Server Instances Using MBeans

3.14.1 Configuring the Dispatcher

The attributes for the Dispatcher are exposed through the DispatcherSettings MBean, which you can configure through the Oracle Enterprise Manager console or through JConsole. For more information, see Section 5.2, "Configuring the Dispatcher."

3.14.2 Configuring the Engine

The attributes of the Engine are exposed through the EngineSettings MBean. For more information, see Section 5.3, "Configuring the Engine." After you configure the EngineSettings MBean, you next configure the parameters of the `setProfileParameter` operation (listed in Table 3–4). For more information, see Section 4.8.2, "Managing Profile Parameters."

Table 3–4   Parameters of the setProfileParameter Operation

Profile	Section	Parameter	Description
Exchange2007	Task	URL for transformations	The URL where the XLST transformation files are stored on the server
Exchange2007	Calendar	URL for transformations	The URL where the XLST transformation files are stored on the server

Table 3–4   (Cont.) Parameters of the setProfileParameter Operation

Profile	Section	Parameter	Description
Exchange2007	Contact	URL for transformations	The URL where the XLST transformation files are stored on the server
Exchange2007	Task	Filter Condition	The URL where the XML filter files are stored on the server
Exchange2007	Calendar	Filter Condition	The URL where the XML filter files are stored on the server
Exchange2007	Contact	Filter Condition	The URL where the XML filter files are stored on the server

> **Tip:** Review other parameter values to determine if they should be changed from their default values.

3.14.3 Configuring PIM Server Instances Using MBeans

In addition to the installation scripts, the operations of the ServerDomainConfigMBean of the fusionsync-fusionsync-WS application enable you to create, delete, and update connectors. For more information, see Chapter 4, "Getting Started with Administering BDSS".

Table 3–5   Operations for Configuring BDSS Instances

Operation	Function
showConnectorDetails	Viewing the Connector Configuration
updateConnectorForPIM	Adding and Updating Connectors
deleteConnector	Deleting a Connector
updatePimServerInstance	Adding and Updating BDSS Instances
deletePimServerInstance	Deleting a PIM Server Instance
updateConnectorInstance	Adding and Updating BDSS Instances
deleteConnectorInstance	Deleting a PIM Server Instance
updateDomainForConnector	Creating a Domain for a Connector
deleteDomainForConnector	Deleting a Domain from a Connector
updateFilterForDomain	Creating a Filter for a Domain
deleteFilterForDomain	Deleting a Filter from a Domain

3.14.3.1 Creating PIM Server Instances and Connector Instances

The `CreateBDSSConnectorInstances.sql` script provides an alternative method to MBeans for creating server and connector instances. To seed the proper tables with PIM server instances and connector instances for the server, run the `CreateBDSSConnectorInstances.sql` script. You can run this script, which is located in the database folder, from any Oracle database client using the SQL*Plus (`SQLPlus.exe`) utility.

> **Note:** This script is optional, because you can also add and modify servers using the MBean interface.

The `CreateBDSSConnectorInstances.sql` script requires the following parameters:

- The Microsoft Exchange 2007 server
- The name for the The Exchange 2007 Connector
- The URL for the Exchange 2007 Connector

Part II

Basic Administration

This part presents information about performing basic administration tasks for Oracle Business Data Synchronization Server (BDSS) and contains the following chapters:

- Chapter 4, "Getting Started with Administering BDSS"
- Chapter 5, "Configuring the BDSS Hub Services"
- Chapter 6, "Configuring the BDSS Connector for Microsoft Exchange 2007"
- Chapter 7, "Configuring Security for BDSS"
- Chapter 8, "Mapping Connector Fields to Hub Fields"
- Chapter 9, "Synchronizing the Calendar Domain"

4

Getting Started with Administering BDSS

This chapter describes the administrative tasks for Business Data Synchronization Server (BDSS).

This chapter includes the following topics:

- Overview of Administration Tasks
- Managing Connectors
- Managing BDSS Server Instances
- Managing Connector Instances
- Managing Domains
- Managing Domain Filters
- Managing Users
- Managing Profiles

4.1 Overview of Administration Tasks

Typically, the administration tasks include:

- Managing the instances of PIM servers
- Managing connectors:
  - Creating and deleting connectors
  - Setting the synchronization direction (inbound, outbound, or both) for a connector
  - Creating connector instances
  - Pointing the connector instance to the connector's WSDL
- Managing Domains:
  - Creating and deleting the domains for a connector
  - Setting the synchronization direction (inbound, outbound, or both) for a domain
- Managing Users:
  - Creating and deleting a user and a **Hub user**
  - Changing the synchronization direction (inbound, outbound, or both) for the domain of a specified connector user

- Using filters to remove outdated task records

- Creating, deleting, or updating the parameters for a connector's profile

These tasks are enabled through the attributes and operations of the ServerDomainConfigMBean, the UserConfigMBean, and the ProfileRuntimeInfo MBean. An MBean (managed bean) is a Java object that represents a JMX-manageable resource in a distributed environment, such as an application, a service, a component, or a device.

The attributes and operations of the ServerDomainConfigMBean (listed in Table 4–1) enable you to perform tasks related to managing connectors, PIM servers, and domains.

Table 4–1   Operations of the ServerDomainConfigMBean

Task	Operation
Viewing the Connector Configuration	showConnectorDetails
Adding and Updating Connectors	updateConnectorForPIM
Deleting a Connector	deleteConnector
Adding and Updating BDSS Instances	updatePimServerInstance
Deleting a PIM Server Instance	deletePimServerInstance
Adding and Updating Connector Instances	updateConnectorInstance
Deleting a Connector Instance	deleteConnectorInstance
Creating a Domain for a Connector	updateDomainForConnector
Deleting a Domain from a Connector	deleteDomainForConnector
Creating a Filter for a Domain	updateFilterForDomain
Deleting a Filter from a Domain	deleteFilterForDomain

The operations of the UserConfigMBean (listed in Table 4–2) enable you to perform such tasks as creating users and configuring user domains.

Table 4–2   Operations of the UserConfigMBean

Task	Operation
Viewing User Status	showUserList, showFailinguserList, showUserStatus
Creating Connector Users	addConnectorUser
Deleting Hub Users	removeHubUser
Configuring User-Level Synchronization	updateUserSyncEnablementFlag, updateConnectorUserDomainSyncLevel

4.1.1 Managing BDSS Components Using MBeans

You can configure these MBeans through a JMX console (such as JConsole) supported by any J2EE-compliant server. This document describes accessing the MBeans using Oracle Enterprise Manager 11*g* Fusion Middleware Control (the Enterprise Manager console).

4.1.1.1 Accessing MBeans through the Oracle Enterprise Manager Console

To access the MBeans through the Oracle Enterprise Manager console:

1. Select **Application Deployment**.

2. Select **System MBean Browser** from the list (Figure 4–1). The System MBean Browser page appears. The navigation tree view displays the MBeans registered to the application.

Figure 4–1   Selecting the System MBean Browser
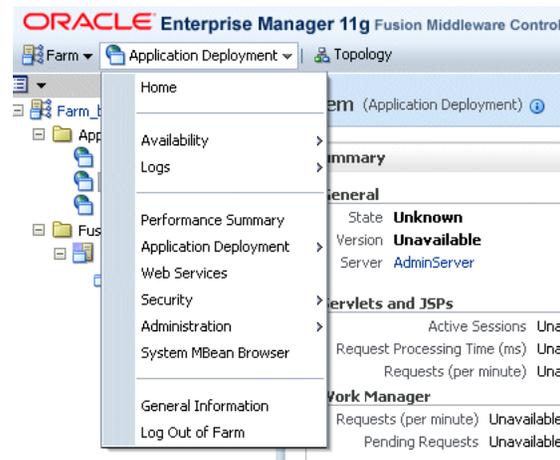
3. Select an MBean to view its attributes, operations, statistics, and notifications. Oracle Enterprise Manager organizes these components as tabs.

4. Click **Invoke** to commit any changes to the MBean's operations. Click **Apply** to commit any changes to the MBean's attributes.

4.1.1.2 Accessing MBeans through JConsole

To access the MBeans using JConsole, select the **MBeans** tab:

1. Click the **MBeans** tab.

2. Expand a MBean node in the navigation tree. The attributes, operations, and notifications of the MBean appear as subnodes.

3. To invoke operations:

   a. Select the **Operations** node in the tree. All of the operations appear as buttons in the right pane. The parameters appear as fields.

   b. Update the parameter fields as needed for an operation.

   c. Click the operation button to invoke the operation.

4. To update an individual operation:

   a. Expand the **Operations** node in the tree. The operation nodes appear.

   b. Select an operation in the tree to display its MBeanOperationInfo and Descriptor information in the right pane. The operation itself is represented as a button, which you click to invoke the operation.

   c. Update the MBeanOperationInfo as needed.

   d. Click the button for the operation to invoke the operation.

5. To set the attributes of an MBean:

   a. Expand the **Attributes** node from the tree. The attributes node appear.

   b. Select an attribute. The attribute's value, its MBeanAttributeInfo, and its descriptor appear in the right pane.

   c. Set the attribute by first selecting it and then editing it.

Refer to Java SE Technical Documentation (`http://docs.oracle.com/javase/7/docs/index.html`) for information on using JConsole.

4.1.2 Specifying the Synchronization Direction

The MBean operations enable you to set the inbound, outbound, and full synchronization direction at the connector, domain, and user levels. For more information, see Table 4–15.

Connector Level Synchronization

You can set the maximum synchronization direction for every domain that the connector synchronizes. For information about setting synchronization at the connector level, see Section 4.2.2.1, "Configuring Connector-Level Synchronization and Event Notification."

Domain Level Synchronization

You can set the synchronization direction for a specific data domain for a connector. For information about setting synchronization of the domain level, see Section 4.5.1.1, "Configuring Domain-Level Synchronization."

User-Level Synchronization

You can set the synchronization direction for a particular user for a particular domain. This is the most restrictive level of synchronization. For information about setting synchronization at the user level, see Section 4.7.3.1, "Enabling Synchronization for Hub Users."

4.2 Managing Connectors

Use the operations of ServerDomainConfigMBean listed Table 4–3 to create, delete and view operations of this MBean that enable you to manage the configuration details for connectors.

Table 4–3   Connector Management Operations of the ServerDomainConfigMBean

Operation	Administrative Task
showConnectorDetails	Viewing the Connector Configuration
updateConnectorForPIM	Adding and Updating Connectors
deleteConnector	Deleting a Connector

4.2.1 Viewing the Connector Configuration

To view the configuration for all connectors or for a specific connector:

1. Select the showConnectorDetails operation.

2. To view the configuration for a specific connector, enter the display name for the connector as the value for the `connectorName` attribute. Leave the field blank to view the configuration information for all connectors.

3. Click **Invoke**.

4.2.2 Adding and Updating Connectors

The `updateConnectorForPIM` operation of the ServerDomainConfigMBean enables you to create or update a connector and also enables you to specify how the connector synchronizes.

4.2.2.1 Configuring Connector-Level Synchronization and Event Notification

You enable synchronization for a connector by setting the `EnableForSync` parameter to `true` and then selecting a synchronization level (`Full`, `Inbound`, `Outbound`, or `None`). Setting the synchronization at the connector level sets the direction for every domain synchronized by the connector.

The `updateConnectorForPim` operation also allows you to enable the connector to send event notifications. Using this operation, you can also enable a connector to inform BDSS when Insert, Update, or Delete events occur for user data.

> **Note:** Perform event based synchronization through MBeans instead of directly setting the `SYNC_EVENTS_ENABLED_FLG`, `SUPPORTS_EVENTS_FLG`, and `USER_EVENT_FLG` flags in the CONNECTORS table.

To add or update a connector:

1. Select the `updateConnectorForPIM` operation.

2. Define the parameters listed in Table 4–4 and then click **Invoke**.

Table 4–4   Parameters for the updateConnectorForPIM Operation

Parameter	Description
ConnectorName	The display name for the connector. If you do not define a name for the connector, the BDSS creates one. Otherwise, BDSS modifies the existing one.
SupportsEventNotifications	Select `true` if the connector can notify BDSS when Insert, Update, or Delete events occur for a user's data. Otherwise, select `false` to prevent a connector from notifying BDSS of these events.
syncEventsEnabled	Select `true` to enable the connector to send event notifications. (A connector can support event notification even if it is not prompted to send them).
PimServerType	Specify the PIM server type, such as Exchange2007. You can configure other types of PIM servers.

Table 4–4   (Cont.) Parameters for the updateConnectorForPIM Operation

Parameter	Description
MaxSyncLevel	The maximum synchronization level for the connector:
	■ Full: Enables bidirectional (that is, both inbound and outbound) synchronization. If you set the MaxSyncLevel parameter to Full, then the connector can both propagate and extract Create, Read, Update, and Delete events between the Hub and the PIM server.
	■ Inbound: Enables propagation of Create, Update, and Delete record event data into a PIM server. If you select Inbound Only as the value for the MaxSyncLevel parameter, then it does not extract data from the PIM server.
	■ Outbound: Enables the connector to extract Create, Read, Update, and Delete record event data from the PIM server. If you select Outbound Only as the value for the MaxSyncLevel parameter, then it does not propagate Create, Read, Update, and Delete record event data to the PIM server.
	■ None: Setting the MaxSyncLevel parameter to None prevents the connector from synchronizing.
EnabledForSync	Select true to enable synchronization. Select false to disable synchronization.

4.2.3 Deleting a Connector

To delete a connector:

1. Select the deleteConnector operation.

2. Specify the ConnectorName parameter (described in Table 4–4) and then click **Invoke**.

4.3 Managing BDSS Server Instances

Use the operations of the ServerDomainConfigMBean listed in Table 4–5 to create or delete the server instances used by the connectors.

Table 4–5   Operations for Managing Server Instances

Operation	Administrative Task
updatePimServerInstance	Adding and Updating BDSS Instances
deletePimServerInstance	Deleting a PIM Server Instance

4.3.1 Adding and Updating BDSS Instances

To add or update a BDSS instance:

1. Select the updatePimServerInstance operation.

2. Specify the parameters listed in Table 4–6 and then click **Invoke**.

Table 4–6   Parameters of the updatePimServerInstance Operation

Parameter	Description
PimServerInstance	The name of the PIM server instance. This should be the name of the PM server as it would be returned by Active Directory or by the network directory service used by your network.

Table 4–6   (Cont.) Parameters of the updatePimServerInstance Operation

Parameter	Description
ConnectorName	The display name for the connector for which the PIM server instance is configured.

4.3.2 Deleting a PIM Server Instance

To delete a PIM server instance:

1. Select the deletePimServerInstance operation.

2. Specify the parameters listed in Table 4–6 and the click **Invoke**.

4.4 Managing Connector Instances

A connector instance represents a specific installation of a connector. The instance can be a single connector or represent a group of connectors that are accessed from the same load-balanced URI (Universal Resource Identifier).

Use the operations of the ServerDomainConfigMBean listed in Table 4–7 to manage the instances of a connector.

Table 4–7   Operations for Managing Connector Instances

Operation	Administrative Task
updateConnectorInstance	Adding and Updating Connector Instances
deleteConnectorInstance	Deleting a Connector Instance

4.4.1 Adding and Updating Connector Instances

To add or update a **connector instance**:

1. Select the updateConnectorInstance operation.

2. Specify the parameters listed in Table 4–8 and then click **Invoke**.

Table 4–8   Parameters of the updateConnectorInstance Operation

Parameter	Description
connectorInstName	The display name for the connector instance
connectorName	The name of the connector for which the connector instance is configured. You can find this name using the showConnectorDetails operation. Although the SQL scripts seed the names for the Exchange 2007 and BPEL Task connectors, you can add a different name.
pimServerInst	The PIM server instance used by this connector
url	The URL of the connector's Web service

4.4.2 Deleting a Connector Instance

To delete a connector instance:

1. Select the deleteConnectorInstance operation.

2. Specify the connectorInstName and connectorName parameters (described in Table 4–8) and then click **Invoke**.

4.5 Managing Domains

A domain is a class of records, such as a calendar, business contacts, personal contacts, or tasks. Each connector has a set of domains on the PIM servers (connector domains) that is mapped to a central set of Hub domains. A Hub domain is an abstract representation of a domain defined in the XSD (XML Schema Definition) for the Hub. For more information, see Chapter 8, "Mapping Connector Fields to Hub Fields."

Use the operations of the ServerDomainConfigMBean listed in Table 4–9 to manage domains.

Table 4–9   Operations for Managing Domains

Operation	Administrative Task
updateDomainForConnector	Creating a Domain for a Connector
deleteDomainForConnector	Deleting a Domain from a Connector

4.5.1 Creating a Domain for a Connector

To add or update a domain for a connector:

1. Select the updateDomainForConnector operation.

2. Specify the parameters listed in Table 4–10 and then click **Invoke**.

Table 4–10   Parameters of the updateDomainForConnector

Parameter	Description
domainName	The name of the domain
domainType	The type of domain, such as *Task*
connectorName	The display name of the connector for which the domain is configured
synclevel	The maximum synchronization level for the domain: ■ Full ■ Inbound Only ■ Outbound Only ■ None
Priority	Select an integer that determines the priority that the connector domain takes over other connector domains of the same type. You can use any integer (negative, positive or 0) to set the priority. Domains with priority of -2 would be synchronized before domains with a priority of 1. For more information, see Section 4.5.1.1. You must set the Priority parameters to different values for the same domain on each connector. For example, you cannot set the Tasks domain priority to 1 on both the Exchange 2007 connector and another type of connector. If the priorities match for the same domain, then BDSS throws exceptions.

4.5.1.1 Configuring Domain-Level Synchronization

The updateDomainForConnector operation enables you to set the synchronization configuration for a specific domain for a connector through its synclevel and Priority parameters. The synclevel parameter sets the level of synchronization for the domain and the Priority parameter determines which record change event wins a synchronization conflict between record changes from different PIM servers. The

Engine synchronizes Hub domains serially, not concurrently. For example, it synchronizes Hub domain A before domain B during a given synchronization session for a user.

> **Note:** Any number of connectors can link to a Hub domain concurrently. Connector domains that share the same parent Hub domain must have a unique priority number.

The synchronization levels available using the `synclevel` parameter are as follows:

Full

Enables bidirectional synchronization of a Hub domain across the connectors. If you set the `synclevel` parameter to `Full` for a connector domain, then the connector can both propagate and extract Create, Read, Update, and Delete events between the domain and its corresponding domain on another PIM server.

Inbound Only

If you select `Inbound Only` for a connector domain, then the Engine only propagates record change event data to the PIM for that connector domain. In this case, the Engine does not extract data from the PIM server for the corresponding connector domain when you select this option.

Outbound Only

A domain with an outbound only synchronization level enables the connector to extract Create, Read, Update, and Delete record event data from the corresponding domain on the PIM server. If you select `Outbound Only` as the value for the `synclevel` parameter, then the connector does not propagate Create, Read, Update, and Delete record event data to the PIM server for this domain.

None

Setting a synchronization level of `None` for a connector domain prevents a domain from participating in either inbound or outbound synchronization. No Create, Read, Update, or Delete record event data is propagated to, or extracted from, the corresponding domain on another PIM server.

4.5.2 Deleting a Domain from a Connector

To delete a domain from a connector:

1. Select the `deleteDomainForConnector` operation.

2. Specify the `domainName` and `connectorName` parameters (described in Table 4–10) and then click **Invoke**.

4.6 Managing Domain Filters

Filters are specific to connector domains and enable you to exclude a set of records from the extract result set based on defined criteria. BDSS supports any filter used by a connector.

> **Note:** For Oracle Fusion Middleware 11*g* release 1, BDSS ships with a coordinated universal time (UTC) date-time value filter that the Exchange 2007 Connector uses to remove old task records. Using this filter, the Exchange 2007 Connector evaluates the due date of each task record and excludes the record from the extract result set if the filter's date value is more recent than the record's due date.

4.6.1 Creating a Filter for a Domain

To create or update a filter for a connector's domain:

1. Select the `updateFilterForDomain` operation of the ServerDomainConfigMBean.

2. Specify the parameters listed in Table 4–11 and then click **Invoke**.

Table 4–11   Parameters of the updateFilterForDomain Operation

Parameter	Description
domainFilterName	The name of the filter. This name must be unique within each connector domain
domainName	The name of the domain for which this filter is configured
connectorName	The display name of the connector that owns the filter
DomainFilterValue	The value configured for this filter
DomainFilterSeq	An integer greater than 0 that determines the order that this filter is applied compared to other filters for this domain

4.6.2 Deleting a Filter from a Domain

To delete a filter from a domain:

1. Select `deleteFilterForDomain`.

2. Specify the `domainFilterName`, `domainName`, and `connectorName` parameters and then click **Invoke**.

4.7 Managing Users

The operations of the UserConfigMBean (listed in Table 4–12) enable you to view user status, create and delete users, and set the synchronization configuration of domains on a per-user basis.

Table 4–12   Operations for Managing Users

Operation	Task
`showUserList`, `showFailinguserList`, and `showUserStatus`	Viewing User Status
`addConnectorUser`	Creating Connector Users
`removeHubUser`	Deleting Hub Users
`updateHubUserSyncEnableFlag`	Enabling Synchronization for Hub Users
`updateUserDomainSyncLevel`	Setting the Synchronization Levels for a User's Domains

4.7.1 Viewing User Status

The operations listed in Table 4–13 enable you to view the synchronization status for users.

Table 4–13   Synchronization Status Operations

Operation	Description
showUserList	Returns a list of all users and their respective status
showFailinguserList	Returns a list of all users that are currently failing to synchronize
showUserStatus	Displays the status for a user, including record level status. There is an optional flag that forces the operation to only return failing items.

4.7.2 Creating Connector Users

The addConnectorUser operation adds a new user to a connector and also associates the user with a **Hub user**. This operation also creates a Hub user for the connector user if one does not exist.

> **Note:**   If you enable synchronization for a user, then you must also create a user for a connector. The Hub user must also enable the connector user to synchronize by setting the updateUserSyncEnableFlag to true. See Section 4.7.3 for information about setting the synchronization level for a user.

Table 4–14   Parameters of the addConnectorUser Operation

Parameter	Value
hubUserName	The name of the **Hub user**
connectorUserId	The ID of the connector user. For Exchange 2007 users, enter this value in the format of *firstname.lastname@mailserver.com*.
target	The location where the user's data is stored. For most Exchange users, this value is root. See also Section C.1.5, "PIM Domain Target."
connectorName	The name of the connector

4.7.2.1 Deleting Hub Users

The removeHubUser operation removes a Hub user from the database and all of the connector users associated with the Hub user. To remove a Hub user:

1. Select the removeHubUser operation.

2. Define the hubUserName parameter (described in Table 4–14) and then click **Invoke**.

4.7.3 Configuring User-Level Synchronization

The updateUserSyncEnablementFlag and updateConnectorUserDomainSyncLevel operations set the synchronization configuration for users.

4.7.3.1 Enabling Synchronization for Hub Users

Setting the `updateUserSyncEnablementFlag` to `true` enables a user to synchronize for all subsequent synchronization sessions. Setting the operation to `false` prevents the user from synchronizing. To enable synchronization for a Hub user:

1. Set the `updateUserSyncEnablementFlag`.

2. Define the `hubUserName` parameter (described in Table 4–14) and then click **Invoke**.

4.7.3.2 Setting the Synchronization Levels for a User's Domains

The synchronization levels configured for the domains described in Section 4.5.1.1 are affected by the synchronization levels configured for domains on a per-user basis through the `updateUserDomainSyncLevel` operation. Using this operation, you can alter the synchronization level for a connector user for a given domain.

> **Note:** The `updateUserDomainSyncLevel` operation creates a domain context for a user if one does not exist.

The synchronization levels are:

- Full
- Inbound Only
- Outbound Only
- None

These levels, when applied to a user, change the final synchronization direction for a domain, as listed in Table 4–15.

Table 4–15   Domain Synchronization Level Directions

Connector Domain Synchronization Direction	User Synchronization Direction	Final Direction
Full	Full	Full
Full	Inbound Only	Inbound Only
Full	Outbound Only	Outbound Only
Inbound Only	Full	Inbound Only
Inbound Only	Inbound Only	Inbound Only
Inbound Only	Outbound Only	None
Outbound Only	Full	Outbound Only
Outbound Only	Inbound Only	None
Outbound Only	Outbound Only	Outbound Only
None	Any synchronization level (Full, Inbound Only, Outbound Only, None)	None
Any synchronization level (Full, Inbound Only, Outbound Only, None)	None	None

To set the synchronization level for a user's connector domains:

1. Select the `updateUserDomainSyncLevel` operation.

2. Select the synchronization level.

3. Specify the following parameters:

   ■ `connectorUserId` (described in Table 4–14)

   ■ `connectorName`

   ■ `domainName` (described in Table 4–10)

   ■ `syncLevel` (described in Table 4–10)

4. Click **Invoke**.

4.8 Managing Profiles

A profile stores the configuration metadata for a connector. Connectors retrieve their profiles, which are conceptually equivalent to the Windows `.ini` configuration file, by using the `GetConfigurationMetadata` interface. Each profile includes one or more sections that contain parameters. Each parameter has a single value. The ProfileRuntimeInfo MBean enables you to manage the profiles for a connector.

4.8.1 Viewing Profiles

The `displayAllProfiles` and `displayProfile` operations enable you to view connector profiles.

4.8.1.1 Viewing All Profiles for a Connector

To view a list of all parameters grouped by profile and by section:

1. Select the `displayAllProfiles` operation.

2. Click **Invoke**.

4.8.1.2 Viewing a Specific Profile

To view a specific profile:

1. Select the `displayProfile` operation.

2. Enter a profile name.

3. Click **Invoke**. The parameters appear, grouped by section of the selected profile.

4.8.2 Managing Profile Parameters

The `setProfileParameter` and `removeProfileParameter` operations enable you to create, update, and delete profiles.

4.8.2.1 Creating a Profile

To create a profile:

1. Select the `setProfileParameter` operation.

2. Enter the name of the profile.

3. Enter the name of the section.

4. Specify the parameters for the profile (name, value, and comment).

5. Click **Invoke**.

4.8.2.2 Updating a Profile

To update a profile:

1. Select the `setProfileParameter` operation.

2. Enter the name of profile.

3. Enter the name of the section.

4. Update the parameters as needed.

5. Click **Invoke**.

4.8.2.3 Deleting a Parameter from a Profile

To remove a parameter from a selected profile:

1. Select the `removeProfileParameter` operation.

2. Specify the name of the parameter, profile, and the section to ensure that you are removing the correct parameter.

3. Click **Invoke**.

5

Configuring the BDSS Hub Services

This chapter describes the basic configuration for the Hub and connectors.

This chapter includes the following topics:

- Overview of Hub Configuration
- Configuring the Dispatcher
- Configuring the Engine
- Creating Connector Configuration Profiles
- Configuring the FtsKeyFields Profile

5.1 Overview of Hub Configuration

Configuration for the Hub and Dispatcher components is exposed as the attributes and operations of MBeans. You can configure these MBeans using the Oracle Enterprise Manager's System MBean Browser or through any JMX console (such as JConsole) that is supported by a J2EE-compliant server. For more information, see Section 4.1.1, "Managing BDSS Components Using MBeans."

5.2 Configuring the Dispatcher

The attributes for the Hub and Dispatcher are exposed through the DispatcherSettings MBean. Table 5–1 lists these attributes.

Table 5–1   Attributes of the DispatcherSettings MBean

Attribute	Description
ChunkSize	The number of users contained in the message sent to the Hub for synchronization. Use this attribute to balance the load to each instance of a running Engine. The default value is 20 users per message
HubEndPointURL	The URL used by the Dispatcher to connect to the DispatcherHub Web service on the server hosting the Hub. For example, a typical Hub URL for Oracle WebLogic Server is `http://<HUB_SERVER_IP_ ADDRESS>:7001/BDSSHubWebServices/DispatcherHubW ebService?WSDL`

5.3 Configuring the Engine

The attributes of the Engine are exposed through the EngineSettings MBean. Table 5–2 lists these attributes.

Table 5–2   Attributes of the EngineSettings MBean

Attribute	Description
EngineEndPointURL	The URL used by the Engine to enable the connectors to extract requests and data update request responses to the Engine.
	To find this URL using the Oracle WebLogic Server Administration Console:
	1. Select **Deployments**.
	2. Expand the tree structure of BDSSHub.
	3. Select the **EngineCallBackInterface** node. The Settings for EngineCallback page appears.
	4. Select the **Testing** tab.
	5. Expand the **EngineCallbackInterface** node.
	6. Select the ?WSDL test point.
ExtractResponseTimeOut	The number of milliseconds that the Engine waits to process an extract response from a connector. The default value is 60000 milliseconds. The timeout value must be sufficiently large to prevent a timeout from occurring under normal circumstances. This value depends on several factors that include:
	■ The size of the largest record set that can be extracted.
	■ The number of Hub domains that are to be extracted. This has bearing because Hub domains are processed serially by the Engine for a synchronization session. Therefore, Hub domain B must wait longer to retrieve its extract responses if Hub domain A were processed first.
	■ Other factors, including system load and communication latency, that affect the ability of a Engine to retrieve and process extract response messages from connectors before the timeout occurs.
	If a timeout occurs, then synchronization of the corresponding Hub domain is terminated for the affected user synchronization session.
MessageTimeToLive	The number of milliseconds that a message containing an extract response remains on the JMS queue before being removed. The default value is 60000 milliseconds. Calculate this timeout value using an analysis similar to the one performed for the ExtractResponseTimeOut attribute.
RuntimeLibraryURL	The URL that the Engine sends to the connectors to enable them to communicate with the Connector Run-Time Library.
	To find this URL using Oracle WebLogic Server Administration Console:
	1. Select **Deployments**.
	2. Expand the tree structure of BDSSHub.
	3. Select the **ConnectorRuntimeInterface** node. The Settings for ConnectorRuntimeInterface page appears.
	4. Select the **Testing** tab.
	5. Expand the **ConnectorRuntimeInterface** node.
	6. Select the ?WSDL test point.

5.4 Creating Connector Configuration Profiles

Using the setProfileParameter operation described in Section 4.8.2, "Managing Profile Parameters," you can create or update the parameters for configuration profiles.

5.5 Configuring the FtsKeyFields Profile

The FTSKeyFields is a Hub profile used by all connectors. BDSS uses key fields when it first synchronizes a record to determine if a record from one PIM server is the same as a record from another PIM server. If the values in all key fields of a record match, then BDSS assumes that the record is the same. If any key field does not match, then BDSS assumes that the record is a unique record.

> **Note:** BDSS does not continue using key fields to identify a record after it has made a match. From that point, BDSS uses record IDs.

Table 5–3 lists the parameters grouped into the FtsKeyFields profile.

Table 5–3   Parameters of the FtsKeyFields Profile

Parameter	Description
KeyField1	The first field used for record matching
KeyField2	The second field used for record matching

6

Configuring the BDSS Connector for Microsoft Exchange 2007

This chapter describes the profile parameters for the Oracle BDSS Connector for Microsoft Exchange 2007 (the Exchange 2007 Connector).

This chapter includes the following topics:

- Overview of Exchange 2007 Connector Configuration
- Configuring the Microsoft Exchange 2007 Connector for Data Exchange
- Configuring Record Transformation for the Exchange 2007 Connector
- Configuring the Microsoft Exchange 2007 Connector's PIM Transport Component
- Configuring Domain Synchronization
- Configuring Data Filtering
- Configuring the Microsoft Exchange 2007 Connector to Use Active Directory
- Configuring Event Receiving for the Exchange 2007 Connector
- Configuring the Exchange 2007 Connector to Use the Microsoft Autodiscover Service
- Testing Domain Synchronization using the Exchange 2007 Connector

6.1 Overview of Exchange 2007 Connector Configuration

As described in Section 4.8, "Managing Profiles," you can create or update the configuration profile for a connector using the `setProfileParameter` operation. The Exchange 2007 Connector ships with the following profiles:

- Hub Transport
- Task
- Contact
- Calendar
- Active Directory
- Auto Discovery
- Event Receiver

6.2 Configuring the Microsoft Exchange 2007 Connector for Data Exchange

The Hub Transport section includes the following parameters (listed in Table 6–1), which enable the Exchange 2007 Connector to communicate with the **Hub**. For information on the Hub Transport, see Section 2.2.1, "Hub Transport."

> **Note:** You must create the parameters listed in Table 6–1 using the `setProfileParameter` operation. The profile name must be the name of the connector. For more information, Section 4.8.2, "Managing Profile Parameters."

Table 6–1   Parameters of the Hub Transport Section

Parameter	Description
Pim Transport Implementation Class	`oracle.bdss.connectors.exchange2007.Exchange2007PimTransportAdapter`
Extract Response Batch Size	The number of records in an extract response. For Oracle Fusion Middleware 11*g* release 1 of BDSS, set this value to represent a large number (10,000 or more) of records, so that it exceeds the number of records in any given Exchange folder configured for synchronization for a user. Typically, most users have thousands of records as opposed to tens of thousands of records.

6.3 Configuring Record Transformation for the Exchange 2007 Connector

The parameters of the hubDomainName section (listed in Table 6–2) enable you to configure the Exchange 2007 Connector to perform translations of record data to and from the Hub format to the PIM format by setting the URL locations of the XSLT and XSD files.

> **Note:** If you configure the parameters to use URIs for the transformation files, do not use the `file:///./` form. Instead, enter the URI in the following form:
>
> `C:/directory1/directory2.../file.NNN`
>
> For example, if `c:/fusion_sync` is the `<root>` directory for transformer files, enter the following URI:
>
> `c:/fusion_sync/transform/exchange/task/exchange2007task.xsd`

Table 6–2   Parameters of the hubDomainName Section

Parameter	Description
HubToPimXSL	The URL to the location of the XSL file that transforms a record from the Hub XML format to a PIM XML format.
PimToHubXSL	The URL to the location of the XSL file that transforms a record in a PIM XML format to the Hub XML format.
PimXSD	The URL to the location of the XSD for the PIM domain.

6.4 Configuring the Microsoft Exchange 2007 Connector's PIM Transport Component

Use the parameters of the PimTransport section (listed in Table 6–3) to set the Microsoft Exchange 2007 Connector's PIM Transport component.

Table 6–3   The PimTransport Section Parameter

Parameter	Description
TryUrlsWithoutDomain	If set to `true`, then the PIM Transport attempts the autodiscover, LDAP, and EWS URLs that do not contain the domain. The default is `false`.

6.5 Configuring Domain Synchronization

The Exchange 2007 Connector enables you to configure domain synchronization through its Contact, Task, and Calendar sections.

6.5.1 Configuring Contact Synchronization

The parameters of the Contact section (listed in Table 6–4) enable the Exchange 2007 Connector to synchronize the Contact domain.

Table 6–4   Parameters for Configuring Contact Domain Synchronization

Parameter	Description
Folder Class	The Folder class for the domain. The default value is `IPF.Contact`.
Pim Domain Id	The PIM domain identifier, used as the message class property for each class that is synchronized. The default value is `IPM.Contact`.
Pim Xsd	The URI of the `exchange2007contact.xsd` file.
PimToHubUrl	The URI of the `Exchange2007ContactToHubContact.xsl`.
HubToPimUrl	The URI of the `HubContactToExchange2007Contact.xsl`.

6.5.2 Configuring Task Synchronization

The parameters of the Task section (listed in Table 6–5) enable the Exchange 2007 Connector to synchronize the Task domain.

Table 6–5   Parameters for Configuring Task Domain Synchronization

Parameter	Description
Pim Xsd	The URI of to the `exchange2007task.xsd` file.
PimToHubUrl	The URI of to the `Exchange2007TaskToHubTask.xsl`.
HubToPimUrl	The URI of to the `HubTaskToExchange2007Task.xsl`.
Domain Class Type	The Exchange Web Service interface class that is used when synchronizing the domain. The default value is `TaskType`.
Exchange Domain Batch Size	The maximum number of records downloaded from Exchange when extracting records from the domain target. Valid values are 1 through 512. The default value is `512`.
Folder Class	The folder class for the domain. The default value is `IPF.Task`.

Table 6–5   (Cont.) Parameters for Configuring Task Domain Synchronization

Parameter	Description
Body Type	How the Body field for the domain is synchronized. Valid values are `Text` or `HTML`. The default value is `Text`.
Conflict Resolution Type	Determines how Exchange handles a conflict type. Valid values are `AutoResolve`, `AlwaysOverwrite`, and `NeverOverwrite`. The default value is `AlwaysOverwrite`.
Disposal Type	Determines how Exchange handles deleted records. Valid values are `HardDelete`, `SoftDelete`, and `MoveToDeletedItems`. The default value is `MoveToDeletedItems`.
Pim Domain id	The PIM domain identifier, used as the message class property for each synchronizing record. The default value is `IPM.Task`.

6.5.3 Configuring Calendar Synchronization

The parameters of the Calendar section (listed in Table 6–6) enable the Exchange 2007 Connector to synchronize the Calendar domain.

Table 6–6   Parameters for Configuring Calendar Domain Synchronization

Parameter	Description
Time Zone Xsd	The URI to `timezoneregistryschema.xsd`.
VEvent Xsd	The URI to `veventasxml.xsd`.
PimToHubUrl	The URI to `exchange2007calendartohubcalendar.xsl`.
Pim Xsd	The URI to `pimicalbody.xsd`.
Calendar Item Type Xsd	The URI to `calendaritemtype.xsd`.
Time Zone Xml	The URI to `timezoneregistry.xml`.
HubToPimUrl	The URI to `hubcalendartoexchange2007calendar.xsl`.
CalendarItemToVEvent Xsl	The URI to `calendaritemtypetoveventasxml.xsl`.
VEventToCalendarItem Xsl	The URI to `veventasxmltocalendaritemtype.xsl`.
Body Type	The URI to `veventasxmltocalendaritemtype.xsl`. The default value is `Text`.
Conflict Resolution Type	Determines how Exchange handles a conflict type. Valid values are `AutoResolve`, `AlwaysOverwrite`, and `NeverOverwrite`.The default value is `AlwaysOverwrite`.
Disposal Type	Determines how Exchange handles deleted records. Valid values are `HardDelete`, `SoftDelete`, and `MoveToDeletedItems`. The default value is `MovetoDeletedItems`.
Domain Class Type	The Exchange Web Service interface class that is used when synchronizing a domain. The default value is `CalendarItemType`.
Exchange Domain Batch Size	The maximum number of records downloaded from Exchange when extracting records from the domain target. Valid values are 1 to 512, inclusive. The default value is 512.
Fanning Limit	The number of records created when a recurring calendar record cannot be natively represented in Exchange. Valid values are 1 to 100, inclusive. The default value is 30.
Folder Class	The folder class for the domain. The default value is `IPF.Appointment`.

Table 6–6   (Cont.) Parameters for Configuring Calendar Domain Synchronization

Parameter	Description
Pim Domain id	The PIM domain identifier, used as the message class property of each record synchronized. The default value is `IPM.Appointment`.

6.6 Configuring Data Filtering

Filtering controls the records that are synchronized by the Exchange 2007 connector. For example, you can define a filter that ignores public records but synchronizes only public records. Filters are defined in an XML file. There is one such XML file per domain. BDSS ships with the following filters:

- Contact—Specifies records having an `IPM.Contact` item class should synchronize.

- Task—The filter specifies that records having an `IPM.Task` item class and a Due Date after 2009-12-31T00:00:00 should synchronize.

- Calendar—The filter specifies that records having an `IPM.Appointment` class and an End Date after 2009-12-31T00:00:00 should synchronize.

Table 6–7 lists the filtering parameters for the Contact domain.

Table 6–7   Parameters for Configuring Contact Filtering

Parameter	Description
Filter Condition	The URI to `contactfilter.xml`.
Body Type	How the Body field for the domain is synchronized. Valid values are `Text` or `HTML`. The default value is `Text`.
Conflict Resolution Type	Determines how Exchange handles a conflict type. Value values are `AutoResolve`, `AlwaysOverwrite`, and `NeverOverwrite`. The default value is `AlwaysOverwrite`.
Disposal Type	Determines how Exchange handles deleted records. Valid values are `HardDelete`, `SoftDelete`, and `MoveToDeletedItems`. The default value is `MoveToDeletedItems`.
Domain Class Type	The Exchange Web Service interface class that is used when synchronizing the domain. The default value is `ContactItemType`.
Exchange Domain Batch Size	The maximum number of records downloaded from Exchange when extracting records from the domain target. Valid values are 1 through 512. The default value is `512`.
Folder Class	The folder class for the domain. The default value is `IPF.Contact`.
Pim Domain Id	The PIM domain identifier, used as the message class property of each record synchronized. The default value is `IPF.Contact`.

Table 6–8 lists the parameters for task filtering.

Table 6–8   Parameters for Configuring Task Filtering

Parameter	Description
Filter Condition	The URI to `taskfilter.xml`.

Table 6–9 lists the parameters for calendar filtering.

Table 6–9   Parameters for Configuring Calendar Filtering

Parameter	Description
Filter Condition	The URI to `calendarfilter.xml`.

6.7 Configuring the Microsoft Exchange 2007 Connector to Use Active Directory

The parameters of the Active Directory section (listed in Table 6–10) enable you to configure the Exchange 2007 Connector to use Microsoft Active Directory to query for and obtain Autodiscover SCP URLs.

Table 6–10   Parameters for the Active Directory Section

Parameter	Description
Port	The LDAP port number used when querying the Active Directory. The default port is 636 for secure LDAP. The non-secure port is typically 389.
ServerN (where *N* is a unique, nonzero ordinal that specifies an order preference).	Enter the name of the Active Directory server. The connector uses the name in the LDAP URL when querying the Active Directory. For example, if you enter *server360007*, then the connector uses *ldap://server36007:<ldapPort>/...* If you enter *server36007.us.example.com*, then the connector uses *ldap://server36007.us.example.com:<ldapPort>...*
Use SSL	Whether the connector should use secure LDAP. The default is value is *true*. In a development environment, set the value to *false* because the Domain Controller is not enabled for secure LDAP.
Referral	Can be `follow`, `throw` or `defer`. The default value is `follow`.
Max Query Result Size	Max number of results to return per Active Directory query. The default value is 20.
Max Query Time	The time (in milliseconds) to wait for an Active Directory query to return. The default value is 30 seconds (30000 milliseconds).
Read Timeout	The time (in milliseconds) to wait for a response before timing out. The default value is 5000.
Connect Timeout	The time (in milliseconds) to establish a connection to Active Directory before timing out. The default value is 5000.
Client Certificate Path	Set to `true` if Use SSL is set to true and the Domain Controller requires a client certificate. The default value is `false`.
Using Client Certificates	Indicates if the client provider uses LDAP certificates. The default value is `false`.

6.8 Configuring Event Receiving for the Exchange 2007 Connector

The parameters of the Event Receiver Section (listed in Table 6–11), enable you to configure the event receiver, a web service-based component used by the Exchange 2007 Connector's PIM Transport for event notification. The event receiver establishes subscriptions for Create, Update, and Delete events for each user, maintains the subscription metadata, receives change events from a client-access server, and informs BDSS when users have changes that require synchronization through the Run-Time Library API.

Table 6–11   Parameters for the Event Receiver Section

Parameter	Description
CHANGE_ME	The name of an Exchange 2007 connector as it appears in the NAME Column of the CONNECTORS table. The default value should be the URL of the deployed event receiver for the connector.
Status Frequency	The frequency, in minutes, that Exchange provides a notification for each subscription. The default value is 1.
Subscription Status Factor	The event receiver performs a check on all subscriptions to verify that the Exchange server has not terminated any subscriptions. This parameter sets frequency in which this check occurs. The frequency in which this check occurs is the value set for this parameter multiplied by the value set for the Status Frequency parameter. For example, if you set the Status Frequency parameter to 1 minute, and then set the *Subscription Status Factory* parameter to 3, then the event receiver performs the subscription check every 3 minutes. The default value is 3.

6.9 Configuring the Exchange 2007 Connector to Use the Microsoft Autodiscover Service

The Exchange 2007 Connector attempts to discover information about a user's Exchange account by querying the Microsoft Exchange 2007 Autodiscover service. There are Service Connection Point (SCP) URLs that point to the locations where the service is available. The connector does the following when attempting to discover a user:

1. Try each configured SCPN URL

2. If that fails, query each Active Directory server listed in the Server*N* configuration for a list of SCP URLs and attempt each of them.

3. If the first two steps fail, the connector attempts the following default URLs:

   – `<protocol>://<domainName>:<port>/autodiscover/autodiscover.xml`

   – `<protocol>://<domainName>.autodiscover:<port>/autodiscover/autodiscover.xml`

Using the parameters listed in Table 6–12, the Exchange 2007 Connector can forgo searching for the Autodiscover service if you provide the SCPN URLs.

Table 6–12 lists the following parameters of Auto Discovery section that enable you to configure the connector to use the Microsoft Exchange 2007 Autodiscover service.

Table 6–12   Parameters of the Auto Discovery Section

Parameter	Description
SCPN (where *N* is a unique, nonzero ordinal that specifies an order preference).	Enter either a full URL to the location of an Exchange Autodiscover service or the name of the server that runs the Autodiscover service. If you enter the URL to the location of an Exchange Autodiscover server, then the URL is constructed to include the given entry as follows:
	`<protocol>://<configuredValue>:<port>/autodiscover/autodiscover.xml`
	You can have 1 or more rows if and only if each SCPN is unique. When discovering a user, the connector tries each URL in the order specified by the SCPN parameter. For example, the URL for SCP1 is tried before the URL for SCP2, and so on.
Port	The port used for sending HTTP requests for user discovery. The default value is the typical SSL port, 443. Set the value to the HTTPS port if *Use SSL* is set to `true`. If it is set to `false`, then this value should be the HTTP port. The value is only used when attempting default SCP URLs (which are attempted when all SCP URLS have been tried (those in the SCPN configuration and those obtained from AD per the Server*n* configuration)
UseSSL	Indicates whether to use secure HTTP (SSL) when sending HTTP requests to discover users. The default value is `true`.
Remove Domain From Urls	Directs the connector to remove the domain information from a discovered SCP URLs obtained from Active Directory.
	For the development environment, set the value to `true` because DNS prevents the URL from being resolved when the URL contains a domain (that is, `http://server360007.us.example.com/autodiscover/autodiscover.xml` does not work). When the option set to true, the URL becomes `http://server360007/autodiscover/autodiscover.xml`
Read Timeout	The time (in milliseconds) to wait for a response before timing out. The default value is 5000.
Connect Timeout	The time (in milliseconds) to establish a connection to Auto Discover server before timing out. The default is 5000
Client Certificate Path	Set this value to the location of the PEM (Privacy Enhance Mail security certificate) file containing the client certificate that is used if Auto Discover requires client certificates
Client Certificate Key Path	Set this value to the location of the PEM (Privacy Enhance Mail security certificate) file containing the client certificate private key

6.10 Testing Domain Synchronization using the Exchange 2007 Connector

You can use the Exchange 2007 Connector for **reflective synchronization**. You can test the Hub's synchronization of the Task, Contact, and Calendar domains using the Exchange 2007 Connector, because it provides synchronization for multiple Hub domains. You can configure the Exchange 2007 Connector to present itself as any type of PIM server to the Hub, because server type is irrelevant to the Hub. If you configure a second Exchange 2007 Connector using the same PIM server type, the Hub can then

synchronize between two different Exchange users as if it were synchronizing between two users of different PIM servers.

To test Hub synchronization:

1. Configure the Pim Transport Implementation Class and Extract Response Batch Size parameters for each Exchange 2007 Connector whereby the value for the Profile is the name of the connector as it appears in the NAME column of the CONNECTORS table. For example, suppose you have deployed a single Exchange 2007 Connector and want to use the single instance to synchronize two Exchange 2007 users that are mapped to a single **Hub user**. Further, assume that Exchange2007 and Exchange2007_2 are in the NAME column of the CONNECTORS table and both have a SERVER_TYPE_CD corresponding to the Exchange2007 LOV value.

2. Add the following four rows to the PROFILES table using the `setProfileParameter` of the ProfileRuntimeInfo MBean. For more information, see Section 4.8, "Managing Profiles."

Table 6–13   Adding Profiles to Test Synchronization

Profile	Section	Parameter	Value
Exchange2007	Hub Transport	Pim Transport Implementation Class	oracle.bdss.connectors.exchange2007 pimtransport.PimTransport Adapter
Exchange2007	Hub Transport	Extract Response Batch Size	100000
Exchange2007_2	Hub Transport	Pim Transport Implementation Class	oracle.bdss.connectors.exchange2007.pimtransport.PimTransportAdapter
Exchange2007_2	Hub Transport	Extract Response Batch Size	100000

7

Configuring Security for BDSS

This chapter describes how to configure security between the Engine and the connectors.

This chapter includes the following topics:

- Overview of Communication Security
- Securing the Web Services Exposed by the Hub
- Configuring BDSS for HTTPS

7.1 Overview of Communication Security

Because the connector interface is a Web service, you can use a Web-based security protocol such as SSL to secure communications between the Engine and a connector. The connector implementation is solely responsible for the security between itself and the PIM server with which it communicates.

After you install BDSS, you can modify its configuration to secure the Web services exposed by both its Hub and connectors by completing the following tasks:

- Securing the Web Services Exposed by the Hub
- Configuring BDSS for HTTPS

7.2 Securing the Web Services Exposed by the Hub

To secure the Web services exposed by Hub components running on Integrated WebLogic Server, see *Oracle Fusion Middleware Securing Web Services for Oracle WebLogic Server*.

> **Tip:** After you configure BDSS for SSL, first deploy BDSS and then open a Web browser. Enter the HTTPS URLs for the connector Run-Time Library and the Hub Callback Library Web services to verify that they can accept secure requests.

7.3 Configuring BDSS for HTTPS

After the Web services are configured to use SSL, you must modify the URLs identifying the Connector Run-Time Library, Hub Callback, and Exchange 2007 Connector to reflect the secure protocol (HTTPS) and secure port, as described in the following steps:

1. Start JConsole (or any other MBean-compliant application) and then log in.

2. Navigate to the EngineSettings MBean.

   a. Change the `EngineEndPointURL` attribute. Specify HTTPS as the protocol and set the port as the SSL port used by Integrated WebLogic Server. For example:

   ```
   https://<system>:<secureport>/BDSSHubWebServices/EngineCallbackInterface?wsdl
   ```

   b. Change the `RuntimeLibraryURL` attribute. Specify HTTPS as the protocol and the SSL port used by Integrated WebLogic Server. For example:

   ```
   https://<system>:<secureport>/BDSSHubWebServices/ConnectorRuntimeInterface?wsdl
   ```

3. Navigate to the ServerDomainConfigMBean.

   a. In the `showAllConnectorsForPim` operation, specify Exchange 2007 as the `pimServerType` parameter value and click **showAllConnectorsForPim**.

   b. Note the PIM Server Name and Connector Identification values.

   c. In the `updateConnectorForPim` operation, for the `pimServerInstance` value, specify the value for PIM Server Name obtained in Step 3b. For the `connectorName` value, specify the value for Connector Identification obtained in Step 3b. For the URL, enter the HTTPS URL for the Exchange 2007 Connector.

   > **Note:** You also perform these steps if you create two Exchange users to test connector domain synchronization, but you specify `BPEL Tasks` as the value for the `pimServerType` attribute instead of `Exchange 2007`.

8

Mapping Connector Fields to Hub Fields

This chapter describes how records are transformed between the Hub and PIM servers.

This chapter includes the following topics:

- Overview of Data Transformation
- About Domain Schemas
- Data Side Effects Caused by Synchronization

8.1 Overview of Data Transformation

To enable the exchange of information between different types of PIM servers, the Hub defines a schema representation for each connector domain type in a schema (XSD) file. The data that is extracted from the PIM servers and returned to the Hub is transformed into the format defined by the Hub schema. Connectors transform the data into Hub XML format and also into the format of the PIM server for both inbound and outbound transactions. For every inbound record extracted, the connector must convert it from the native PIM API format to an XML message that conforms to a PIM XML schema defined for the connector domain. After the connector constructs the PIM XML message, it then uses XSLT to transform the message into a Hub XML message. This message conforms to a provided Hub XML schema.

To enable data to map between the Hub domain and the connector domain, a connector's Transformer component includes the schemas for both the Hub and connector domains and also XSLT documents. As described in Appendix C, "Connector API," an `UpsertRecord` is a representation of a Hub record that flows between the Hub and connectors. This data structure includes a member called the `hubRecordData`, which contains the XML representation of the Hub record. Because the Hub record's XML must conform to the XSD for the Hub domain, connector developers must create the following:

- An XSD for a corresponding connector domain that maps to the Hub domain
- An XSL definition for transforming the Hub XML representation of the record to the PIM XML representation of the record
- An XSL definition for transforming the PIM XML representation of the record to the Hub XML representation of the record

During extractions from the PIM server, a connector must ensure that each extracted record is first converted into a PIM XML message. It then uses the XSLT to convert the message from PIM XML to Hub XML. For record events pushed to the PIM server (that is, `Create` and `Update` operations), the connector uses XLST to convert the Hub XML record to a PIM XML record before writing to the PIM server.

8.2 About Domain Schemas

During record transformation, the BDSS connectors apply an XLST document that contains references to the Hub and connector domain schemas, thus providing robust document validation for the record data contained in the messages that pass between the Hub and the connectors.

A domain schema has the following attributes defined for each domain field:

- Schema Data Types

  A connector must support the following schema data types:

  - W3C Schema Data Types

  - Hub Schema Data Types

  - Connector Schema Data Types

- PIM field class

  A connector can use a PIM field class during transformation or to control flow. For more information, see Section C.1.4, "Field Class."

8.2.1 Schema Data Types

A schema uses the following schema data types:

- W3C Schema Data Types

- Hub Schema Data Types

- Connector Schema Data Types

8.2.1.1 W3C Schema Data Types

The data types used by the Hub are a subset of the native W3C XML Schema (WXS) data types. These data types are defined at `http://www.w3.org`. Connectors use any WXS data types for their schema documents.

The subset of the native W3C XML Schema (WXS) data types used by the Hub schemas are as follows.

- `xsd:string`

- `xsd:boolean`

- `xsd:dateTime`

> **Note:** Because `xsd:dateTime` generally allows the ISO 8601 date
> and time representations, all varieties of date and time can be
> specified in standard coordinated universal time (UTC).
>
> All date and time Hub fields that are passed between the Engine and
> the connector must:
>
> 1. Be in UTC time and in the format of YYYY-MM-DDhh:mm:ss.fff. In this
>   format:
>
>   - *YYYY* is the 4 digit year, such as 2007.
>
>   - *MM* is the month (January is 01 and
>   December is 12.)
>
>   - *DD* is the day of the month (starting from
>   01).
>
>   - *hh* is the hour of day in 24-hour clock time
>   (00 through 23, inclusive).
>
>   - *mm* is the minute of the hour (00 through 59,
>   inclusive)
>
>   - *ss* is the second of the minute (00 through 59,
>   inclusive).
>
>   - *fff* are fractional seconds (that is,
>   milliseconds—000 through 999, inclusive).
>
> 2. All date/time PIM field values have a corresponding UTC offset field in
>   the format of +HH:MM or -HH:MM:
>
>   - + indicates a positive UTC offset.
>
>   - - indicates a negative UTC offset.
>
>   - *HH* is the UTC offset hour in 24-hour clock
>   time (00 through 23, inclusive).
>
>   - *MM* is the UTC offset minute (00 through
>   59, inclusive).
>
> By providing both the UTC date-time and UTC offset for each of the
> PIM date-time fields, a receiving connector can calculate a local time if
> the PIM server uses local time or it can use the UTC date-time if the
> PIM server uses UTC date-time for the field.

- `xsd:nonNegativeInteger`

- `xsd:unsignedLong`

- `xsd:positiveInteger`

8.2.1.2 Hub Schema Data Types

Hub domains are constructed using W3C schema data types.

> **Caution:** If you alter the Hub schemas or Hub schema data types, you must also make corresponding changes to the XSLT documents for each connector. The XSLT documents must change to ensure proper transformation between the Hub and PIM schema representations.

Hub-Defined Data Types

In addition to the native W3C XML Schema (WXS) data types, BDSS provides the XSDs for such common PIM data types as phone numbers and addresses. These data types have only basic restrictions. For example, HubPhoneNumber is restricted to a simple string, as illustrated in Example 8–1.

Example 8–1   Predefined Restrictions to a Hub-Defined Data Type

```
<?xml version="1.0" encoding="windows-1252" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/pss/hub/ContactDomain"
            elementFormDefault="qualified">
  <xsd:simpleType name="HubPhoneNumber">
   <xsd:restriction base="xsd:string"></xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

Although BDSS does not define other restrictions, you can define more elaborate restrictions, if needed. The bold font in Example 8–2 illustrates additional restrictions to the length and format of the HubPhoneNumber data type.

Example 8–2   Additional Restrictions to a Hub-Defined Data Type

```
<?xml version="1.0" encoding="windows-1252" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/pss/hub/ContactDomain"
            elementFormDefault="qualified">
  <xsd:simpleType name="HubPhoneNumber">
   <xsd:restriction base="xsd:string"></xsd:restriction>
<xsd:pattern value="([0-9]){0,2}\([0-9]{3}\)[0-9]{3}\-[0-9]{4}( Ext:[0-9]{4})?" />
  </xsd:simpleType>
</xsd:schema>
```

The Hub-defined data types include:

- xsd:HubPercent

- xsd:HubUnsignedLong

- xsd:HubTokens

- xsd:HubUtcOffest

- xsd:HubPhoneNumber

- xsd:HubAddress

- xsd:HubState

- xsd:HubCountry

- xsd:HubEmailAddress

- xsd:HubNamePrefix

- `xsd:HubNameSuffix`

8.2.1.3 Connector Schema Data Types

Each connector publishes a set of schema data types for defining the domain schemas of its PIM server. The following restrictions apply to these data types:

- Any new connector domain fields added to a connector domain must only be assigned to a single schema data type published by the corresponding connector.

- The schema data types published by the connector cannot be modified, restricted, or extended. Do not change them.

8.2.2 Hub Schema Documents

A connector must process the following schemas provided by the Hub:

- Hub Schema Type Library

- Hub Schema Documents

8.2.2.1 Hub Schema Type Library

The Hub defines a library of custom XML schema types in the `HubTypeLibrary.xsd` document (Example 8–3). These schema types are available for use by any Hub or connector domain schema.

Example 8–3   Custom XML Schema Types

```
<?xml version="1.0" encoding="windows-1252" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/pss/hub/TypeLibrary"
            elementFormDefault="qualified">
  <xsd:simpleType name="HubPercent">
    <xsd:restriction base="xsd:positiveInteger">
      <xsd:maxInclusive value="100"/>
      <xsd:minInclusive value="0"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="HubUnsignedLong">
    <xsd:annotation>
      <xsd:documentation>Specifies an amount of work in terms of
minutes.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:unsignedLong"/>
  </xsd:simpleType>
  <xsd:complexType name="HubTokens">
    <xsd:sequence>
      <xsd:element name="Token" maxOccurs="unbounded" nillable="false"
                   minOccurs="0" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:simpleType name="HubUtcOffset">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value='[+|-][0-9]{2}:[0-9]{2}'/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

8.2.2.2 Hub Task Schema

The Hub defines an XML schema as the central data format for passing task record data between the Hub and the connectors. The schema is defined in the `HubTask.xsd` document (Example 8–4).

Example 8–4   Hub Task XML Schema Listing

```
<?xml version="1.0" encoding="windows-1252" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/pss/hub/TaskDomain"
            targetNamespace="http://xmlns.oracle.com/pss/hub/TaskDomain"
            elementFormDefault="qualified">
  <xsd:include schemaLocation="HubTypeLibrary.xsd"/>
  <xsd:simpleType name="HubTaskPriority">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="High"/>
      <xsd:enumeration value="MediumHigh"/>
      <xsd:enumeration value="Medium"/>
      <xsd:enumeration value="MediumLow"/>
      <xsd:enumeration value="Low"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="HubTaskStatus">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="NotStarted"/>
      <xsd:enumeration value="InProgress"/>
      <xsd:enumeration value="Completed"/>
      <xsd:enumeration value="WaitingOnSomeoneElse"/>
      <xsd:enumeration value="Deferred"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="HubTaskDomain">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ActualWork" type="xsd:unsignedLong" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="BillingInfo" type="xsd:string" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="Category" type="HubTokens" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="Comments" type="xsd:string" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="Companies" type="HubTokens" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="CompletedFlag" type="xsd:boolean" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="DateCompleted" type="xsd:dateTime" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="DateCompletedUtcOffset" type="HubUtcOffset"
minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="DueDate" type="xsd:dateTime" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="DueDateUtcOffset" type="HubUtcOffset" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="Mileage" type="xsd:string" minOccurs="0"
                     maxOccurs="1" nillable="true"/>
        <xsd:element name="PercentComplete" type="HubPercent"/>
        <xsd:element name="Priority" type="HubTaskPriority"/>
        <xsd:element name="Private" type="xsd:boolean"/>
```

```
                <xsd:element name="Reminder" type="xsd:boolean" minOccurs="0"
                        maxOccurs="1" nillable="true"/>
                <xsd:element name="ReminderOffset" type="xsd:dateTime" minOccurs="0"
                        maxOccurs="1" nillable="true"/>
                <xsd:element name="ReminderOffsetUtcOffset" type="HubUtcOffset"
minOccurs="0"
                        maxOccurs="1" nillable="true"/>
                <xsd:element name="StartDate" type="xsd:dateTime"/>
                <xsd:element name="StartDateUtcOffset" type="HubUtcOffset" minOccurs="0"
                        maxOccurs="1" nillable="true"/>
                <xsd:element name="Status" type="HubTaskStatus"/>
                <xsd:element name="Subject" type="xsd:string"  minOccurs="0"
                        maxOccurs="1" nillable="true"/>
                <xsd:element name="TotalWork" type="HubUnsignedLong" minOccurs="0"
                        maxOccurs="1" nillable="true"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

8.3 Data Side Effects Caused by Synchronization

Data loss can occur because of:

- Data Model Incompatibility
- Differing Data Type Facets
- List Transformations
- Record Collisions

8.3.1 Data Model Incompatibility

Data loss may occur when data models for Hub and PIM XSD schemas cannot represent the same set of data. For example, if a Hub Task domain field (field-X) is not represented in Task domain schema for PIM server B and a Hub Task record extracted from PIM server A contains a value for Hub field-X, then synchronization still propagates the Task record from PIM server A to PIM Server B. However, the value of Hub field-X is not propagated to PIM server B. A subsequent update of the Task record from PIM server B to PIM Server A causes the data for field-X to be removed from PIM server A. This loss occurs because the data is not available from the updated record coming from PIM server B.

8.3.2 Differing Data Type Facets

Data loss can result as transformations occur between different data types with different constraining facets. Some data type facets that may result in data loss include:

- Length
- Pattern
- Enumeration
- White Space

8.3.2.1 Length

Data loss may occur when field length differs between the Hub and the connector domains for the same corresponding field. For example, if PIM Server A can store more task comment text than PIM server B, then the synchronization of task records causes the comment text to be truncated to the length allowed by PIM server B.

8.3.2.2 Pattern

Data may change unexpectedly or content may be lost if any PIM server or connector imposes a data pattern or data value restriction that changes the data value content. For example, suppose the connector for PIM server A extracts a PIM contact record with customer text notes appended to a phone number. Although synchronization still propagates the record to PIM server B, the business logic for PIM Server B strips the text notes as the record is written to PIM server B. A subsequent update for this record from PIM server B results in the removal of the text notes from the contact record in PIM server A.

8.3.2.3 Enumeration

Enumeration granularity is reduced to the most coarse-grained level available from any PIM server or Hub domain. For example, if PIM server A has task priority levels of High and Low and PIM server B has task priority levels of High, Medium and Low, then the granularity of all task priority levels is reduced to the lower granularity of PIM server A, as synchronization occurs between the two PIM servers.

8.3.2.4 White Space

Extraneous white space may be truncated during XSLT transformation by a connector or during synchronization if it is truncated by the business logic of the PIM server. For example, suppose PIM server A trims white space, but PIM server B does not. As synchronizations occur between the two PIM servers, the white space in PIM server B is trimmed because of updates coming from PIM server A.

8.3.3 List Transformations

The categories of list transformations that can result in data loss include:

- Multivalue Field
- Field Group

In general, data loss may occur if a list or set of values from a source domain cannot be represented in a target domain.

8.3.3.1 Multivalue Field

A multivalue field is a single field that contains a list of tokens (that is, a list of values). The Exchange category field is an example of this data type. Data loss occurs if the entire list of values from the source domain field cannot be represented in the target domain. For example, if a task category field from Microsoft Exchange has multiple value tokens but the Hub task domain can only represent a single value for the corresponding category field, then transformation to the Hub schema format strips all but one token value.

8.3.3.2 Field Group

A field group is a collection or group of related fields. Information loss may occur if the number of fields in the group differs between the Hub and PIM XSD schemas. For example, PIM server A allows three address lines, but PIM server B allows four

address lines. Although this does not necessarily mean that data loss can occur, special transformation processing may be necessary to prevent data loss.

8.3.4 Record Collisions

The types of record collisions that result in data loss are as follows:

- Engine Merge Collision: The collisions that occur when the Engine merges changes from multiple records can cause the loss of data for the record that does not win the collision merge competition.

- Connector PIM Collision: The collisions that occur when a connector writes changes to a PIM Server that conflict with new PIM record changes can cause data loss. At some point, a change is overwritten, resulting in a loss of data.

9

Synchronizing the Calendar Domain

This chapter describes how Business Data Synchronization Server (BDSS) synchronizes Calendar domain records.

This chapter includes the following topics:

- Overview of Calendar Synchronization Support
- Transforming Custom Calendar Fields
- Supporting Calendar Synchronization in Hub Schema
- Creating Custom Calendar Fields
- Data Side Effects Caused by Synchronizing ICAL Fields

9.1 Overview of Calendar Synchronization Support

With the exception of the record data format- and attendee-related processing, the Calendar domain synchronizes in the same way as any other domains. However, the record data format for the Calendar domain is different than the record data format of other domains. That is, the other domains represent a domain record as a XML document where each field of the record is represented as a node in the XML document. For the Hub Calendar domain, a record is still a XML document, but instead of each field being a separate node in the document, there is a single node whose value is a iCalendar (ICAL) message (see RFC 2445: Internet Calendaring and Scheduling Core Object Specification, located at `http://www.ietf.org/rfc/rfc2445.txt`).

The embedded ICalendar message contains the individual field names and values.

> **Note:** While ICAL is the widely accepted standard for exchanging calendar data, it is not defined in terms of XML and, to date, there are no standards that map ICAL-to-XML (see the IETF draft for XCAL at `http://www.ietf.org/old/2009/proceedings/02mar/I-D/draft-ietf-calsch-many-xcal-01.txt`).

9.1.1 Transforming Calendar Records

The Hub Calendar domain has only one field, ICalBody. The value for ICalBody is `VCALENDAR` content that conforms to RFC 2445. The Hub-to-PIM and PIM-to-Hub XSL files map this field.

Example 9–1   Hub XML

```
<?xml version="1.0" encoding="windows-1252" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://xmlns.oracle.com/bdss/hub/CalendarDomain"
targetNamespace="http://xmlns.oracle.com/bdss/hub/CalendarDomain"
elementFormDefault="qualified">
<xsd:element name="HubCalendarDomain">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="ICalBody" type="xsd:string" minOccurs="1"
maxOccurs="1" nillable="false"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

> **Note:** This document refers to an XML document containing a field
> whose value is an ICAL message as *Embedded-ICAL*.

As with all other domains, it is a connector implementation detail regarding what the XSL transformation produces when translating between the Hub and PIM XML forms if the transformation satisfies the two XSDs.

When transforming a Hub XML record to PIM XML record, connectors perform a Embedded-ICAL-to-XML or a Embedded-ICAL-to-Embedded-ICAL transformation. Similarly, when transforming a PIM XML record to Hub XML record, connectors most likely perform an XML-to-Embedded-ICAL or an Embedded-ICAL-to-Embedded-ICAL transformation. For example, when transforming XML from Hub to PIM format, one connector's XSL implementation may transform the Hub ICalBody field to the XML-equivalent form of ICAL while another connector may keep the ICalBody as an ICAL message. In both cases, the resulting PIM XML must be parsed and converted to a form usable by the PIM API when creating or updating a record in the targeted PIM. For connectors that transform the Hub ICalBody field to the XML-equivalent form of ICAL, the PIM API conversion routines obtain the ICAL from the transformed XML, parse the ICAL, and build PIM API constructs. For connectors that retain the ICalBody as an ICAL message, the PIM API conversion parses the ICAL-like XML and build the PIM API constructs.

9.2 Transforming Custom Calendar Fields

It is a connector implementation detail regarding how to perform field mapping customization, or if the connector implementation even allows customization. As described in Chapter 8, "Mapping Connector Fields to Hub Fields," connectors use XSD and XSLT to translate between Hub and PIM XML representations of a record. This is also true for the Calendar domain.

Because the calendar data flows through the system as an XML document with Embedded-ICAL, customizing field mappings between the Hub and PIM Calendar pose mappings that are more complicated than those described in Chapter 8, "Mapping Connector Fields to Hub Fields").

If a connectors' XSLT can transform ICAL-to-XML to XML-to-ICAL, then you can modify the XSL to customize the field mapping. This XSLT field mapping may be more complex than other domain XSLTs, however.

9.2.1 Transformations Through Secondary Translation

If a connectors' XSLT does not transform Embedded-ICAL-to-XML or XML-to-Embedded-ICAL, a connector implementation might provide a mapping customization capability by defining a separate set of PIM XSD and XSLT files that represent a template of a PIM calendar record (called a template in this document because it represents a calendar record containing a subset of fields but no recurrence properties). One XSD could contain a subset of the ICAL field names (that is, the fields of a single VEVENT and perhaps not all fields are listed in the XSD) that would be mappable. The other XSD would contain the PIM field names. As the connector parses and processes a given VEVENT form the ICAL message provided by the Hub, it can build a XML message for the VEVENT using the ICAL data and then use the XSL to transform the data and thereby get the ability to customize the mappings. The connector can then perform the PIM API translation using the resulting XML. This document refers to this method as the secondary-translation method.

Figure 9–1 is a graphical representation of the XSD and XSLT. As the connector processes an ICAL VEVENT, it builds an XML message conforming to exchangevirtualcalendar.xsd. It would then use the XSL to transform the XML to one that conforms to the exchange2007calendar.xsd. The customization depicted in Figure 9–1 is that the LOCATION is not mapped for synchronization.

Figure 9–1   Customized Field Mapping
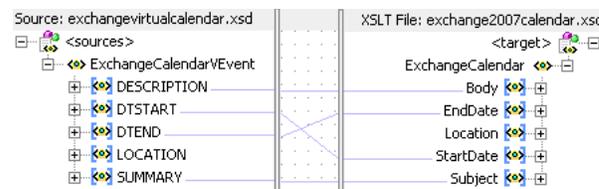
9.2.1.1 Enabling Custom Fields for the Exchange 2007 Connector

You can add custom fields to a set of XSDs that are not exposed to the Hub using the secondary translation method. To enable a connector to synchronize with the Hub, you must create the following XSD files:

- An XSD file that defines the field definitions for a connector

- An XSD file that defines the ICAL counterparts of the field definitions

- An XSLT file that transforms the connector's field definitions into ICAL definitions

- An XSLT file that transforms the ICAL field definitions into the connector's definitions

These files are considered secondary files because they are never exposed to the Hub. For the Exchange 2007 connector, you can enable the synchronization of custom calendar fields by updating the secondary translation XSD and XSLT files. For more information, see Appendix D, "Custom Field XSLTs and XSDs."

9.2.2 Translations Through Java-Based Connectors

Java-based connectors that leverage the BDSS Generic Connector components could use the Transformer API to perform this transformation. This translation is transparent to the Hub and the BDSS Generic Connector HubTransport component.

> **Note:** The advantages to synchronization using the secondary translation method include:
>
> - A simpler XSLT to maintain. The XSLT would be similar to that of other domains.
>
> - You do not have to know the specifics of ICAL
>
> - Customizing field mappings is simpler

9.3 Supporting Calendar Synchronization in Hub Schema

Synchronization support in the Hub schema is restricted to the `VEVENT`, `VTIMEZONE`, `VALARM` calendar components.

> **Note:** In Oracle Fusion Middleware 11*g* release 1, BDSS does not support calendar synchronization for the `VTODO`, `VJOURNAL`, and `VFREEBUSY` components.

The Hub also requires the following:

- The `METHOD` property must always be `REQUEST`.

- A single ICAL message must represent a PIM calendar record. If the calendar record contains a recurrence pattern that has exceptions, the ICAL message must contain multiple `VEVENT` components, such that the first `VEVENT` component of the ICAL message is the **master record** of the pattern and each subsequent `VEVENT` component represents an exception.

- All ICAL messages that describe a meeting must have the same meeting ID. Identical meetings may present a problem on extraction if records from different system key fields match each other. If they match, then the records have different meeting IDs. When the Hub matches the records based on the configured key fields, it treats the match as a Create/Create Conflict and pushes an update to the loosing connector. The losing connector then performs an update operation to ensure that its record data matches the winning connectors' record.

  Updating the `MeetingId` in the target system may present problems to the calendaring system. Connectors should be careful when changing the `MeetingId`. For example, Exchange uses the `MeetingId` to correlate meeting update and meeting invitation e-mails across all attendees of the meeting. If the Exchange connector modifies the `MeetingId`, it breaks the Exchange normal meeting processing and the e-mails would no longer function the way they should (that is, when attendee reads a meeting invitation and accepts it, it creates a calendar record. When an attendee reads a meeting update e-mail, the corresponding calendar record is updated, and so on).

9.4 Creating Custom Calendar Fields

As described in RFC 2445, Section 4.8.8.1: Non-Standard Properties, the ICAL standard allows for custom fields in the form of properties whose name begins with `X-`. For example, `X-ORACLE-BDSS-EXCHANGE-F1` and `X-ORACLE-BDSS-BPEL-F1` are examples of valid custom field names. Further, custom fields and custom field values may differ for each exception to a recurring calendar record. To enable successful synchronization of the Calendar domain across all of the connectors that synchronize the Calendar domain, developers must implement connectors to allow custom fields

through secondary translation or by including ICAL and custom fields in the Hub calendar definition.

If a connector opts to use the secondary-translation mechanism, you can define the custom fields in the PIM-specific XSDs and define the mappings. The custom fields remain in the ICAL message provided by the Hub.

If a connector opts not to use the secondary-translation option, you can define a Hub data type in the Hub type library XSD, and define the custom Hub and PIM fields to be of this type in the respective XSDs. You can then map them in the PIM-To-Hub and Hub-To-PIM XSLT files. The connectors would place custom field data outside of the ICalBody. The data type definition must allow multiple values so that connectors can specify different custom properties and values for each exception (perhaps something similar to the HubTokens data type). With this approach, there is a danger of data loss because it would be possible for a ICAL custom field in a source connector to be mapped to a regular ICAL field on the target connector. For more information, see Section 9.5.2, "Mapping a Custom ICAL Field to a Regular ICAL Field."

9.5 Data Side Effects Caused by Synchronizing ICAL Fields

Data loss can occur because of:

- Failed Synchronization of Any ICAL Field
- Mapping a Custom ICAL Field to a Regular ICAL Field

9.5.1 Failed Synchronization of Any ICAL Field

As mentioned in Section 9.2, "Transforming Custom Calendar Fields," there is the danger of data loss if any ICAL field is not synchronized. Therefore, connectors should attempt to synchronize every ICAL field using hidden custom fields if the target calendar system does not support a given field.

If the connectors' PIM Server does not support hidden custom fields, a connector implementation must devise a method to persist and retrieve the field data. Data loss can occur even with this approach. In the case of an XSL that converts Embedded-ICAL to XML, if the XSL omits a ICAL field then data loss can occur because the target connector does not write a value to the field. If the connector uses the secondary-translation approach, the same problem occurs if a mapping of a ICAL field is not made. In both cases, this can be remedied by ensuring each ICAL field is mapped. A connector may appear as though a field is not synchronizing if it synchronizes the field to a custom PIM field that the client applications do not display by default. If the connectors' system does not support custom hidden fields, then the connector must devise a method to persist the field data so it can be provided back to the Hub during extractions.

If a connector cannot translate Embedded-ICAL to XML, or opts not to do the secondary-translation method, it may not support field mapping customization and rely on a **natural map** of ICAL field names to PIM field names. In this case, there would be no support for field definition or mapping customizations.

9.5.2 Mapping a Custom ICAL Field to a Regular ICAL Field

As noted in Section 9.4, "Creating Custom Calendar Fields," a connector that places field data outside of the ICalBody may result in an ICAL custom field in a source connector to be mapped to a regular ICAL field on the target connector. For example, suppose the BPEL calendar has a field called HTTP Link that, when clicked, navigates the BPEL user to a page in the BPEL application that provides information associated

with the calendar record. Further, assume that the administrator wants to synchronize this field to a field in Exchange. Since there is no ICAL field that can represent this field, define a custom ICAL field, such as `X-ORACLE-BDSS-BPEL-HTTP-LINK`. You then modify the Hub XSD to have a ICAL X- field, such as `X-ORACLE-BDSS-HUB-HTTP-LINK`, and then map the two fields in the Hub-To-PIM and PIM-To-Hub XSLs used by the BPEL Task Connector. To synchronize the field to Exchange, define an Exchange custom ICAL field, such as `ORACLE-BDSS-EXCHANGE-HTTP-LINK` and map it to the `ORACLE-BDSS-HUB-HTTP-LINK` field in the Hub-to-PIM and PIM-to-Hub XSL used by the Exchange connector.

When defining the Exchange custom field in the Exchange XSD, you can define it as an extended field or as a non-indexed field. For the HTTP Link to the `LOCATION` field on the Exchange calendar, define this field as the location of the non-indexed field. Because this non-indexed field maps to the `ICAL LOCATION` field, the connector uses the `ORACLE-BDSS-EXCHANGE-HTTP-LINK` value instead of the `ICAL LOCATION` value to update the `LOCATION` field, the BPEL Task Connector overwrites the field during subsequent synchronization sessions to which it maps the `ICAL LOCATION` field with the link if the change comes from Exchange. If the connector uses the `ICAL LOCATION` field and ignores the `ORACLE-BDSS-EXCHANGE-HTTP-LINK` value to update the Location field, then a subsequent session results in the BPEL Task Connector deleting the HTTP Link field value if the change originates from Exchange.

If you defined the field as an extended field that specifies a custom field (that is, it would remain hidden unless you define an Outlook custom form that would expose the field in the Outlook UI), then this form of data loss would not occur.

> **Note:** In this example, the BPEL Task Connector would not hard-code this customization by specifying the HTTP Link value as the LOCATION field value when it builds the ICAL that is sent to the Hub.

Part III

Appendixes

This part contains the following appendixes and a glossary:

- Appendix A, "BDSS Configuration Operations"
- Appendix B, "Tables in the BDSS Data Stores"
- Appendix C, "Connector API"
- Appendix D, "Custom Field XSLTs and XSDs"
- Appendix E, "Troubleshooting BDSS"
- Glossary

A

BDSS Configuration Operations

This appendix describes how Business Data Synchronization Server (BDSS) synchronizes Calendar domain records.

This appendix includes the following topics:

- UserConfigMBean
- ServerDomainConfigMBean
- DispatcherSettings MBean
- EngineSettings MBean
- ProfileRuntimeInfo MBean
- Profile Parameters

A.1 UserConfigMBean

Section A–1 lists the operations of the UserConfigMBean. For more information, see Section 4.7, "Managing Users."

Table A–1   Operations of the User Configuration MBean

Operation	Description
showUserList	Returns a list of all users and their status
showFailinguserList	Returns a list of all users that are currently failing to synchronize
showUserStatus	Displays the status for a user, including record level status. There is an optional flag that forces the operation to only return failing items.
addConnectorUser	Adds a new connector user and associates the new user with a **Hub user**. This operation creates a Hub user if one does not exist. For a user who is configured to synchronize, this operation must be performed for each connector. The Hub user must then be enabled for synchronization using updateUserSyncEnablementFlag.
removeHubUser	Removes a Hub user from the database and also all of the connector users associated with that Hub user

Table A–1   (Cont.) Operations of the User Configuration MBean

Operation	Description
updateHubUserSyncEnableFlag	Updates the synchronization-enabled flag for a user. T or True means that the user synchronizes in all of the subsequent synchronization sessions. F or False means the user does not synchronize.
updateUserDomainSyncLevel	Changes the synchronization level for a connector user on the given domain. The valid synchronization levels are currently Full, None, Inbound Only and Outbound Only.

A.2 ServerDomainConfigMBean

Table A–2 lists the operations of the ServerDomainConfigMBean. For more information see Section 4.2, "Managing Connectors" and Section 4.4, "Managing Connector Instances."

Table A–2   Operations of the ServerDomainConfigMBean

Operation	Description
showConnectorDetails	Returns configuration for specified connectors. If the connector name is empty, then this operation returns configuration information for all configured connectors.
updateConnectorForPim	Adds or modifies a connector
deleteConnector	Removes the given connector
updatePimServerInstance	Updates an existing PIM server instance or creates a new PIM server instance
deletePimServerInstance	Deletes a PIM server instance
updateConnectorInstance	Creates or modifies a connector instance for a connector
deleteConnectorInstance	Deletes a connector instance from a connector
updateDomainForConnector	Creates or modifies a domain for a connector
deleteDomainForConnector	Deletes a domain from a connector
updateFilterForDomain	Creates or modifies a filter for a connector domain
deleteFilterForDomain	Deletes a filter from a domain

A.3 DispatcherSettings MBean

Table A–3 lists the attributes of the DispatcherSettings MBean. For more information, see Section 5.2, "Configuring the Dispatcher."

Table A–3   Attributes of the DispatcherSettings MBean

Attribute	Description
ChunkSize	The number of users contained in the message sent to the Hub for synchronization. Use this attribute to balance the load to each instance of a running Engine. The default value is 20 users per message.

Table A–3 (Cont.) Attributes of the DispatcherSettings MBean

Attribute	Description
HubEndPointURL	The URL used by the Dispatcher to connect to the DispatcherHub Web service on the server hosting the Hub. For example, a typical Hub URL for the WebLogic Server is `http://<HUB_SERVER_IP_ADDRESS>:7001/BDSSHubWebServices/DispatcherHubWebService?wsdl`.

A.4 EngineSettings MBean

Table A–4 lists the attributes of the EngineSettings MBean. For more information, see Section 5.3, "Configuring the Engine."

Table A–4 Attributes of the EngineSettings MBean

Attributes	Description
EngineEndPointURL	The URL used by the Engine to enable the connectors to extract requests and data update request responses to the Engine. To find this URL using Oracle WebLogic Administration Console: **1.** Select **Deployments**. **2.** Expand the tree structure of BDSSHub. **3.** Select the **EngineCallBackInterface** node. The Settings for EngineCallback page appears. **4.** Select the **Testing** tab. **5.** Expand the **EngineCallbackInterface** node. **6.** Select the `?wsdl` test point.
ExtractResponseTimeOut	The number of milliseconds that the Engine waits to process an extract response from a connector. The default value is 60000 milliseconds.The timeout value must be sufficiently large to prevent a timeout from occurring under normal circumstances. This value depends on several factors that include: ■ The size of the largest record set that can be extracted. ■ The number of Hub domains that are to be extracted. This has bearing because Hub domains are processed serially by the Engine for a synchronization session. Therefore, Hub domain B must wait longer to retrieve its extract responses if Hub domain A were processed first. ■ Other factors, including system load and communication latency, that affect the ability of a Engine to retrieve and process extract response messages from connectors before the timeout occurs. If a timeout occurs, then synchronization of the corresponding Hub domain is terminated for the affected user synchronization session.
MessageTimeToLive	The number of milliseconds that a message containing an extract response remains on the JMS queue before being removed. The default value is 60000 milliseconds. Calculate this timeout value using an analysis similar to the one performed for the `ExtractResponseTimeOut` attribute.

Table A–4 (Cont.) Attributes of the EngineSettings MBean

Attributes	Description
RuntimeLibraryURL	The URL that the Engine sends to the connectors to enable them to communicate with the Connector Run-Time Library.
	To find this URL using Oracle WebLogic Server Console:
	1. Select **Deployments**.
	2. Expand the tree structure of BDSSHub.
	3. Select the ConnectorRuntimeInterface node. The Settings for ConnectorRuntimeInterface page appears.
	4. Select the **Testing** tab.
	5. Expand the **ConnectorRuntimeInterface** node.
	6. Select the `?wsdl` test point.

A.5 ProfileRuntimeInfo MBean

The ProfileRuntimeInfo MBean provides operations for modifying and viewing profile parameters. For more information, see Section 4.8, "Managing Profiles."

Table A–5 Operations of the ProfileRuntimeInfo MBean

Operation	Description
`displayAllProfiles`	Returns a list of all parameters grouped by their profiles and sections
`displayProfile`	Displays only the parameters for the given profile
`removeProfileParameter`	Removes a given parameter. You must provide a profile name and a section name to ensure that the correct parameter is removed.
`setProfileParameter`	Updates an existing parameter or creates a new parameter

A.6 Profile Parameters

A profile stores the configuration metadata for a connector. Each parameter has a single value. The operations of the ProfileRuntimeInfo MBean (listed in Table A–5) enable you to manage the profiles for a connector. BDSS ships with profiles seeded for the Exchange Connector and the FTSKeyFields Profile. The profiles are listed in Table A–6. For more information, see Chapter 6, "Configuring the BDSS Connector for Microsoft Exchange 2007," and Section 5.5, "Configuring the FtsKeyFields Profile."

Table A–6 The BDSS Profile Parameters

Profile	Section	Parameter	Description
BdssKeyFields	Calendar	KeyField1	First field used for record matching
BdssKeyFields	Calendar	KeyField2	Second field used for record matching
BdssKeyFields	Contact	KeyField1	First field used for record matching
BdssKeyFields	Contact	KeyField2	Second field used for record matching
BdssKeyFields	Task	KeyField1	First field used for record matching

Table A–6   (Cont.) The BDSS Profile Parameters

Profile	Section	Parameter	Description
BdssKeyFields	Task	KeyField2	Second field used for record matching
FTSKeyFields	Tasks	KeyField1	The first field used for record matching
FTSKeyFields	Tasks	KeyField2	The second field used for record matching
Exchange2007	Hub Transport	Pim Transport Implementation Class	The Hub Transport generic component uses this value to initialize the Exchange PIM Transport.
Exchange2007	Hub Transport	Extract Response Batch Size	Set this value to a sufficiently large number so that no user synchronized byBDSS has records that exceeds this value.
Exchange2007	Task	Pim Xsd	The URI to the `exchange2007task.xsd` file.
Exchange2007	Task	PimToHubUrl	The URI to the `Exchange2007TaskToHubTask.xsl`
Exchange2007	Task	HubToPimUrl	The URI to the `HubTaskToExchange2007Task.xsl`.
Exchange2007	Task	Domain Class Type	The Exchange Web Service interface class that is used when synchronizing the domain
Exchange2007	Task	Exchange Domain Batch Size	The maximum number of records downloaded from Exchange when extract records from the domain target. Valid values are 1 through 512.
Exchange2007	Task	Folder Class	The folder class for the domain.
Exchange2007	Task	Body Type	How the Body field for the domain is synchronized. Valid values are `Text` or `HTML`.
Exchange2007	Task	Conflict Resolution Type	Determines how Exchange handles a conflict type. Valid values are `AutoResolve`, `AlwaysOverwrite`, and `NeverOverwrite`.
Exchange2007	Task	Disposal Type	Determines how Exchange handles deleted records. Valid values are `HardDelete`, `SoftDelete`, and `MoveToDeletedItems`.
Exchange2007	Task	Pim Domain id	The PIM domain identifier, used as the message class property for each synchronizing record

Table A–6   (Cont.) The BDSS Profile Parameters

Profile	Section	Parameter	Description
Exchange2007	Contact	Exchange Domain Batch Size	The maximum number of records downloaded from Exchange when extracting records from the domain target. Valid values are 1 to 512, inclusive
Exchange2007	Contact	Folder Class	The folder class for the domain
Exchange2007	Contact	Pim Domain Id	The PIM domain identifier, used as the message class property of each record synchronized
Exchange2007	Contact	Pim Xsd	The URI to the `exchange2007contact.xsd` file
Exchange2007	Contact	PimToHubUrl	The URI to the `Exchange2007ContactToHubContact.xsl`
Exchange2007	Contact	HubToPimUrl	The URI to the `HubContactToExchange2007Contact.xsl`
Exchange2007	Calendar	Time Zone Xsd	The URI to `timezoneregistryschema.xsd`
Exchange2007	Calendar	VEvent Xsd	The URI to `veventasxml.xsd`
Exchange2007	Calendar	PimToHubUrl	The URI to `exchange2007calendartohubcalendar.xsl`
Exchange2007	Calendar	Pim Xsd	The URI to `pimicalbody.xsd`
Exchange2007	Calendar	Calendar Item Type Xsd	The URI to `calendaritemtype.xsd`
Exchange2007	Calendar	Time Zone Xml	The URI to `timezoneregistry.xml`
Exchange2007	Calendar	HubToPimUrl	The URI to `hubcalendartoexchange2007calendar.xsl`
Exchange2007	Calendar	CalendarItemToVEvent Xsl	The URI to `calendaritemtypetoveventasxml.xsl`
Exchange2007	Calendar	VEventToCalendarItem Xsl	The URI to `veventasxmltocalendaritemtype.xsl`
Exchange2007	Calendar	Body Type	How the Body field for the domain is synchronized. Valid values are `Text` or `HTML`.
Exchange2007	Calendar	Conflict Resolution Type	Determines how Exchange handles a conflict type. Valid values are `AutoResolve`, `AlwaysOverwrite`, and `NeverOverwrite`.

Table A–6 (Cont.) The BDSS Profile Parameters

Profile	Section	Parameter	Description
Exchange2007	Calendar	Disposal Type	Determines how Exchange handles deleted records. Valid values are `HardDelete`, `SoftDelete`, and `MoveToDeletedItems`.
Exchange2007	Calendar	Domain Class Type	The Exchange Web Service interface class that is used when synchronizing a domain.
Exchange2007	Calendar	Exchange Domain Batch Size	The maximum number of records downloaded from Exchange when extracting records from the domain target. Valid values are 1 to 512, inclusive.
Exchange2007	Calendar	Fanning Limit	The number of records created when a recurring calendar record cannot be natively represented in Exchange. Valid values are 1 to 100, inclusive.
Exchange2007	Calendar	Folder Class	The folder class for the domain
Exchange2007	Calendar	Pim Domain id	The PIM domain identifier, used as the message class property of each record synchronized.
Exchange2007	Active Directory	ServerN (where N is a ordinal >= 0)	The LDAP URL used by the connector to obtain Service Connection Point URLs from Active Directory when discovering a user's Exchange account information. You can add multiple entries. The ordinal specifies a preference order (that is, Server0 is tried before Server1, and so on). The connector tries each URL until the user is discovered or each entry has been tried.
Exchange2007	Active Directory	Referral	Specifies how the LDAP provider handles referrals. Values are *follow*, *throw* or *defer*. The default value is *follow*.
Exchange2007	Active Directory	Max Query Result Size	Max number of results to return per Active Directory query. The default value is 20.
Exchange2007	Active Directory	Max Query Time	The time (in milliseconds) to wait for an Active Directory query to return. The default value is 30 seconds (30000 milliseconds).
Exchange2007	Active Directory	Read Timeout	The time (in milliseconds) to wait for a response before timing out. The default value is 5000.

Table A–6   (Cont.) The BDSS Profile Parameters

Profile	Section	Parameter	Description
Exchange2007	Active Directory	Connect Timeout	The time (in milliseconds) to establish a connection to Active Directory before timing out. The default value is 5000.
Exchange2007	Active Directory	Use SSL	Whether secure LDAP is used. The default is `true`. However, in the development environment, set the value to `false` because the Domain Controller is not enabled for secure LDAP
Exchange2007	Active Directory	Using Client Certificates	Indicates if the client provider uses LDAP certificates
Exchange2007	Active Directory	Client Certificate Path	Set to `true` if Use SSL is set to true and the Domain Controller requires a client certificate. The default value is `false`.
Exchange2007	Auto Discovery	SCP*N* where *N* is an ordinal >= 0	The URL to a Service Connection Point (SCP) that the connector should use when discovering a user. The ordinal specifies a preference order (the connector tries *SCP0* is tried before *SCP1*, and so on). The connector tries each URL until the user is discovered or each entry has been tried.
Exchange2007	Auto Discovery	Remove Domain From Urls	Directs the connector to remove the domain information from a discovered SCP URLs obtained from Active Directory. For the development environment, set the value to *true* because DNS prevents the URL from being resolved when the URL contains a domain (that is, `http://server1.us.example.com/autodiscover/autodiscover.xml` does not work.) When the option set to true, the URL becomes `http://server1/autodiscover/autodiscover.xml`.
Exchange2007	Auto Discovery	Read Timeout	The time (in milliseconds) to wait for a response before timing out. The default value is 5000.
Exchange2007	Auto Discovery	Connect Timeout	The time (in milliseconds) to establish a connection to Auto Discover server before timing out. The default is 5000.

Table A–6   (Cont.) The BDSS Profile Parameters

Profile	Section	Parameter	Description
Exchange2007	Auto Discovery	Use SSL	Whether secure HTTP (HTTPS) is used. The default is *true*. HTTPS may be required for the Auto Discover service if attempts to configure the server to allow HTTP fail.
Exchange2007	Auto Discovery	Client Certificate Path	Set this value to the location of the PEM (Privacy Enhance Mail security certificate) file containing the client certificate that is used if Auto Discover requires client certificates
Exchange2007	Auto Discovery	Client Certificate Key Path	Set this value to the location of the PEM (Privacy Enhance Mail security certificate) file containing the client certificate private key
Exchange2007	Auto Discovery	Port	Defaults to 443, the typical HTTPS secure port. Set the value to the HTTPS port if Use SSL is set to true. If false, this value should be the HTTP port. The value is only used when attempting default SCP URLs (which are attempted when all SCP URLS have been tried (those in the SCPN configuration and those obtained from AD per the Server$n$ configuration).
Exchange2007	Event Receiver	CHANGE_ME	The name of an Exchange 2007 connector as it appears in the NAME Column of the CONNECTORS table.
Exchange2007	Event Receiver	Status Frequency	The frequency, in minutes, that Exchange provides a notification for each subscription
Exchange2007	Event Receiver	Subscription Status Factor	The event receiver performs a check on all subscriptions to verify that the Exchange server has not terminated any subscriptions. This parameter sets frequency in which this check occurs. The frequency in which this check occurs is the value set for this parameter multiplied by the value set for the Status Frequency parameter. For example, if you set the Status Frequency parameter to 1 minute, and then set the Subscription Status Factory to 3, then the event receiver performs the subscription check every 3 minutes.

B

Tables in the BDSS Data Stores

This appendix lists the tables that store data for Business Data Synchronization Server (BDSS) components.

This appendix includes the following topics:

- Overview of BDSS Tables
- PIM_SERVER_INSTANCES
- CONNECTORS
- CONN_INSTANCES
- CONNS_TO_PIM_SERVERS
- HUB_DOMAINS
- CONN_DOMAINS
- CONN_DOMAIN_FILTERS
- DESC_FIELDS
- HUB_USERS
- CONN_USERS
- USER_CONTEXTS
- USER_CONTEXT_STATES
- STATE_DATA
- USER_HISTORY
- CONN_USER_RECORDS
- HUB_RECORDS
- CONN_RECORD_DESCS
- ASSOC_DATA
- LOVS
- PROFILES
- SEQUENCE

B.1 Overview of BDSS Tables

This appendix lists the tables that store data for such BDSS-related components as PIM servers, connectors, domains, and filters.

B.2 PIM_SERVER_INSTANCES

The PIM_SERVER_INSTANCES table (Table B–1) represents an instance of a PIM server, not a type of PIM Server.

Table B–1   PIM_SERVER_INSTANCES Table

Column	Type
CONNECTOR_ID	Number
NAME	VARCHAR2[100]

Unique Constraints

PIM_SERVER_INSTANCES_UK1 on CONNECTOR_ID and NAME

B.3 CONNECTORS

The CONNECTORS table (Table B–2) stores data that represents a type of connector grouping. This table includes the SUPPORTS_EVENTS_FLG and USER_EVENT_FLG columns, which enable the **Hub** to synchronize only users with changes, rather than synchronizing all users regardless of whether the users have changed data or not. The SUPPORTS_EVENTS_FLG enables a connector to determine that a change has been made to a user's data and can send a message to the Hub telling it to synchronize the user. The USER_EVENT_FLG enables the Hub to synchronize the user immediately or during the next synchronization cycle.

Table B–2   CONNECTORS Table

Column	Type	Comment
NAME	VARCHAR2[100]	The name of the connector. This name, which must be unique across BDSS, is used (along with a connector user ID) to uniquely identify a connector user.
SERVER_TYPE_CD	Number	The foreign key to the LOVS table for a PIM server type
MAX_SYNC_LVL_CD	Number	The foreign Key to the LOVS table to a synchronization level (SYNC_LVL) synchronization level
ENABLED_FLG	VARCHAR2[1]	A flag that determines if this connector is enabled
SYNC_EVENTS_ ENABLED_FLG	VARCHAR2[1]	A flag that determines if this connector supports user synchronization notification events
SUPPORTS_EVENTS_ FLG	VARCHAR2[1]	A flag that determines if the Hub synchronizes INSERT/UPDATE/DELETE events for a connector user. Setting this flag to T (true) notifies the Hub that it should synchronize a user's data when it has been changed. When this flag is set to T, the Hub synchronizes the user data independent of a **synchronization session** for the user.
USER_EVENT_FLG	VARCHAR2[1]	A flag that determines if the connector sends notifications to the Hub when an INSERT/UPDATE/DELETE event occurs for a connector user. The Hub uses this flag to initiate a synchronization session for the user.

Unique Constraints

CONNECTORS_UK1 on NAME

B.4 CONN_INSTANCES

The CONN_INSTANCES table (Table B–3) stores the values for a **connector instance**.

Table B–3   CONN_INSTANCES Table

Column	Type	Comment
NAME	VARCHAR2[100]	The name of the connector instance. It must be unique across BDSS
SERVICE_URI	VARCHAR2[100]	The URI (Universal Resource Identifier) used by the Hub components to communicate to the connector instance
CONNECTOR_ID	Number	The foreign key to the LOVS table for a SYNC_LVL (synchronization level)
ACTIVE_FLG	VARCHAR2[1]	A flag to determine if this connector instance is active. (This flag should only be set from an MBean management console.)

B.5 CONNS_TO_PIM_SERVERS

The CONNS_TO_PIM_SERVERS table (Table B–4) intersects the CONN_INSTANCES table and the PIM_SERVER_INSTANCES table.

Table B–4   CONNS_TO_PIM_SERVERS Table

Column	Type	Comment
CONN_INSTANCE_ID	Number	The foreign key to the CONN_INSTANCES table
PIM_SERVER_INSTANCE_ID	Number	The foreign key to PIM_SERVER_INSTANCES table

B.6 HUB_DOMAINS

The HUB_DOMAINS table (Table B–5) stores the data for a **Hub domain**.

Table B–5   HUB_DOMAINS Table

Column	Type	Comment
NAME	VARCHAR2[100]	The name of the domain. This should be unique in this table.
PRIORITY	Number	The priority with which this domain should be synchronized. Zero (0) is the highest priority. This should be unique in this table.
MAX_SYNC_LVL_CD	Number	The foreign key to the LOVS table for a SYNC_LVL (synchronization level).
DOMAIN_CLASS_CD	Number	The foreign key to the LOVS table for a DOMAIN_CLASS (such as task).

Unique Constraints

HUB_DOMAINS_UK1 on NAME

HUB_DOMAINS_UK2 on PRIORITY

B.7 CONN_DOMAINS

Table B–6 lists the columns of the CONN_DOMAINS table.

Table B–6   CONN_DOMAINS Table

Column	Type	Comment
DOMAIN_NAME	VARCHAR2[100]	The name of the domain
PRIORITY	Number	The priority at which this domain is synchronized. Zero (0) is the highest priority.
MAX_SYNC_LVL_CD	Number	The foreign key to the LOVS table for a SYNC_LVL (synchronization level)
DOMAIN_CLASS_CD	Number	The foreign key to the LOVS table for a DOMAIN_CLASS (such as task)
CONNECTOR_ID	Number	The foreign key to the CONNECTORS table
HUB_DOMAIN_ID	Number	The foreign key to the HUB_DOMAINS table
FILTER_FLD_LOV_TYPE	VARCHAR2[50]	The type of LOV (list of values) to use for the filter fields of this connector domain. This is unique for a connector type and for a domain.

Unique Constraints

CONN_DOMAINS_UK1 on CONNECTOR_ID and HUB_DOMAIN_ID.

B.8 CONN_DOMAIN_FILTERS

Table B–7 lists the columns of the CONN_DOMAIN_FILTERS table.

Table B–7   CONN_DOMAIN_FILTERS Table

Column	Type	Comment
CONN_DOMAIN_ID	Number	The foreign key to the CONN_DOMAINS table
FIELD_VALUE	VARCHAR2[100]	The field value used to filter with this field
FILTER_SEQ	Number	The sequence in which to use this filter value
OPERATOR_LOV_CD	Number	The foreign key to the LOVS table for an OPERATOR
FILTER_NAME	VARCHAR2[100]	The name of the filter

B.9 DESC_FIELDS

The columns of the DESC_FIELDS table (listed in Table B–8) hold the fields that are configured for the descriptions of a Hub record. In log messages, these description fields are used instead of the ID to identify a record. The field descriptions can be flagged for use as key fields for the HUB_DOMAINS table.

Table B–8   DESC_FIELDS Table

Column	Type	Comment
HUB_DOMAIN_ID	Number	The foreign key to the HUB_DOMAINS table
FIELD_NAME	VARCHAR2[100]	The Hub field name used in the description of a log message
KEY_FIELD_FLG	VARCHAR2[1]	A flag that determines if this Hub field should also be used as a key field

B.10 HUB_USERS

The HUB_USERS table (Table B–9) holds the data for a Hub user (a representation of the mapping of CONNECTOR_USERS).

Table B–9   HUB_USERS Table

Column	Type	Comment
NAME	VARCHAR2[100]	The name for the Hub representation of the user. This name must be unique across BDSS.
SESSION_STATE	VARCHAR2[100]	The state of the user (dispatched, synchronization ended) or a session ID
SYNC_ENABLED_FLG	VARCHAR2[1]	A flag to determine if the Hub user should be considered for synchronizing. Zero (0) indicates that the Hub should not synchronize the user.
LAST_SYNC_HIST_ID	Number	The foreign key to the USER_HISTORY table for the history of the last synchronization cycle for this user
LAST_GOOD_SYNC_HIST_ ID	Number	The foreign key to the USER_HISTORY table for the history of the last successful synchronization cycle for this user.
LAST_UPD	Date	The time stamp of the last synchronization cycle for this user.
SYNC_START_TS	timestamp	The timestamp that indicates the start of the current synchronization cycle for a user. This should be set by a database trigger.
		Triggers: BDSS.HUB_USERS_SYNC_ START_TRG – sets the SYNC_START_ TS using database time. This trigger should be fired only when the HUB_ USERS.SESSION_STATE is updated with a valid sync session state (that is, not null, DISPATCHED or SYNC_ ENDED).

Unique Constraints

HUB_USERS_UK1 on NAME

B.11 CONN_USERS

The columns of the CONN_USERS table (listed in Table B–10) hold the data for a **connector user**.

Table B–10   CONN_USERS Table

Column	Type	Comment
USER_ID	VARCHAR2[100]	The user ID for the connector user. For example: pimsi1001@pimsisiebel.com for Microsoft Exchange 2007. This field (along with the CONNECTOR_ID) can uniquely identify a PIM user.
LANG_CD	VARCHAR2[25]	The language used by the user
LAST_SYNC_TS	Date	The time stamp for the last synchronization session in which this user participated
ERROR	VARCHAR2[100]	Any error encountered by the user in the last synchronization session
LAST_SYNC_STATUS	VARCHAR2[1]	A value of 0 indicates that the last synchronization finished successfully. A value of 1 indicates that the synchronization did not complete successfully.
LAST_GOOD_SYNC_TS	Date	The time stamp for the last successful synchronization session for this user
SYNC_NOW_FLAG	VARCHAR2[1]	A flag that determines if a user is synchronized when using synchronization events to initiate synchronization
CONNECTOR_ID	Number	The foreign key to the CONNECTORS table
CONNECTOR_INST_ID	Number	The foreign key to the CONN_INSTANCES table
HUB_USER_ID	Number	The foreign key to the HUB_USERS table
SYNC_NOW_FLG	varchar2(1)	Set this flag to 1 to indicate that the user must be synchronized.
SYNC_NOW_TS	timestamp	The timestamp that marks when the SYNC_NOW_FLG was last set. The new trigger must to update SYNC_NOW_TS. BDSS.CONN_USERS_SYNC_NOW_TRG sets the SYNC_NOW_TS using the time of the database. This trigger should be fired only when the SYNC_NOW _FLG is updated to 1.

Unique Constraints

CONN_USERS_UK1 on CONNECTOR_ID and USER_ID

CONN_USERS_UK2 on HUB_USER_ID and CONNECTOR_ID

B.12 USER_CONTEXTS

Table B–11 lists the columns of the USER_CONTEXTS table. Each record in this table represents a connector domain for a user.

Table B–11 USER CONTEXTS Table

Column	Type	Comment
CONTEXT_NAME	VARCHAR2[500]	The version of the PIM record
HUB_USER_ID	Number	The foreign key to the HUB_USERS table
CONN_DOMAIN_ID	Number	The foreign key to the CONN_DOMAINS table
TARGET	VARCHAR2[100]	The target folder used for synchronizing records
SYNC_LVL_CD	Number	The foreign key to the LOVS table to a SYNC_LVL (synchronization level)
LAST_SYNC_HIST_ID	Number	The foreign key to the USER_HISTORY table
LAST_GOOD_SYNC_TS	Date	The timestamp for the last successful synchronization session for this user
PREV_SYNC_LVL_CD	Number	The foreign key to the LOVS table for a SYNC_LVL (synchronization level) to the previous SYNC_LVL
SYNC_NOW_FLG	varchar2(1)	Set this flag to 1 to indicate that the user must be synchronized.
SYNC_NOW_TS	timestamp	The timestamp that marks when the `SYNC_NOW_FLG` was last set. The new trigger must to update `SYNC_NOW_TS`. `BDSS.CONN_USERS_SYNC_NOW_TRG` sets the `SYNC_NOW_TS` using the time of the database. This trigger should be fired only when the `SYNC_NOW _FLG` is updated to 1.

Unique Constraints

USER_CONTEXT_UK1 on HUB_USER_ID and CONN_DOMAIN_ID

B.13 USER_CONTEXT_STATES

The USER_CONTEXT_STATES table (Table B–12) enables connectors to store synchronization state data for users that can be used by the connector on the next synchronization cycle to find record changes.

Table B–12 USER_CONTEXT_STATES Table

Column	Type	Comment
USER_CONTEXT_ID	Number	The foreign key to the USER_CONTEXTS table
STATE_INDEX	VARCHAR2[100]	Stores the index of the state
LAST_SYNC_TS	Date	The time stamp for the last synchronization session in which this user participated
TEMP_FLG	VARCHAR2[1]	The flag that indicates a temporary synchronization state

B.14 STATE_DATA

Table B–13 lists the columns of the STATE_DATA table.

Table B–13   STATE_DATA Table

Column	Type	Comment
USER_CONTEXT_STATE_ID	Number	The foreign key to the USER_CONTEXT_STATES table
SEQ_NUM	Float	The sequence in which to return the state data
TEMP_FLG	VARCHAR2[1]	The flag that indicates temporary data synchronization
DATA	VARCHAR2[500]	A chunk of data

B.15 USER_HISTORY

The USER_HISTORY table (Table B–14) stores the history data for a HUB_USER or a USER_CONTEXT.

Table B–14   USER_HISTORY Table

Column	Type	Comment
SYNC_TS	Date	The time stamp for the synchronization session
SYNC_STATUS	VARCHAR2[200]	The status of the synchronization record. A value of zero (0) indicates that the last synchronization completed successfully. A value of 1 indicates that the synchronization did not complete successfully.
ERROR	VARCHAR2[200]	An error message
HUB_USER_ID	Number	The foreign key to the HUB_USERS table
USER_CONTEXT_ID	Number	The foreign key to the USER_CONTEXTS table

B.16 CONN_USER_RECORDS

Table B–15 lists the columns of the CONN_USER_RECORDS table. Each record represents a user's record on a specific PIM server.

Table B–15   CONN_USER_RECORDS Table

Column	Type	Comment
VERSION	VARCHAR2[500]	The version of the PIM record
DELETE_FLG	VARCHAR2[1]	A flag that indicates if this record is deleted
USER_CONTEXT_ID	Number	The foreign key to the USER_CONTEXTS table
CONNECTOR_ROW_IDEN	VARCHAR2[225]	The actual record ID for this PIM record on the PIM server
LAST_INBND_TS	Date	The time stamp of the last inbound synchronization for this record

Table B–15   (Cont.) CONN_USER_RECORDS Table

Column	Type	Comment
LAST_INBND_STATUS	VARCHAR2[1]	The status of the synchronization record. A value of 0 (zero) indicates that the synchronization completed successfully. A value of 1 indicates that the synchronization did not complete successfully.
SOURCE_TS	Date	The time stamp when this record was updated from the source PIM server
ERROR	VARCHAR2[200]	Any errors for this record
HUB_RECORD_ID	Number	The foreign key to the HUB_RECORD table
ECHO_PENDING_FLG	VARCHAR2[1]	A flag that indicates if the system expects an echo for this record, meaning that the record was updated because of a change from another PIM server and the returning update should be dropped.

Unique Constraints

CONN_USER_RECORDS_UK1 on USER_CONTEXT_ID, CONNECTOR_ROW_IDEN, and HUB_RECORD_ID.

B.17 HUB_RECORDS

The HUB_RECORDS table (Table B–16) stores the **Hub record** data.

Table B–16   HUB_RECORDS Table

Column	Type	Comment
HUB_DOMAIN_ID	Number	The foreign key to the HUB_DOMAINS table
LAST_OUTBND_TS	Date	The time stamp for the last outbound synchronization of this record
LAST_OUTBND_STATUS	VARCHAR2[1]	The status of the synchronization record. A value of 0 (zero) indicates that the last synchronization completed successfully. A value of 1 indicates that the synchronization did not complete successfully.
DELETE_FLG	VARCHAR2[1]	A flag used for processing deletions that represent a transient state

B.18 CONN_RECORD_DESCS

The CONN_RECORD_DESCS table (Table B–17) holds the values for the description field for each record.

Table B–17   CONN_RECORD_DESCS Table

Column	Type	Comment
FIELD_ID	Number	The foreign key to the DESC_FIELDS table
FIELD_VALUE	VARCHAR2[100]	The field value
CONN_RECORD_ID	Number	The foreign key to the CONN_USER_RECORDS table

B.19 ASSOC_DATA

The ASSOC_DATA table (Table B–18) represents the associated data for CONN_
USERS or CONN_USER_RECORDS.

Table B–18   ASSOC_DATA Table

Column	Type	Comment
SEQUENCE	Number	The sequence in which to return this data
DATA_ID	VARCHAR2[50]	The ID of the associated data
DATA_GROUP_ NAME	VARCHAR2[50]	The name for a grouping of data, such as ASSOC1 or ASSOC2
DATA_KEY_NAME	VARCHAR2[50]	The key name for the data
DATA_KEY_VALUE	VARCHAR2[100]	The data store
CONN_USERS_ID	Number	The foreign key to the CONN_USERS table
CONN_RECORD_ID	Number	The foreign key to the CONN_USER_ RECORDS table

B.20 LOVS

The LOVS table represents the lists of values (LOVS) for the synchronization level
(SYNC_LVL), PIM server type (PIM_SERVER_TYPE), domain class, and also standard
comparison operators. Table B–19 lists the lists of values by type (VARCHAR2[50]) and
value (VARCHAR2[100])

Table B–19   LOVS

LOV_TYPE (VARCHAR2[50])	LOV_VALUE (VARCHAR2[100])
SYNC_LVL	Values include: ■ Full ■ Outbound Only ■ Inbound Only ■ None
PIM_SERVER_TYPE	Values include: ■ Exchange 2007 ■ BPEL Tasks
DOMAIN_CLASS	Values include: ■ Task ■ Contact ■ Calendar

Table B–19   (Cont.) LOVS

LOV_TYPE (VARCHAR2[50])	LOV_VALUE (VARCHAR2[100])
Operators	Values include: ■ EQUALS ■ BEFORE ■ AFTER ■ GREATER_THAN ■ LESS_THAN

B.21 PROFILES

The PROFILES table (Table B–20) stores profile, section, parameter, and value information.

Table B–20   PROFILES Table

Column	Type	Comment
PROFILE_NAME	VARCHAR2[100]	The name of the profile
SECTION_NAME	VARCHAR2[100]	The name of the section
PARAMETER	VARCHAR2[100]	The name of the parameter
VALUE	VARCHAR2[500]	The value of the parameter
COMMENTS	VARCHAR2[200]	Notes and comments about this parameter
LAST_UPD	Date	The last time that this parameter was updated

B.22 SEQUENCE

The SEQUENCE table (Table B–21) stores sequence information used by Oracle TopLink to create unique IDs for new records in each table.

Table B–21   SEQUENCE Table

Column	Type	Comment
SEQ_NAME	VARCHAR2[50]	The name of the sequence
SEQ_COUNT	Number(30,0)	The counter for the sequence

C

Connector API

This appendix lists the connector APIs, their contractual agreements between the connectors and the Engine, and the data representation structures.

This appendix includes the following topics:

- Overview of the Connector API
- Connector Interface
- Engine Callback Interface API
- Connector Run-Time Interface API
- Data Representation Structures
- Best Practices for Building Connectors
- The Exchange 2007 Connector API
- Generic Component Support for Calendar Synchronization

C.1 Overview of the Connector API

The Connector API includes:

- Connector Interface
- Engine Callback Interface API
- Connector Run-Time Interface API

Terminology

Terms that relate to the API include:

- Associated Data
- Connector Name
- Create/Create Conflict
- Field Class
- PIM Domain Target
- PIM Record Description
- PIM Record ID
- PIM Record Version ID
- Synchronization Session

■ Synchronization State

C.1.1 Associated Data

Associated data is connector-defined data that the Hub stores on behalf of connectors. Although this can be data of any type, a connector must ensure that the data conforms to the AssociatedData structure.

An associated data consists of a unique identifier and a collection of groups. Each group, in turn, has a group identifier and an arbitrarily large number of KeyValuePair structures that contain the associated data. Connectors are free to define any number of groups and any number of KeyValuePair structures within each group. For example, the Oracle BDSS Connector for Microsoft Exchange 2007 defines user-level associated data with a identifier of MailboxProperties that contains a single group whose group identifier is also MailboxProperties. This group contains two KeyValuePair structures containing the keys of legacyExchangeDN and msExchHomeServerName and the values are populated with data obtained from Active Directory. Similarly, the Exchange 2007 Connector defines record-level associated data with a identifier of the PIM Record ID that contains three groups with group identifiers of SourceKey, ChangeKey, and PredecessorChangeList. The SourceKey group holds a single KeyValuePair where SK is the key and the value is the Exchange records source key value. The ChangeKey group holds a single KeyValuePair where CK is the key and the value is the Exchange records change key value. The PredecessorChangeList group holds an arbitrarily large number of KeyValuePair structures (each change key in the Exchange records predecessor change list is represented as a KeyValuePair).

Associated Data has the following characteristics:

■ It is opaque to the Hub (that is, unseen by the Hub).

■ It is optional.

■ It is stored in the data store as text.

■ It can be arbitrarily large.

■ In addition to the record-level RTL (Run-Time Library) access APIs, connectors may provide record level-associated data in the result structure of a CreateRecord or UpdateRecord call. The Hub updates the data store with the new associated data.

■ In addition to the RTL record-level access APIs, connectors may provide record level associated data in each UpsertRecord in an ExtractResponse. The Hub updates the data store with the associated data.

C.1.2 Connector Name

The name for a set of configurations for connector instances that service a specified set of PIM server instances.

> **Note:** The connector name does not change during a given synchronization session. Therefore, stateful connectors may choose to cache the value for a given user and synchronization session and use it when calling the Hub callback or RTL (Run-Time Library) APIs.

C.1.3 Create/Create Conflict

A create/create conflict occurs when two or more records from different systems are extracted during the same synchronization session that have not been previously synchronized and have key fields with matching field values.

C.1.4 Field Class

A field class is a classification of a PIM field that applies additional semantics to a PIM field beyond a data type. Examples of field classes are *Phone*, *Address*, *Email*, *Category*, *To*, *CC*, and *BCC*. The field class performs the following:

- Facilitates auto-mapping of fields between connectors.

- Enables connector implementations to implement logic based on a particular class. For example, a connector may require field information about a field class (such as *To*, *CC*, or *BCC)* to resolve a value (such as an attendee) in its system.

C.1.5 PIM Domain Target

The PIM Domain Target is optional configuration metadata for the location where the PIM data for the domain resides for the given user and a given domain. The PIM Domain Target is available to connectors whose PIM server stores PIM data in different locations for each user. For example, Exchange stores task and contact data in different folders for each user: the Exchange task data is stored in the *Task* folder (or a subfolder of the *Task* folder) and the Exchange contact data is stored in the *Contact* folder (or a subfolder of the *Contact* folder). The connector implementation interprets the configuration metadata because it is connector-specific. For example:

- A configuration of a relative path of folders such as `\root\calendar\mycalendar` would indicate synchronizing against a subfolder of a domain.

- A configuration of an abstract value such as *Root* would indicate synchronizing against the default primary folder for a domain.

Some connectors may not have the notion of PIM folders and might specify something else, such Siebel PDQ filter. For example, for the contact domain, the connector identifies a PDQ of *Sync List* as the domain target. Unlike folder paths, the PDQ does not identify a physical location on the target, but, like folder paths, it identifies the set of records that are targeted for synchronization.

> **Note:** Because this optional metadata is specified on a per-user, per-domain basis, it is incumbent on administrators to specify the PIM Domain Target for each user and each domain that is synchronized by BDSS. To simplify connector configuration, developers can create a default PIM Domain Target that enables connectors to use a default value if the configuration is not available. For example, a connector may use the default folder for a domain if the configuration is unavailable. Developers can also require a configuration value such as *Root* to indicate that the connector should use the default folder for a domain. Without a default PIM Domain Target, a connector can require the configuration if it requires the metadata for each user and domain

C.1.6 PIM Record Description

The PIM Record Description is an array of KeyValuePair structures. Each `KeyValuePair` structure holds a Hub field name as the *key* and a Hub field value as the *value*. Connectors must obtain the list of Hub fields configured to be included in the PIM Record Description for each **Hub domain** by using the RTL (Run-Time Library) API `GetConfigurationMetadata`.

The configuration of the Hub fields used in the PIM Record Description is contained in the *FtsKeyFields* profile. The section name of the profile contains the **Hub domain** name. The parameters in the profile contain such names as *KeyFieldX* where X is a number that not only distinguishes the row from other rows in the table, but specifies the order in which the field should appear when a connector constructs a PIM Record Description. Connectors must build PIM Record Descriptions (if the configuration is present) and ensure the order that the fields appear in the `KeyValuePair` array are in the same sequence as specified by the *KeyFieldX*. For example, the *Task* Hub domain has the following configuration in the *FtsKeyFields* profile (listed in Table C–1).

Table C–1   Task Hub Domain Configuration

Profile	Section (a Hub Domain Name)	Parameter	Value
FtsKeyFields	Task	KeyField1	Subject
FtsKeyFields	Task	KeyField2	StartDate

When a connector extracts a record, it constructs a `KeyValuePair` array containing two elements. The first element (index0) contains a `KeyValuePair` where the key is *Subject* and the value is the Hub field value on the record. The second element (index1) contains a KeyValuePair where the *key* is *StartDate* and the value is the Hub field value is on the record.

If a Hub record does not have a value for a configured key field, the connector must provide an empty string for that field.

> **Note:** The connector must provide Hub field names and Hub field values in the record description. Because the connector provides this information, connectors must extract the record from the PIM, transform it to Hub format and then construct the PimRecordDescription. Similarly, the connector must provide Hub field names and values upon a successful Create or Update operation. This is crucial because the Hub uses the PIM Record Description containing these key fields to perform key field matching on PIM records when it evaluates extract responses from multiple PIM servers. The PIM Record Description is also used when logging information about a given operation for a Hub record as it travels through the system.

For Oracle Fusion Middleware 11*g* release 1 of BDSS, the field designated as a member of the record description cannot be a multivalue type.

C.1.7 PIM Record ID

The PIM Record ID is an identifier used by a connector for uniquely identifying a record on the PIM server. The PIM Record ID should be unique within a PIM server on

a per-user, per-domain basis. The connector should ensure that the property used as the PIM Record ID can be used to identify and access the record.

Connectors provide the PIM Record ID to BDSS when:

- Extracting records from the PIM server

- Completing a pushed create operation

BDSS provides the PIM Record ID to connectors when pushing an Update or Delete operation to the PIM server.

C.1.8 PIM Record Version ID

The PIM Record Version ID is a connector-defined value that captures the version of a PIM record. It can be any value if it can track a modified PIM record. For example, if a PIM record has a *Last Modification Date/Time* field, a connector might opt to use it as the PIM Record Version ID because the *Last Modification Date/Time* field is updated whenever the record is updated. Similarly, if a PIM record has an integer value that is incremented when the record changes, then the connector might use the integer value.

C.1.9 Synchronization Session

A synchronization session is a session in which the Hub synchronizes the data for a **Hub user** for one or more Hub domains between one or more PIM stores through the BDSS connectors. The session starts when the Hub calls the connector API `InitializeUserSyncSession` and is identified by a unique identifier called the Synchronization Session ID. The session ends when the Hub calls the `EndUserSyncSession` API. In between these two calls, the Hub may call the connector APIs:

- `ExtractDomains` API: This API is not called on a connector if the Hub configuration is such that the connector is configured for inbound synchronization only for all Hub domains. It is called if the Hub configuration is such that the connector is configured for bidirectional or outbound-only synchronization for at least one Hub domain.

- `CreateRecord`, `UpdateRecord`, or `DeleteRecord` APIs may be called any number of times during a session.

C.1.10 Synchronization State

The Engine tracks the user's domain records that have been exported successfully from a PIM server using connector-provided data known as the synchronization state. The synchronization state is monitored at the connector domain level.

Connectors define the synchronization state and its format. For example, connectors can use a date/time stamp to indicate the time that they performed an extract query during a prior synchronization session, or they can use a convention defined by the PIM APIs that describe how the connector exchanges data with its PIM server. The Oracle BDSS Connector for Exchange 2007, for example, uses an Incremental Change Synchronization (ICS) binary BLOB. For more information about how connectors manage synchronization state, see Section C.4.3, "CommitCachedSyncState Method API," Section C.4.8, "GetSyncState API," and Section C.4.10, "SaveSyncState API."

> **Note:** Although this data is opaque to the Engine, the Engine dictates when the synchronization state is saved to the Hub data store.

There is one synchronization state for each connector domain for a user. The synchronization state, which resides in the Hub data store, enables the system to extract all of the records from a domain or just the Create, Read, Update, and Delete operations that have occurred from a prior synchronization state.

> **Note:** The Engine updates the synchronization state of a connector domain after each synchronization session for a user, but only if all other connectors do not report errors when processing the records they are sent. If the connectors report errors, then the synchronization state cannot be updated, as the system must again extract some records in a future synchronization session (after the errors have been fixed).

C.1.11 Update/Update Conflict

An update/update conflict occurs when a previously synchronized record is updated in multiple systems and is extracted during the same synchronization session.

C.2 Connector Interface

A connector's PIM server synchronization services are exposed as a Web service. Connector developers implement the Web service interface defined in this document. The Engine, test applications, and other Web service clients are consumers of the Connector Interface, which includes the following API:

- InitializeUserSyncSession API

- ExtractDomains API

- EndDomainSynchronization API

- EndUserSyncSession API

- CreateRecord API

- UpdateRecord API

- DeleteRecord API

- GetPimServerEndPoint API

> **Note:** The ExtractDomains API is asynchronous. The other APIs are all synchronous.

C.2.1 InitializeUserSyncSession API

The Hub calls the InitializeUserSyncSession API at the beginning of each **synchronization session** for each user. Connectors should perform any user-based initialization required for a user to enable them to service subsequent requests during the synchronization session identified by the given syncSessionId. Table C–2 lists the parameters of the InitializeUserSyncSession API.

Table C–2   Parameters of the InitializeUserSyncSession API

Parameter	Data Type	Type	Description
connectorName	String	[in]	Contains the **connector name**. This parameter must be non-null and cannot be empty.

Table C–2   (Cont.) Parameters of the InitializeUserSyncSession API

Parameter	Data Type	Type	Description
syncSessionId	String	[in]	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
hubUserId	String	[in]	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	[in]	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
rtlUrl	String	[in]	Contains a URL that the connector should call if it must obtain metadata from BDSS. This parameter must be non-null and cannot be empty.
hubDomainArray	DomainInfo []	[in]	Contains an array of DomainInfo structures. The array contains metadata for all configured Hub domains. This parameter must be a non-null array. Each DomainInfo structure must meet the following criteria:
			■ pimDomainTarget—This member must be non-null and cannot be empty.
			■ pimFilterCondition—This member must be non-null, but may be empty.
			■ hubDomain—This member must be non-null and cannot be empty.
			■ pimDomainSyncState—This member must be non-null, but may be empty.
			Connectors should validate the array and return an error if the given DomainInfo array does not meet this criteria.
connectorUrl	String	[out]	An optional parameter. On input, the parameter is NULL. On output, it contains the full URL of the connector that received the call or remains NULL.
			The connector that received this call can provide its URL if the connector requires BDSS to call the same connector instance (identified by the URL) when making subsequent calls for the given **Hub user** during the synchronization session identified by the given synchronization session ID.

Return Value

void

Exceptions

If the connector cannot initialize the user, it should throw an exception and provide a message identifying the problem. This allows the message to be logged by BDSS.

C.2.1.1 Engine Contract

To call the `InitializeUserSyncSession` API, the Engine must perform the following:

- Provide the domain metadata for all Hub domains.

- Call the `InitializeSyncSession` API before calling `ExtractDomains`, `CreateRecord`, `UpdateRecord`, or `DeleteRecord`.

> **Note:** The Engine can call `GetPimServerEndPoint` before calling `InitializeSyncSession` API.

- If the connector throws an exception, the Hub considers this an indication that the connector could not initialize the user. Therefore, the Hub does not make any subsequent calls to the connector for the given **Hub user** and synchronization session.

- Provide the Connector Run-Time Library URL the connector should use for a given user and synchronization session. If a connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided URL.

- Provide the connector name. If the connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided `connectorName`.

C.2.1.2 Connector Contract

The connector must perform in response to the Engine's `InitializeUserSyncSession` API call:

- Validate each parameter passed in to the API to ensure that it meets the criteria outlined in Table C–2.

- Perform any required user-based initialization to enable the connector to service subsequent Hub requests. These requests may be for extracting Hub domains, or for creating, updating, or deleting records.

- Provide the URL of the connector instance that received the call if the connector requires all subsequent calls to be routed to the same connector instance.

- If a connector must cache any of the parameters passed to this API to service subsequent Hub calls, it may use a combination of the `syncSessionId` and `pimUserId`. Most connectors require caching the provided domain information data for subsequent use during the **synchronization session**. It is considered a connector implementation detail if a stateless connector must maintain state.

- Use the provided RTL (Run-Time Library) URL and connector name if the connector must call the Connector Run-Time Library.

C.2.2 ExtractDomains API

The Engine calls `ExtractDomains`, an asynchronous Web service API, to request a connector to extract the given Hub domains. Extraction is the process of pulling and downloading all PIM records from the PIM store that have undergone a change. Connectors should extract all of the PIM records that have been modified since the last synchronization state, which is provided by the Hub for each user and domain at the beginning of the synchronization session when the Hub calls `InitializeUserSyncSession`. Each PIM record should be converted to a Hub

format as defined in Chapter 8, "Mapping Connector Fields to Hub Fields". The Hub-formatted records are provided to the Hub through the ExtractDomainResultsCallback API.

Table C–3   Parameters of the ExtractDomains API

Parameter	Data Type	Type	Description
extractRequest	ExtractRequest	[in]	Contains information related to the user and domains that need extraction. This parameter must not be null.
			The ExtractRequest members must meet the following criteria:
			■ hubUserId—The member must be non-null and cannot be empty.
			■ pimUserId—This member must be non-null and cannot be empty.
			■ hubDomainArray—This member must not be null. Each element in the array must be a non-null and cannot be empty string containing a single Hub Domain ID.
hubContext	HubContext	[in]	Contains context information that a connector may require to service the extract request. Must not be null.
			The HubContext members must meet the following criteria:
			■ engineEndPointURL—This member must be non-null and cannot be empty.
			■ syncSessionId—This member must be non-null and cannot be empty.
			■ connectorRuntimeLibraryURL—This member must be non-null and cannot be empty.
			■ connectorName—This member must be non-null and cannot be empty.

Connectors must implement this method as a nonblocking asynchronous method. The implementation should ensure data extractions for each domain on a separate connector-owned thread, as opposed to performing the extraction on the thread of the calling Web service.

C.2.2.1 Engine Contract

The Engine performs the following when it calls the ExtractDomains API.

- Ensures a successful call to InitializeUserSyncSession has been made for the given Synchronization Session before calling this API.

- Analyzes the ExtractResponses obtained from all connectors; resolve conflicts, and push Create, Read, Update, and Delete operations to other PIM servers through other PIM connectors.

- Maintains mappings of PIM records that have been exported from one system and imported to another.

- Ensures that the `ExtractResponses` provided by a connector are asynchronously processed.

- Provides the Connector Run-Time Library URL that the connector should use for a given user and synchronization session. If a connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided URL. The same URL is provided in each subsequent call to the connector during the same synchronization session.

- Provides the connector name. If the connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided `connectorName`.

> **Note:** The array of Hub domains contained in the `ExtractRequest` must be a subset of the Hub domains specified in the Hub call to `InitializeUserSyncSession` during the same synchronization session.

C.2.2.2 Connector Contract

The connector responds as follows to the `ExtractDomains` API:

1. Extracts all records from the PIM store belonging to a domain for the given **PIM User ID** and PIM Domain Target since the last **synchronization state**. Connector implementations should extract each domain in the given domain array on separate threads and make asynchronous callbacks to the Engine using the callback URL contained in the given `HubContext` structure. The `ExtractResponse` should not include the following types of updated PIM records:

   a. Any records in the same PIM Domain Target that are not part of the domain must be ignored. For example, it is possible for an Exchange E-mail item (`IPM.Note` domain) to reside in the *Task* folder (`IPM.Task`). In this case, the Exchange 2007 Connector must ignore the `IPM.Note` item when extracting `Task`.

   b. Any records that are filtered out by the PIM Filter Condition.

2. Provides `ExtractResponses` to the Hub for each **Hub domain** specified in the request. For each PIM record that has been modified since the last synchronization state, there should be one UpsertRecord containing a Hub representation of the modified PIM record. There should also be one PIM Record ID for each record that has been deleted from the PIM server since the last synchronization state.

3. Each connector has a BDSS profile parameter configuration called *Extract Response Batch Size*. The value of this parameter contains the maximum number of records that each `ExtractResponse` can have. Connectors should be written to provide a series of `ExtractResponses` where each `ExtractResponse` contains this maximum number of records. The Engine uses the receipt of these batches to reset internal timeouts used for determining connector failures, such an abnormal termination of a connector. For example, if a connector must extract 5,000 calendar records and the connector's configured batch size is 100, then the connector performs at least 50 callbacks to the Engine.

> **Note:** For Oracle Fusion Middleware 11*g* release 1, BDSS does not support receiving multiple `ExtractResponses` for each Hub domain. Therefore, the connector configuration for the *Extract Response Batch Size* should be large enough such that extraction of any domain for any user results in a single `ExtractResponse`.

4. Uses the provided Synchronization Session ID, **Hub domain**, and **Hub User ID** in logging statements.

5. Constructs a PIM Record Description for each extracted record.

6. If extraction of any record fails, the connector should terminate the extraction and notify the Hub through the `ExtractDomainResultsCallback` API. The connector must provide an indication of the error by setting the result code to indicate an error and providing a error description string so the Hub can log information about the failure.

7. As indicated in the PIM Filter condition definition, the connector must ensure to evaluate the configured filter to each record extracted to determine whether the record should be synchronized.

> **Note:** For Oracle Fusion Middleware 11*g* release 1 of BDSS, connectors should include which PIM date field is used for comparisons against the PIM Filter Condition in the software code. If the PIM record does not contain a value for the PIM date field that is used in the comparison, then the record should be included in the ExtractResponse.

For more information about how connectors apply filters, see Section C.2.2.3.3, "Preventing the Download of Incorrect Records."

8. For every inbound record extracted, the connector must convert it from the native PIM API format to an XML message that conforms to a PIM XML schema (XSD file) defined for the PIM domain. After the PIM XML message is constructed, the connector must then transform the PIM XML message to a Hub XML message that conforms to a provided Hub XML schema (XSD) using XSL transformations. Because the connector does not know which PIM field is mapped to which Hub PIM, nor how the data is to be translated, it cannot convert directly from PIM API format to Hub XML format. See also Section 8.1, "Overview of Data Transformation."

9. Perform **echo** suppression and remove all echoes from the collection of records extracted from the PIM server before providing the record set to the Hub through the `ExtractDomainResultsCallback` API. If any echoes are removed, then the connector must set the `echoesSuppressed` flag member of the `ExtractResponseMetadata` to `TRUE` when providing the echo-less record set to the engine.

10. Some connectors may be required to maintain state across BDSS calls while others may not. Regardless of whether a connector maintains state, it must meet the contractual agreements with BDSS related to saving synchronization state. Therefore, the following APIs are provided to facilitate saving of synchronization state for stateful and stateless connectors:

    a. If the connector is stateless, it should cache the synchronization state through `CacheTempSyncState` as soon as it collects the final batch of modified

records from the PIM server and just before providing the batch to BDSS. When the Hub completes the domain synchronization, it calls `EndDomainSynchronization` and indicate whether the synchronization state must be saved. If so, the connector must commit the cached synchronization state by calling `CommitCachedSyncState`.

b. If the connector maintains state information between BDSS calls, it should use only the `SaveSyncState` API to directly save the synchronization state when BDSS calls `EndDomainSynchronization` and indicates that the synchronization state should be saved.

11. Delete Detection.

The Hub provides support for the following two delete use cases for connectors:

a. Use Case 1: A connector cannot natively detect delete operations. In this case, connectors can determine which records have been deleted from their PIM server by building an array containing all the PIM Record Identifiers for the given user and domain. (This array should contain record identifiers for modified and unmodified records.) The connector can provide this array of record identifiers to the Connector Run-Time Interface method `GetDeletesByIds` to obtain an array of record identifiers that identify deleted records. The connector can then add these record identifiers to the `deleteRecordIdArray` of the `ExtractResponseData` member of the `ExtractResponse`. If the connector does not have the ability to query for all the record identifiers, then the connector cannot synchronize delete operations.

b. Use Case 2: A connector cannot natively detect deletions of records created by BDSS when an end user deletes the record before a subsequent synchronization session.

Because the synchronization state tracks the records that have been successfully exported from the PIM, some connectors cannot detect the deletion because the just-created record is deleted from the PIM store before being exported from the store (that is, the record has been imported to the store, but never exported, so the synchronization state may not detect the deletion).

This use case is the equivalent of an end user creating a record and then deleting it in the PIM whereby both operations occur outside the scope of a synchronization session. For example, if BDSS synchronizations occur every 5 minutes, then a user could use Outlook to create and delete a record in Exchange before the 5 minutes elapses. When that user is synchronized, the synchronization state would not capture the record as having been created and then deleted. In other words, the synchronization state is not necessarily a historical record of operations done on the PIM store. In the context of BDSS, the only difference is that BDSS performs the record creation.

The connector must be able to identify echoes to leverage this feature. In this case, as the connector performs echo suppression, the connector builds an array containing PIM Record Identifiers that identify the echoes and then calls the `GetDeletesByPendingCreateEchoes` to obtain the record IDs of deleted records. The connector can then add these record identifiers to the `deleteRecordIdArray` of the `ExtractResponseData` member of the `ExtractResponse`.

12. This API is asynchronous. The connector implementation should do as little preprocessing of the request as possible and return control to the Hub. That is, the API should post the extract request (or metadata about the request) to a

connector-owned thread and then return control to the Hub. In this way, the more time-consuming operation of extracting PIM data occurs on a connector-owned thread. If the connector successfully receives the Web service call, it is free to throw exceptions. However, it should only throw exceptions if the preprocessing fails. After the API returns, the threads that performs the extraction should provide the Hub with an `ExtractResponse` for each Hub domain that indicates success or failure through the `ExtractDomainResultsCallback` API.

13. Use the provided RTL (Run-Time Library) URL and connector name if the connector must call the Connector Run-Time Library. Both the RTL URL and the connector name are provided in the `HubContext` structure.

Return Value

void

Exceptions

If the connector cannot post the extract request to a connector-owned thread, it should throw an exception and provide a message identifying the problem. This action enables BDSS to log the message.

C.2.2.3 Best Practices for ExtractDomains

This section describes the best practices for ExtractDomains. Topics include:

- Economical Extracts
- Caching PIM Field Definitions
- Preventing the Download of Incorrect Records

C.2.2.3.1 Economical Extracts Connectors should predetermine whether there are any changes that require extraction for a given domain, because performing a PIM extract where no changes exist is a resource-intensive and time-consuming operation. For example, when the Engine calls the connector to extract a domain, the connector then associates the user to a connector application and then allocates numerous MAPI resources, only to discover that there are no changes. As a more economical approach, the connector associates the user with an application whose only function is issuing a WebDAV request to the Exchange server that queries for changes since the last synchronization state.

C.2.2.3.2 Caching PIM Field Definitions Connectors should obtain PIM field definitions and other domain-related configuration information through the Connector Run-Time Interface method, `GetPimDomainMetadata`, and cache them for use during the extract cycle.

C.2.2.3.3 Preventing the Download of Incorrect Records When extracting the calendar and task domains, apply the `pimFilterCondition` in a manner that prevents records from being downloaded from the server if they fail to the meet the criteria of the filter. For example, a connector could do a brute force method of filtering records by first downloading all changed records from the server and then evaluating every record against the filter to determine if it should be included in the extract result set. A better method is to apply the filter in a manner that prevents those records from being downloaded from the PIM server at all. For example, MAPI restriction constructs are used for the Exchange 2007 Connector to prevent the downloading of old records.

C.2.3 EndDomainSynchronization API

The Hub calls this `EndDomainSynchronization` interface to end the domain synchronization for a given user. This interface may be called to terminate a **synchronization session** gracefully (or not gracefully) for a user's **Hub domain**.

Table C–4 Parameters of the EndDomainSynchronization API

Parameter	Data Type	Type	Description
connectorName	String	[in]	Contains the **connector name**. This parameter must be non-null and cannot be empty.
syncSessionId	String	[in]	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
rtlUrl	String	[in]	Contains the URL to the connector run-time interface that the connector should use for the given synchronization session.
hubDomain	String	[in]	Identifies the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	[in]	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	[in]	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
saveSyncState	Boolean	[in]	A Boolean value that indicates whether the connector should save the **synchronization state**
extractResponseCount	Integer	[out]	Contains the number of batches that the connector successfully sent to the Hub for the given domain and user for a particular synchronization session. For Oracle Fusion Middleware 11*g* release 1 of BDSS, this parameter is either 0 or 1.

C.2.3.1 Engine Contract

The Engine performs the following when it calls the `EndDomainSynchronization` API:

1. Calls this method at the conclusion of processing the given domain.

2. Indicates whether the connector should save the synchronization state. Sets the flag to `False` if:

   - The domain was never extracted during the current synchronization session.

   - The domain was extracted and the connector provided an `ExtractResponse` containing 0 records and an `echoesSuppressed` flag of false. In this case, no records were extracted (echoes or otherwise).

> **Note:** The connector sets the `echoesSuppressed` flag in the `ExtractResponse` indicating whether the `ExtractResponse` had echoes removed. For Oracle Fusion Middleware 11*g* release 1 of BDSS, the Hub does not support receiving multiple `ExtractResponses` for a single Hub domain.

3. Uses the `extractResponseCount` for cleaning up `ExtractResponses` that may have been received by the Hub when errors occur.

4. Provides the Connector Run-Time Library URL used by the connector for a given user and synchronization session. If a connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided URL

5. Provides the connector name. If the connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided `connectorName`.

C.2.3.2 Connector Contract

A connector responds as follows when the Engine calls the `EndDomainSynchronization` API:

1. Cleans up any domain-level resources related to the user, if appropriate.

2. If `saveSyncState` is `true`, the connector saves the domain synchronization state to the Hub data store through a Connector Run-Time Interface API as indicated in the connector contract section of the ExtractDomains API.

3. Ensures that no further `ExtractResponses` are sent to the Hub even in a scenario where the Hub calls the API to terminate the synchronization session non-gracefully.

4. Ensures that `extractResponseCount` is always populated with the correct number of successfully sent `ExtractResponses` to the Hub for the given Synchronization Session.

5. Uses the provided Synchronization Session ID, **Hub domain**, and **Hub User ID** in logging statements.

6. Uses the provided RTL (Run-Time Library) URL and connector name if the connector must call the Connector Run-Time Library.

> **Note:** The connector should perform a do-nothing operation (no-op) on this call if the Hub calls it for a user session of which the connector is unaware. This may occur, for example, if the Hub decides to terminate a synchronization session due to another connector failure. This helps to simplify the Hub logic when the Hub must terminate a **synchronization session** abruptly.

Return Value

Integer

- 0 for SUCCESS/OK
- Nonzero for ERROR/NOTOK

C.2.4 EndUserSyncSession API

The Engine calls the `EndUserSyncSession` interface at the conclusion of synchronizing all domains for the given user to end the synchronization session identified by the given `syncSessionId`. Connector implementations can use this opportunity to free any user-specific resources that may have been associated with the given user. Table C–5 lists the parameters of the `EndUserSyncSession` interface.

Table C–5   Parameters of the EndUserSyncSession API

Parameter	Data Type	Type	Description
connectorName	String	[in]	Contains the **connector name**. This parameter must be non-null and cannot be empty.
syncSessionId	String	[in]	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
rtlUrl	String	[in]	Contains the URL to the Connector Run-Time Interface that the connector should use for the given **synchronization session**
hubUserId	String	[in]	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	[in]	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.

C.2.4.1 Engine Contract

The Engine performs the following when it calls the `EndUserSyncSession` API:

1. Does not call the connector for the same user until another synchronization session has started.

2. Calls this API to end the synchronization session identified by the given `syncSessionId`.

3. Provides the Connector Run-Time Library URL that the connector should use for a given user and synchronization session. If a connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided URL.

4. Provides the connector name. If the connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided `connectorName` parameter.

C.2.4.2 Connector Contract

The connector responds as follows when the Engine calls the `EndUserSyncSession` API:

1. Cleans up all resources related to the user.

2. Uses the provided Synchronization Session ID, **Hub domain** and **Hub User ID** in logging statements.

3. Uses the provided RTL (Run-Time Library) URL and connector name if the connector must call the Connector Run-Time Library.

Return Value

Integer:

- 0 for SUCCESS/OK

- Nonzero for ERROR/NOTOK

> **Note:** The connector should perform a do-nothing operation (no-op) on this call if the Hub calls it for a user session of which the connector is unaware. This may occur, for example, if the Hub decides to terminate a synchronization session due to another connector failure. This helps to simplify the Hub logic when the Hub must terminate a **synchronization session**.

C.2.5 CreateRecord API

The Hub calls the `CreateRecord` interface any time a record must be created in the PIM. Table C–6 lists the parameters of the `CreateRecord` interface.

> **Note:** The Hub can call this API even if the domain has not been extracted.

Table C–6   Parameters of the CreateRecord API

Parameter	Data Type	Type	Description
connectorName	String	[in]	Contains the connector name. This parameter must be non-null and cannot be empty.
syncSessionId	String	[in]	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
rtlUrl	String	[in]	Contains the URL to the connector run-time interface that the connector should use for the given synchronization session when calling the run-time library
hubDomain	String	[in]	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	[in]	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	[in]	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.

Table C–6   (Cont.) Parameters of the CreateRecord API

Parameter	Data Type	Type	Description
hubRecord	**upsert record**	[in/out]	On input, contains the Hub record in Hub XML format. The RecordMetadata member must be null. On output, the RecordMetadata is allocated and populated by the connector to contain:

- PIM Record ID—A required parameter, which must be non-null and cannot be empty.

- PIM Record Associated Data— An optional parameter, which may be null. If non-null, at least one of the associated data members must be non-null and cannot be empty.

- PIM Record Version ID—A required parameter, which must be non-null and cannot be empty.

- Optionally, set recordError to *true* if the connector failed to create the record and specify a recordErrorDescription.

- PIM Record Description—This parameter (one that is optional and configuration-driven) may be null if no KeyField configuration is available for the **Hub domain**. If the KeyField configuration does exist for the **Hub domain**, the array must not be null. Each KeyValuePair in the description array must have a key that is non-null and cannot be empty and also a value that also must be non-null and cannot be empty.

C.2.5.1 Engine Contract

The Engine performs the following when it calls the CreateRecord API:

1. After returning from this connector API call, the Engine must update its data store with the following data contained in the provided RecordMetadata member of the UpsertRecord:

   - The PIM Record ID

   - The PIM Record Version ID

   - The PIM Record Associated Data (if provided)

   - The PIM Record Description (if provided)

2. In addition to updating the Hub data store to contain the metadata, the Engine must also set the echoPending flag in the store for the given row to TRUE if the record was created successfully.

3. Provide the Connector Run-Time Library URL that the connector should use for a given user and synchronization session. If a connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided URL.

4. Provide the connector name. If the connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided `connectorName`.

C.2.5.2 Connector Contract

The connector responds as follows when the Engine calls the `CreateRecord` API:

1. Creates the record in the PIM store and provide the metadata as indicated in Table C–6.

2. Translates Hub-formatted XML messages to a format suitable for creating a record in the PIM server.

3. Uses the provided Synchronization Session ID, **Hub domain**, and **Hub User ID** in logging statements.

4. Uses the provided RTL (Run-Time Library) URL and connector name if the connector must call the Connector Run-Time Library.

Return Value

Integer

■ 0 for SUCCESS/OK

■ Nonzero for ERROR/NOTOK

C.2.6 UpdateRecord API

The Engine calls the `UpdateRecord` interface anytime that a record must to be updated in the PIM server. Table C–7 lists the parameters of the `UpdateRecord` method.

> **Note:** The Hub can call this API even if the domain has not been extracted.

Table C–7   Parameters of the UpdateRecord API

Parameter	Data Type	Type	Description
connectorName	String	[in]	Contains the **connector name**. This parameter must be non-null and cannot be empty.
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
rtlUrl	String	[in]	Contains the URL to the Connector Run-Time Interface that the connector should use for the given synchronization session when calling the run-time library
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.

Table C–7   (Cont.) Parameters of the UpdateRecord API

Parameter	Data Type	Type	Description
hubRecord	UpsertRecord	In	Contains the Hub record and metadata, obtained from the Hub data store, about the record to be updated
			The RecordMetadata member must not be null and should be populated by the Hub with the connector-specific metadata.
updatedMetadata	RecordMetaData	Out	On output, contains updated record metadata, obtained from the PIM Record that includes:
			■ PIM Record ID: This is a required parameter. This parameter must be non-null and cannot be empty.
			■ PIM Record associated data: This parameter is optional and may be null. If it is non-null, then at least one of the associated data members must be non-null and cannot be empty.
			■ PIM Record Description: This parameter (one that is optional and configuration-driven) may be null if no KeyField configuration is available for the **Hub domain**. If the KeyField configuration exists for the Hub domain, then the array must not be null. Each KeyValuePair in the description array must have a key that is non-null and cannot be empty and a value that is non-null and cannot be empty.
			■ Optionally set recordError to true if the connector failed to create the record and specify a recordErrorDescription
			■ PIM Record Version ID (required). This parameter must be non-null and cannot be empty.

C.2.6.1 Engine Contract

The Engine performs the following when it calls the UpdateRecord API:

1. The Engine performs the following when resolving conflicts detected when evaluating extract responses:

    a. For an Update/Update Conflict, the Hub should provide the losing connector with the AssociatedData from the BDSS store and not the AssociatedData provided in the extracted record when calling this API on the losing connector.

    b. For a Create/Create Conflict, the Hub should use the AssociatedData provided in the extracted record from the losing connector when calling this API on the losing connector.

2. In addition to updating the data store to contain the metadata, the Engine must also set the echoPending flag in the store for the given row to FALSE if the record was successfully updated.

3. Provide the Connector Run-Time Library URL that the connector should use for a given user and synchronization session. If a connector must call the Connector

Run-Time Library as part of its implementation of this method, it should use the provided URL.

4. Provide the connector name. If the connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided **connector name**.

C.2.6.2 Connector Contract

The connector responds as follows when the Engine calls the `UpdateRecord` API:

1. Updates the record identified by the given PIM Record ID in the PIM server and populates the RecordMetaData structure.

2. If the record is deleted in the PIM server after extraction by BDSS before the connector processes the update, the connector should provide some indication (that is, a log message) that the update was ignored because the record was deleted. (A subsequent extract cycle should detect the delete operation and propagate it to the Engine.)

3. If the PIM API can detect conflicts that may occur if the PIM record changes just before pushing an outbound update, the connector should resolve the conflict in a "BDSS wins" fashion (that is, the record that BDSS gave to the connector should overwrite the end user's changes).

4. Uses the provided Synchronization Session ID, **Hub domain**, and **Hub User ID** in logging statements.

5. Uses the provided RTL (Run-Time Library) URL and connector name if the connector must call the Connector Run-Time Library.

Return Value

Integer:

- 0 for SUCCESS/OK

- Nonzero for ERROR/NOTOK

C.2.7 DeleteRecord API

The Engine calls the `DeleteRecord` API whenever a record must be deleted from the PIM server. Table C–8 lists the parameters of the `DeleteRecord` method.

> **Note:** The Hub can call this API even if the domain has not been extracted.

Table C–8   Parameters of the DeleteRecord API

Parameter	Data Type	Type	Description
connectorName	String	[in]	Contains the **connector name**. This parameter must be non-null and cannot be empty.
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.

Table C–8   (Cont.) Parameters of the DeleteRecord API

Parameter	Data Type	Type	Description
rtlUrl	String		Contains the URL to the Connector Run-Time Interface that the connector should use for the given synchronization session when calling the Connector Run-Time library.
hubDomain	String	[in]	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
pimRecordId	String	In	Contains the PIM Record ID of the record to delete. This parameter must be non-null and cannot be empty.
pimRecordDescription	PIM Record Description	In	Contains the PIM Record Description

C.2.7.1 Engine Contract

The Engine performs the following when it calls the DeleteRecord API:

1. Provides the Connector Run-Time Library URL that the connector should use for a given user and synchronization session. If a connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided URL.

2. Provides the connector name. If the connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided connectorName parameter.

C.2.7.2 Connector Contract

The connector responds as follows when the Engine calls the DeleteRecord API:

1. Deletes the record in the PIM identified by the given PIM Record ID.

2. If a connector needs the PIM Record associated data, it may obtain it through the Connector Run-Time Interface method, GetRecordAssociatedData. Most PIM servers need only the PIM Record ID.

3. Uses the provided Synchronization Session ID, **Hub domain**, and **Hub User ID** in logging statements.

4. Returns a non-error code if the PIM record is deleted before sending the delete to the PIM. The connector may log a message indicating that the delete failed because the record had been deleted previously.

5. Uses the provided RTL (Run-Time Library) URL and connector name if the connector must call the Connector Run-Time Library.

Return Value

Integer:

- 0 for SUCCESS/OK

- Nonzero for ERROR/NOTOK

C.2.8 GetPimServerEndPoint API

The Hub calls the `GetPimServerEndPoint` interface to obtain the **PIM Server Endpoint** for a given synchronization-enabled user and uses the returned `endPointToken` to determine which connector instance in a cluster of connectors should be called when synchronizing a user during a **synchronization session**. Table C–9 lists the parameters of the `GetPimServerEndPoint` interface.

Table C–9   Parameters of the GetPimServerEndPoint Interface API

Parameter	Data Type	Type	Description
connectorName	String	[in]	Contains the **connector name**. This parameter must be non-null and cannot be empty.
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
rtlUrl	String	[in]	Contains the URL to the Connector Run-Time Interface the connector should use for the given synchronization session when calling the Connector Run-Time Library.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
endPointToken	String	Out	On output, contains a connector-defined endpoint token. This parameter can be NULL. It can also be empty.
canSyncUser	Boolean	Out	On output, indicates whether the receiving connector can synchronize the given user

C.2.8.1 Engine Contract

The Hub calls the `GetPimServerEndPoint` API only when it must obtain the **PIM Server Endpoint** for a user. The Hub performs the following when it calls this API:

1. Uses configuration metadata and the returned `endPointToken` and the `canSyncUser` to determine which connector should be called to extract the given user. The Hub should first check the `endPointToken`. If the `endPointToken` value is a non-NULL and cannot be empty string, then the Hub should use the `endPointToken` value to obtain the connector URL to use to synchronize the

user. If it is NULL or is empty, the Hub should then check the Boolean value to determine if the receiving connector that returned the Boolean value can be used to synchronize the user.

2. Provide the Connector Run-Time Library URL that the connector should use for a given user and synchronization session. If a connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided URL.

3. Provide the connector name. If the connector must call the Connector Run-Time Library as part of its implementation of this method, it should use the provided `connectorName`.

C.2.8.2 Connector Contract

The connector that receives the `GetPimServerEndPoint` API call performs the following:

1. Resolves the **PIM Server Endpoint** for the given user. If the connector cannot resolve the given user, it should return an error.

> **Note:** The receiving connector instance may not be the same instance that performs the synchronization of the user (or even in the same cluster).

The PIM Server Endpoint value (`endPointToken`) is a connector implementation detail. Developers must document the PIM server endpoint values so that administrators can correctly map the URLs to the endpoints. For example, for Exchange, the connector returns the NetBios name of the Exchange server hosting the given users mailbox. Therefore, an administrator must map these Exchange server NetBios names to the cluster URLs.

2. Uses the provided Synchronization Session ID, **Hub domain**, and **Hub User ID** in logging statements.

3. Uses the provided RTL (Run-Time Library) URL and connector name if the connector must call the Connector Run-Time Library.

Return Value

Integer:

- 0 for SUCCESS/OK
- Nonzero for ERROR/NOTOK

C.3 Engine Callback Interface API

The Engine Callback Interface is a Web service-based interface consisting of a single method, `ExtractDomainResultsCallback`, that a connector calls when it wants to provide the Hub with an ExtractResponse.

C.3.1 ExtractDomainResultsCallback API

The `ExtractDomainResultsCallback` API is asynchronous and returns immediately to the calling connector after the Engine queues the information provided in the given `ExtractResponse`. Table C–10 lists the parameters of the `ExtractDomainResultsCallback` method.

Table C–10   Parameters of the ExtractDomainResultsCallback API

Parameter	Data Type	Type	Description
extractResponseData	ExtractResponse	In	Contains a batch of records provided by a connector

C.3.2 Engine Contract

When the Engine calls the `ExtractDomainResultsCallback` API, it queues the `ExtractResponse` and returns immediately.

C.3.3 Connector Contract

The Connector performs the following in response to the Engine's `ExtractDomainResultsCallback` API call:

- When the connector extracts records from the PIM and builds the given `ExtractResponse`, it must remove any echoes from the response before calling this API and it must set the `echoesSuppressed` flag to `true` if echoes were removed.

- The connector must specify a sequence number of the `ExtractResponse`. The first extract response is numbered 1, the second is numbered 2, and so on. For Oracle Fusion Middleware 11*g* release 1 of BDSS, the sequence number is always 1.

Return Value

void

C.4 Connector Run-Time Interface API

The Connector Run-Time Interface is a Web service-based API that allows connectors to access certain metadata, such as `AssociatedData`, in the Hub data store. This interface provides services that are common to all connectors, such as saving synchronization state.

This interface is not implemented by connector authors. Connector implementations are the consumers of this interface.

In general, all APIs return an integer to indicate success (0) or failure (nonzero). All APIs can throw a `RemoteException` when irrecoverable errors occur, such as the inability to read or write to the BDSS store or invalid parameters. Developers should write connectors to account for exceptions and the return value to determine the overall success of the API call.

C.4.1 GetPimRecordVersions API

The `GetPimRecordVersions` method enables a connector to retrieve the PIM Record Version ID of one or more PIM records stored in the BDSS data store. Retrieving this information enables the connector to determine the last version of a PIM record (or PIM records) that BDSS has synchronized successfully.

Table C–11   Parameters of the GetPimRecordVersions API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.

Table C–11   (Cont.) Parameters of the GetPimRecordVersions API

Parameter	Data Type	Type	Description
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the **connector name**. This parameter must be non-null and cannot be empty.
pimRecordIdArray	String []	In	Contains an array of PIM Record IDs for which the connector wants to retrieve the version ID.

Table C–11   (Cont.) Parameters of the GetPimRecordVersions API

Parameter	Data Type	Type	Description
pimRecordVersionIdArray	String []	Out	On output, contains the PIM Record Version ID of each record specified in the pimRecordIdArray. There is a positional relationship between pimRecordIdArray and pimRecordVersionIdArray in that the ith element in pimRecordVersionIdArray contains the Version ID corresponding to the ith element in pimRecordIdArray. The ith element in pimRecordVersionIdArray is a non-null empty string when either of the following conditions are true:
			- The ith element in pimRecordIdArray is not found in the BDSS store. - The ith element in pimRecordIdArray is found in the BDSS store but has a null version, which may occur when a failure is detected in a previous synchronization session. When a synchronization session completes without failures, a cannot be empty Version ID is provided for the ith element in the pimRecordVersionIdArray.
			A connector should echo suppress a PIM record if the GetPimRecordVersions method returns a corresponding Version ID that is identical to the Version ID of the PIM record. As a corollary, a connector should not echo suppress a PIM record if the GetPimRecordVersions method returns the corresponding Version ID as an empty string.
			The ith element in pimRecordVersionIdArray is never returned as null for a corresponding ith element in pimRecordIdArray.

Return Value

Integer:

- 0 for SUCCESS/OK

- Nonzero for ERROR

Exceptions

A RemoteException is thrown if an irrecoverable error occurs.

C.4.2 CacheTempSyncState API

The CacheTempSyncState method saves the given synchronization state to a temporary data location in the BDSS store. Connectors that do not maintain state call this method after extracting the PIM data for the given domain, but before providing the last batch of records to BDSS.

Table C–12   Parameters of the CacheTempSyncState API

Parameter	Data Type	Type	Description
syncSession	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the **connector name**. This parameter must be non-null and cannot be empty.
pimDomainTarget	String	In	Contains the PIM Domain Target. This parameter must be non-null and cannot be empty.
pimDomainSyncState	String	In	Contains the **synchronization state** that is to be cached for the given user, domain, domain target, and server type

Returns Value

Integer:

- 0 for SUCCESS/OK

- Nonzerofor ERROR

Exceptions

A RemoteException is thrown if an irrecoverable error occurs.

C.4.3 CommitCachedSyncState Method API

The CommitCachedSyncState method saves the **synchronization state** that was previously stored in a temporary location in the BDSS store to a permanent location in the BDSS store. It is intended to be used by connectors that cannot maintain state information across BDSS calls.

A connector must have successfully called CacheTempSyncState during the same synchronization session before calling this method. Typically, a stateless connector calls CacheTempSyncState just before completing an extraction of the data for a particular domain. It then calls this method when the Hub calls EndDomainSynchronization, indicating that the synchronization state should be saved. Table C–13 lists the parameters of the CommitCachedSyncState method.

Table C–13   Parameters of the CommitCachedSyncState API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the connector name. This parameter must be non-null and cannot be empty.
pimDomainTarget	String	In	Contains the PIM Domain Target. This parameter must be non-null and cannot be empty.

Return Value

Integer

- 0 for SUCCESS/OK

- Nonzero for ERROR

Exceptions

A `RemoteException` is thrown if an irrecoverable error occurs.

C.4.4 GetConfigurationMetaData API

Connectors call the `GetConfigurationMetaData` API to get profile configuration metadata. A profile can have one or more sections. Each section can contain one or more parameters and each parameter can have a single value. The profile is conceptually equivalent to a Windows `.ini` configuration file. BDSS connectors can define their own profiles to store metadata specific to the connector. Table C–14 lists the parameters of the `GetConfigurationMetaData` method.

Table C–14   Parameters of the GetConfigurationMetaData API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
configMetadataArray	ConfigurationProfile []	In/Out	On input, contains an array of ConfigurationProfile structures populated with profile metadata that is to be obtained. On output, the array is updated to contain the requested profile metadata.

The `GetConfigurationMetaData` API enables a connector to obtain an arbitrary amount of configuration metadata across multiple profiles in a single call. The connector creates and populates the `configMetadataArray` parameter, indicating

what the connector wants to obtain, and the API populates the `configMetadataArray` with the requested configuration (if the configuration exists).

Basically, the connector builds an array of ConfigurationProfile structures and populates them in a manner that scopes which configuration is obtained and the `GetConfigurationMetaData` API finishes populating the structures. Scoping can be done on a profile, session, or parameter basis. For example, if the connector requires an entire section of a profile, it builds a `ConfigurationProfile` structure and specifies only the intended Profile Name and Section Name and adds the `ConfigurationProfile` to the `configMetadataArray`. Similarly, if a connector requires a single parameter of a section of a profile, it builds a `ConfigurationProfile` that specifies the Profile Name, Section Name, and Parameter Name and adds it to the `configMetadataArray`.

Table C–15 outlines some common use cases of how a connector builds the `configMetadataArray` to obtain configuration metadata and how the `GetConfigurationMetaData` API updates the `configMetadataArray`.

Table C–15   ConfigMetadataArray Use Cases

Use Case	Connector Builds configMetadataArray [in]	API Updates configMetadataArray[out]
Obtain a specific parameter value for a particular section in a particular profile Scope is `Profile.Section.Parameter`	Array has a size of 1 and contains a single `ConfigurationProfile` structure that is populated as follows: `ConfigurationProfile`: ■ The profile name member is populated. ■ The `configSectionArray` member is populated with a single `ConfigurationSection` structure. `ConfigurationSection`: ■ The section name member is populated. ■ The `paramNameValueArray` member is populated with a single `KeyValuePair` structure. `KeyValuePair`: ■ The key member is populated to contain the name of the parameter to be obtained.	Updates the `value` member of the `paramNameValueArray` to contain the parameter value. That is, `configMetadataArray[0].configSectionArray[0].paramNameValueArray[0].value`.

Table C–15   (Cont.) ConfigMetadataArray Use Cases

Use Case	Connector Builds configMetadataArray [in]	API Updates configMetadataArray[out]
Obtain parameter names and values of a section in a profile. The scope is `Profile.Section`.	Array has a size of 1 and contains a single `ConfigurationProfile` structure populated as follows: `ConfigurationProfile`: ■ The profile name member is populated. ■ The `configSectionArray` member is populated with a single `ConfigurationSection` structure. `ConfigurationSection`: Array has a size of 1 and contains a single `ConfigurationProfile` structure the is populated as follows: ■ The section name member is populated. ■ `paramNameValueArray` is an empty array.	Updates the `paramNameValueArray` member of the `ConfigurationSection`. (That is, `configMetadataArray[0].configSectionArray[0].paramNameValueArray` is allocated and populated with all the parameters and values in the section.)
Get all parameter names and values of all sections in a profile. The Scope is `Profile`.	An array has a size of 1 and contains a single `ConfigurationProfile` structure populated as follows: `ConfigurationProfile`: ■ The profile name member is populated. ■ The `configSectionArray` member is an empty array. `ConfigurationSection`: Null	The `configSectionArray` is allocated and populated with a `ConfigurationSection` for each section in the profile. Each `ConfigurationSection` is populated with the section name and the `paramNameValueArray` is populated to contain the parameter and values of each configuration option in the section.
Get all parameter names and values of all sections in multiple profiles. The scope is `Multiple Profiles`.	The same builds as the `Profile` scope, but the array size is greater than 1 (that is, there is 1 `ConfigurationProfile` element in the array for each profile requested).	The same builds as the `Profile` scope, but each `ConfigurationProfile` member in the array is updated.

Return Value

Integer:

■ 0 for SUCCESS/OK.

■ Nonzero for ERROR.

The `GetConfigurationMetaData` API returns an error only in cases where it cannot obtain any metadata from the BDSS store because of disruptive circumstances, such as the store not being available or having insufficient memory. It does not return an error if it cannot obtain a particular configuration value, section, or profile. The connector determines whether failure to get a particular configuration metadata value is, in fact, a failure. For example, a connector may want to retrieve the extract response batch size from one of its sections in a connector profile. Assuming that no extraneous events occur, and the customer failed to configure this value, the API returns a SUCCESS/OK error code and leave the corresponding value as it is. (That is, it may be

null or empty, depending on the connector configuration values set for the extract response batch size). At this point, the connector decides the severity of the configuration error and handles it appropriately For example, it could log an error or warning message and use a default value even though it does not consider it an irrecoverable error that stops the connector functioning. It could also log an error or warning and consider it an irrecoverable error.

Exceptions

A `RemoteException` is thrown if an irrecoverable error occurs.

C.4.5 GetDeletesByPendingCreateEchoes API

A connector calls the `GetDeletesByPendingCreateEchoes` API if it cannot natively detect deletions of Hub-created records when the delete operation occurs after the Hub creates the record in the current **synchronization session** and before a subsequent synchronization session after record creation. Table C–16 lists the parameters of the `GetDeletesByPendingCreateEchoes` API.

Table C–16   Parameters of the GetDeletesByPendingCreateEchoes API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the **connector name**. This parameter must be non-null and cannot be empty.
echoIdArray	String []	In	Contains an array of PIM Record IDs, where each identifies an **echo**. This parameter must not be null. The array must have a length >= 1. The Hub ignores any entries that are empty strings or -1, as the PIM Record ID. If a connector has no echoes, it must still provide an array. Oracle recommends an array size of 1 (one) that contains an empty string as the PIM Record ID.
upsertIdArray	String []	In	Contains an array of PIM Record IDs, where each identifies a record that has been updated or created in the PIM since the last Synchronization State. Must not be null. The array must have a length >= 1. The Hub ignores any entries that are empty strings or -1 as the PIM Record ID. If a connector has no **upsert record**s, it must still provide an array with a recommended array size of 1 containing an empty string as the PIM Record ID.
deleteRecordIdArray	String []	Out	On output, contains a PIM Record ID for each record identified as being deleted from the PIM. If there are no delete operations detected, the returned array is null.

Return Value

Integer

- 0 for SUCCESS/OK

- Nonzero for ERROR/NOTOK

Exceptions

A `RemoteException` is thrown if an irrecoverable error occurs.

C.4.6 GetDeletesByIds API

A connector calls the `GetDeletesByIds` method if it cannot natively detect deletions from its PIM server. This method queries the Hub store for a list of PIM Record IDs for the given user, the **Hub domain**, and the connector name that have been synchronized previously. It then compares the Hub store-based array against the connector-provided array to build an array of PIM Record IDs that identify the records that have been deleted. Any PIM Record IDs in the Hub store-based array that are not in the array provided by the connector are considered deletes. Table C–17 lists the parameters of the `GetDeletesByIds` method.

Table C–17   Parameters of the GetDeletesByIds API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the connector name. Must be non-null and cannot be empty.
recordIdArray	String[]	In	An array containing the PIM Record IDs of all PIM records for the given user, domain, and server type. This parameter must not be NULL. If there are no PIM records in the PIM, the connector should provide an array of size 1 containing an empty string as the PIM Record ID.
deleteRecordIdArray	String[]	Out	On output, contains an array of PIM Record IDs identifying records that have been deleted from the PIM. If none are detected, then the array returned is null.

Return Value

Integer:

- 0 for SUCCESS/OK

- Nonzero for ERROR/NOTOK

Exceptions

A `RemoteException` is thrown if an irrecoverable error occurs.

C.4.7 GetRecordAssociatedData API

Connectors call the `GetRecordAssociatedData` API to obtain the PIM Record associated data that is associated with a record from the Hub data store. This is useful for connectors that associate data with PIM records. For example, a connector can associate some data with each PIM record to help facilitate **echo** detection and obtain that data using this API. Table C–18 lists the parameters of the `GetRecordAssociatedData` API.

Table C–18   Parameters of the GetRecordAssociatedData API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. Must be non-null and cannot be empty.
hubDomain	String	In	Contains the **Hub domain**. Must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. Must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. Must be non-null and cannot be empty.
connectorName	String	In	Contains the **connector name**. Must be non-null and cannot be empty.
pimRecordIdArray	String []	Out	An array containing the PIM Record ID of each record whose PIM Record Associated Data is to be retrieved. This parameter must be non-null and have a size > 0. Each element of the array must be non-null and cannot be empty.
pimRecordAssocDataArray	String []	Out	On output, this array contains the PIM Record Associated Data of each record specified in the pimRecordIdArray.
			There is a positional relationship between the elements of pimRecordIdArray and the elements of pimRecordAssocDataArray. The AssociatedData for the ith element in pimRecordIdArray is located in the ith element of pimRecordAssocDataArray. If the ith element in pimRecordIdArray identifies a record that has no PIM Record Associated Data, the ith element in this array contains an empty AssociatedData element (that is, the structure is be allocated but the members of the structure are null).

Table C–18   (Cont.) Parameters of the GetRecordAssociatedData API

Parameter	Data Type	Type	Description
errorArray	String []		On output, an array consisting of success or error codes. There is a positional relationship between this array and pimRecordAssocDataArray. The ith error code corresponds to the success or failure of getting the ith AssociatedData. (0 = OK, 1 = ERROR, 2 = Data does not exist in DB)

Return Value

Integer:

- 0 for SUCCESS/OK. If all of the AssociatedData were read from the data store.

- Nonzero for ERROR/NOTOK. If any of the AssociatedData elements failed to be read. The caller examines errorArray to determine which elements failed to be read and the connector determines the severity of the error. The error codes in the errorArray may contain more explicit error codes.

Exceptions

A RemoteException is thrown if an irrecoverable error occurs.

C.4.8 GetSyncState API

A connector calls the GetSyncState API to obtain connector-defined **synchronization state** associated with a domain. This method only returns the synchronization state stored in the permanent data store location. See CacheTempSyncState API, CommitCachedSyncState Method API, and SaveSyncState API for more information about permanent and temporary Synchronization State. Table C–19 lists the parameters of the GetSyncState API.

Table C–19   Parameters of the GetSyncState API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the **connector name**. This parameter must be non-null and cannot be empty.

Table C–19   (Cont.) Parameters of the GetSyncState API

Parameter	Data Type	Type	Description
pimDomainTarget	String	In	Contains the PIM Domain Target. This parameter must be non-null and cannot be empty.
pimDomainSyncState	String	In	On output, contains the synchronization state for the given user, domain, domain target, and server type. If no synchronization state exists for the given user, domain, connector name, and domain target, the return value is null.

Return Value

Integer

- 0 for SUCCESS/OK

- Nonzero for ERROR/NOTOK

Exceptions

A `RemoteException` is thrown if an irrecoverable error occurs or if there is no synchronization state.

C.4.9 GetUserAssociatedData API

The connector calls the `GetUserAssociatedData` API to obtain user level, connector-specific PIM User Associated Data. The API does not prevent multiple threads from accessing a given user's associated data. Table C–20 lists the parameters of the `GetUserAssociatedData` API.

Table C–20   Parameters of the GetUserAssociatedData API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the **connector name**. This parameter must be non-null and cannot be empty.
assocDataIdArray	String	In	Contains an array of connector-defined user associated data identifiers.

Table C–20   (Cont.) Parameters of the GetUserAssociatedData API

Parameter	Data Type	Type	Description
pimUserAssocDataArray	Associated Data []	Out	On output, contains the user-specific associated data. There is a positional relationship between this array and `assocDataIdArray`. The ith element of `pimUserAssocDataArray` contains the PIM User associated data of the ith element indicated in `assocDataIdArray`.
errArray	Integer []	Out	On output, contains an array of error codes indicating the success or failure of getting the PIM User associated data. The ith element in `errArray` holds the success or failure code of obtaining the ith element indicated in `pimUserAssocDataArray`.0 = OK, 1 = ERROR, 2 = Data does not exist in DB.

Return Value

Integer:

- 0 for SUCCESS/OK. (If all of the `AssociatedData` elements were read from the data store.)

- Nonzero for ERROR/NOTOK. If Any of the `AssociatedData` elements failed to be read. The caller examines `errArray` to determine which elements failed to be read, and the connector determines the severity of the error. The error codes in the `errArray` may contain more explicit error codes.

Exceptions

A `RemoteException` is thrown if an irrecoverable error occurs.

C.4.10 SaveSyncState API

Connectors that maintain state across BDSS calls must call this API to save the connector-defined **synchronization state** associated with a domain to the Hub store. Connectors that do not maintain state must use `CacheTempSyncState` and `CommitCachedSyncState` to save the synchronization state. Table C–21 lists the parameters of the `SaveSyncState` API.

Table C–21   Parameter of the SaveSyncState API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.

Table C–21   (Cont.) Parameter of the SaveSyncState API

Parameter	Data Type	Type	Description
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the **connector name**. This parameter must be non-null and cannot be empty.
pimDomainTarget	String	In	Contains PIM Domain Target. This parameter must be non-null and cannot be empty.
pimDomainSyncState	String	In	Contains the synchronization state for the given user, domain, domain target, and server type. This parameter must be non-null and cannot be empty.

Return Value

Integer:

- 0 for SUCCESS/OK

- Nonzero for ERROR/NOTOK

Exceptions

A `RemoteException` is thrown if an irrecoverable error occurs.

C.4.11 SetRecordAssociatedData

Connectors call the `setRecordAssociatedData` API to associate the connector-defined PIM Record associated data to the given record. Table C–22 lists the parameters of the `SetRecordAssociatedData` API.

Table C–22   Parameters of the SetRecordAssociatedData API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.

Table C–22   (Cont.) Parameters of the SetRecordAssociatedData API

Parameter	Data Type	Type	Description
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the **connector name**. This parameter must be non-null and cannot be empty.
pimRecordIdArray	String []	In	An array of PIM Record IDs of those records whose PIM record associated data is to be written to the BDSS data store.
pimRecordAssocDataArray	Associated Data []	In	An array containing PIM Record Associated Data of each record to be written to the BDSS data store. There is a positional relationship between the `pimRecordIdArray` and `pimRecordAssocDataArray`. The ith PIM Record ID in `pimRecordIdArray` must have its PIM Record Associated Data located as the ith `AssociatedData` in the `pimRecordAssocDataArray`.
errorArray	Integer []	Out	On output, an array error codes. There is a positional relationship between this array and `pimRecordAssocDataArray`. The ith error code corresponds to the successor failure of setting the ith `AssociatedData.0 = OK`, `1 = ERROR`

Return Value

Integer

- 0 for SUCCESS/OK. If all of the `AssociatedData` elements were written to the BDSS data store.

- Nonzero for ERROR/NOTOK. If any of the `AssociatedData` elements failed to be written. Caller examines `errorArray` to determine which elements failed to be written and the connector determines the severity of the error.

Exceptions

A `RemoteException` is thrown if an irrecoverable error occurs.

C.4.12 SetUserAssociatedData API

Connectors call the `SetUserAssociatedData` API to save user-level, connector-specific PIM User Associated Data. The API does not prevent multiple threads from accessing a given user's associated data. Table C–23 lists the parameters of the `SetUserAssociatedData` API.

Table C–23   Parameters of the SetUserAssociatedData API

Parameter	Data Type	Type	Description
syncSessionId	String	In	Contains the Synchronization Session ID. This parameter must be non-null and cannot be empty.
hubDomain	String	In	Contains the **Hub domain**. This parameter must be non-null and cannot be empty.
hubUserId	String	In	Contains the **Hub User ID**. This parameter must be non-null and cannot be empty.
pimUserId	String	In	Contains the **PIM User ID**. This parameter must be non-null and cannot be empty.
connectorName	String	In	Contains the **connector name**. This parameter must be non-null and cannot be empty.
assocDataIdArray	String []	In	An array of connector-defined users associated data identifiers.
pimUserAssocDataArray	Associated Data	In	Contains the user-specific associated data. There is a positional relationship between pimUserAssocDataArray and assocDataIdArray. The ith element of assocDataIdArray has the corresponding PIM User associated data as the ith element in this array.
errArray	Integer []	Out	On output, contains an array of error codes indicating the success or failure of setting (persisting to the BDSS store) the user associated data.There is a positional relationship between this array and assocDataIdArray and pimUserAssocDataArray. The ith element in this array holds an error code for setting the ith element of pimUserAssocDataArray.0 = OK, 1 = ERROR.

Return Value

Integer:

- 0 for SUCCESS/OK. If all of the AssociatedData elements were written to the BDSS data store.

- Nonzero for ERROR/NOTOK. If any of the AssociatedData elements failed to be written. The caller examines errArray to determine which elements failed to be written, and the connector determines the severity of the error.

Exceptions

A RemoteException is thrown if an irrecoverable error occurs.

C.5 Data Representation Structures

This section describes the structure of data that is passed between BDSS and the connectors.

C.5.1 Common Structures

The common structures include the `KeyValuePair` structure.

C.5.1.1 KeyValuePair

The `KeyValuePair` is a basic structure used to hold metadata in a key/value structure. It is used by many other data structures defined for the connector. Table C–24 lists the members of the `KeyValuePair` structure.

Table C–24   Members of the KeyValuePair Structure

Member	Data Type	Description
key	String	Holds the *key* value of a key/value pair.
value	String	Holds the *value* value of a key/value pair.

C.5.2 Connector Interface-Related Structures for the BDSS Engine

The connector interface-related structures include:

- DomainInfo
- ExtractRequest
- ExtractResponseData
- ExtractResponseMetaData
- HubContext
- PimRecordDescription
- AssociatedData
- AssocDataElement
- RecordMetaData
- UpsertRecord
- Enum ExtractResultCode

C.5.2.1 DomainInfo

The `DomainInfo` structure contains the domain information required by a connector to perform an initialization for a user for a **synchronization session**. This structure is used only when the Hub calls the connector `InitializeUserSyncSession` API. Table C–25 lists the members of the `DomainInfo` structure.

Table C–25   Members of the DomainInfo Data Structure

Member	Data Type	Description
pimDomainTarget	String	Contains the PIM Domain Target. Must be non-null. May be empty. For more information about this member, see Section C.1.5, "PIM Domain Target."
pimFilterCondition	String	Contains a PIM filter condition that is applied to the domain extract that filters out records that should not be included in the extract. This member must be non-null. It may be empty.
hubDomain	String	Identifies the **Hub domain**. This member must be non-null and cannot be empty.
pimDomainSyncState	String	Contains the PIM **synchronization state** for the domain. This member must be non-null. This member may be empty.

C.5.2.2 ExtractRequest

The ExtractRequest structure is passed to the connector when BDSS begins an extraction for a user through the ExtractDomains API. Table C–26 lists the members of the ExtractRequest structure.

Table C–26   Members of the ExtractRequest Data Structure

Member	Data Types	Description
hubUserId	String	Contains the **Hub User ID**. This member must be non-null and cannot be empty.
pimUserId	String	Contains the **PIM User ID**. This member must be non-null and cannot be empty.
hubDomainArray	String []	An array of Hub domains. This array must be non-null and be sized >=1. Elements in the array must be non-null and cannot be empty.

C.5.2.3 ExtractResponse

The ExtractResponse structure is used by the connector to propagate record data and status information to the Engine through the Engine Callback interface's ExtractDomainResults callback method. Table C–27 lists the members of the ExtractResponse structure.

Table C–27   Members of the ExtractResponse Data Structure

Member	Data Types	Description
extractResponseMetaData	extractResponseMetaData	Contains metadata about the extract response. This member must be non-null and cannot be empty.
extractResponseData	extractResponseData	Contains the extracted PIM records. This member must be non-null and cannot be empty.

C.5.2.4 ExtractResponseData

The `ExtractResponseData` structure contains records extracted from a PIM server. Table C–28 lists the members of the `ExtractResponseData` structure.

Table C–28   Members of the ExtractResponseData Data Structure

Member	Data Type	Description
hasUpserts	Boolean	Indicates if the structure has an **upsert record**. This array has valid UpsertRecord structures.
upsertRecordArray	UpsertRecord []	An array of UpsertRecord structures. Each element in the array represents a single PIM record that has been updated or created in the PIM since the last **synchronization session**.
		This array must be non-null and cannot be empty. If there are no modified records, the array should have a length of 1 and a non-null UpsertRecord containing non-null members. However, the members may have empty strings as members.
		This array is ignored if `hasUpserts` is false.
hasDeletes	Boolean	Indicates whether `deleteRecordIdArray` has valid string members
deleteRecordIdArray	String []	An array of strings where each element of the array contains the PIM Record ID of a PIM record that has been deleted since the last synchronization session
		Must be non-null and cannot be empty. If there are no deleted records, the array should have a length of 1 and a non-null PIM Record ID that can be empty.
		This array is ignored if `hasDeletes` is false.

C.5.2.5 ExtractResponseMetaData

The `ExtractResponseMetaData` structure contains metadata information about a given `ExtractResponseMetaData` of a domain extraction. Table C–29 lists the members of the `ExtractResponseMetaData` structure.

Table C–29   Members of the ExtractResponseMetaData Data Structure

Member	Data Type	Description
syncSessionId	String	Contains the Synchronization Session ID. This member must be non-null and cannot be empty.

Table C–29   (Cont.) Members of the ExtractResponseMetaData Data Structure

Member	Data Type	Description
hubDomain	String	Contains the **Hub domain**. This member must be non-null and cannot be empty.
hubUserId	String	Contains the **Hub User ID**. This member must be non-null and cannot be empty.
resultCode	String	Contains one of the ExtractResultCode enumeration values.
extractErrorDescription	String	Contains a description of why an extraction failed. This member must be non-null, but may be empty. It should only be populated when the resultCode is an error code.
pimDomain	String	Contains the PIM Domain. This member must be non-null and cannot be empty.
pimUserId	String	Contains the **PIM User ID**. This member must be non-null and cannot be empty.
connectorName	String	Contains the **connector name**
echoesSuppressed	String	Indicates whether the ExtractResponse had echoes removed
sequenceNumber	String	Indicates a sequence number of the ExtractResponse. For example, when the connector sends the first ExtractResponse to the Hub, the sequenceNumber is 1. When it sends the second ExtractResponse, the sequenceNumber is 2, and so on. The sequenceNumber starts at 1 for each domain, user and **synchronization session**. For Oracle Fusion Middleware 11*g* release 1 of BDSS, the connector Extract Response Batch Size configuration should be large enough to allow that only a single ExtractResponse is sent to the Hub for a **Hub domain**.

C.5.2.6 HubContext

The `HubContext` structure (described in Table C–30) provides the connector with information about which BDSS instance should be called when calling the Run-Time Library or the Engine Callback interfaces.

Table C–30   Members of the HubContext Data Structure

Member	Data Type	Description
engineEndPointURL	String	Contains the URL of the server hosting the BDSS Web service components that the connectors call. This member must be non-null and cannot be empty.
syncSessionId	String	Contains a session ID. This member must be non-null and cannot be empty.
connectorRuntimeLibraryURL	String	Contains the URL used by a connector when it makes calls to the Connector Run-Time Library to read or write metadata.
connectorName	String	Contains the **connector name**. This member must be non-null and cannot be empty.

C.5.2.7 PimRecordDescription

The `PimRecordDescription` structure (Table C–31) contains the PIM Record Description. Connectors must build these descriptions for every **upsert record** of a given domain if the `KeyField` metadata is configured for the domain.

Table C–31   Members of the PimRecordDescription Data Structure

Member	Data Type	Description
descriptionArray	KeyValuePair []	An array of KeyValuePair structures containing the Hub field names and Hub field values that comprise the PIM Record Description. The array must not be null.
		■ If `KeyFields` are configured for the domain, the array must have a size equal to the number of configured `KeyFields` and `hasDescription` should be set to `true`. Each `KeyValuePair` in the array must be non-null. Each key must be non-null and cannot be empty. Each value must be non-null but may be empty.
		■ If `KeyFields` are not configured for the domain, the array must have a size equal to 1 and `hasDescription` should be set to `false`. The array should contain a non-null `KeyValuePair`, and the key and value must both be non-null but may both be empty.

Table C–31   (Cont.) Members of the PimRecordDescription Data Structure

Member	Data Type	Description
hasDescription	Boolean	A Boolean value that indicates whether the descriptionArray has any valid elements. If the flag is set to false, then the given descriptionArray is ignored.

C.5.2.8 AssociatedData

The AssociatedData structure (Table C–32) enables connectors to associate data with either a PIM record or a PIM user.

Table C–32   Members of the AssociatedData Data Structure

Member	Data Type	Description
assocDataId	String	Contains a connector-defined identifier that uniquely identifies the associated data. This member must be non-null. It may be empty if hasAssocData is false. If hasAssocData is true, must be cannot be empty.
hasAssocData	Boolean	A Boolean value that indicates if assocDataArray has valid and cannot be empty AssocDataElement members
assocDataArray	AssocDataElement []	An array of AssocDataElement structures defining the group (or groups) of associated data. This member must be non-null.
		This member must have size >= 1.
		If hasAssocData is false, the array should have a size of 1 and contain an empty AssocDataElement.

C.5.2.9 AssocDataElement

The AssocDataElement data structure contains a group of associated data. Table C–33 lists the members of the AssocDataElement structure.

Table C–33   Members of the AssocDataElement Data Structure

Member	Data Type	Description
groupName	String	A string identifier that groups associated data contained in groupAssocDataArray. This member must be non-null. This member may be empty if hasAssocData is false. This member must be unique within a given AssocData.assocDataArray.
hasAssocData	Boolean	A Boolean value that indicates if groupAssocDataArray has valid and cannot be empty KeyValuePair members.
groupAssocDataArray	KeyValuePair []	An arbitrarily large array of KeyValuePair structures representing a single piece of associated data within a group. This member must be non-null. Must have a size of >= 1. If hasAssocData is false, the array should have a size of 1 and contain an empty string for both the key and value. The array can be ignored by the recipient. If hasAssocData is true, then each element must have a non-null and cannot be empty key and value.

> **Note:** If the associated data is binary, the connectors can encode the data to base-64 to save it in the database as a non-BLOB. In such a case, it is the connectors' responsibility to chunk and unchunk the associated data.

C.5.2.10 RecordMetaData

The RecordMetaData structure contains the metadata about a PIM record extracted from, or pushed to, a PIM server. Table C–34 lists the members of the RecordMetaData structure.

Table C–34   Members of the RecordMetaData Data Structure

Member	Data Type	Description
recordId	String	Contains the PIM Record ID. This member must be non-null and cannot be empty.
versionId	String	Contains the PIM Record Version ID. This member must be non-null and cannot be empty.

Table C–34 (Cont.) Members of the RecordMetaData Data Structure

Member	Data Type	Description
recordError	Boolean	A Boolean value that indicates if record processing failed. If this value is set to true, then both the recordId and the recordErrorDescription must be provided.
recordErrorDescription	String	Describes why an error occurred during the processing of an UpsertRecord. This member must be non-null, but it can be an empty string. It must be populated if recordError is set to true.
assocData	AssociatedData	Contains the AssociatedData.
pimRecordDescription	PimRecordDescription	Contains the PIM Record Description. May be null if the KeyField data is not configured for the domain. This member must be null if the KeyField data is configured for the domain.

C.5.2.11 UpsertRecord

The UpsertRecord structure defines the structure of a Hub record extracted from, or pushed to, a PIM server. Deleted Hub records are not represented by this structure. Table C–35 lists the members of the UpsertRecord structure.

Table C–35 Members of the UpsertRecord Data Structure

Member	Data Type	Description
pimRecordMetadata	RecordMetaData	Contains metadata related to the record. This member must be non-null.
hubRecordData	String	Contains an XML representation of a Hub record. This member must be non-null and cannot be empty.

C.5.2.12 Enum ExtractResultCode

The enumeration type that defines the result codes of an extract request. Example C–1 illustrates the Enum definition.

Example C–1 Enum Definition

```
Enum ExtractResultCode
{
FULL_EXTRACT_COMPLETED_OK,
FULL_EXTRACT_COMPLETED_NOTOK,
PARTIAL_EXTRACT_COMPLETED_OK,
```

```
PARTIAL_EXTRACT_COMPLETED_NOTOK
};
```

C.5.3 Configuration-Related Structures

The configuration structures include:

- ConfigurationSection

- ConfigurationProfile

C.5.3.1 ConfigurationSection

The ConfigurationSection structure represents a section of a profile that is conceptually the equivalent of a configuration file or an .ini configuration file. That is, a profile contains a collection of uniquely named sections, each containing a collection of parameters and values. Table C–36 lists the members of the ConfigurationSection structure.

Table C–36   Members of the ConfigurationSection Configuration Structure

Member	Data Type	Description
sectionName	String	Identifies the section name contained in a profile
paramNameValueArray	KeyValuePair []	Contains the configuration parameters and values contained in a section of a profile

C.5.3.2 ConfigurationProfile

The ConfigurationProfile structure represents a profile. Table C–37 lists the members of the ConfigurationProfile structure.

Table C–37   Members of the ConfigurationProfile Configuration Structure

Member	Data Type	Description
profileName	String	Identifies the name of a configuration profile
configSectionArray	ConfigurationSection []	Contains an array of ConfigurationSection structures containing information about a section of a profile

C.6 Best Practices for Building Connectors

Developers should divide a connector into at least three packages: the Hub Transport Package, the PIM Transport Package, and the Transformation Package.

- Hub Transport Package

  The Hub Transport Package contains the components that communicate directly with the Hub. While these components handle all of the details related to communicating with the Hub, they do not perform direct data exchange operations against the PIM store or server. Instead, they communicate with a PIM transport package to exchange data with the PIM store. This component is generic in that it can be reused to communicate with one or more PIM transport packages.

- PIM Transport Package

  The PIM Transport package contains the components that communicate directly with the PIM server and PIM store. These components never communicate directly with the Hub, but instead handle all of the PIM-related details. There should be a PIM Transport API that facilitates communication between the Hub and PIM Transport Packages.

- Transformation Package

  The Transformation Package contains the components that perform translations of record data to and from the Hub format to the PIM format.

For more information, see Section 2.2, "Overview of Connectors."

C.7 The Exchange 2007 Connector API

This section uses the following to illustrate how the Exchange 2007 Connector works:

- A BPEL system that has a users *cjones* and *tsmythe*.

- An Exchange system that has a user *chuck.jones@sample.com* and *tsmythe@sample.com*.

- A Hub user that maps the *cjones* user to the *chuck.jones@sample.com* user.

- A user outside of the BPEL server and Exchange organization that has a e-mail address of *john.smith@otherdomain.com*.

C.8 Generic Component Support for Calendar Synchronization

Similar to the goals for calendar support in the Hub, the goal for the generic connector components is that they should be able to support calendar synchronization in the same manner that they synchronize other domains. That is, the record data of any domain is irrelevant to the generic components. In addition, the generic components should not implement any domain-specific logic.

The Hub Transport provides delete detection for PIM Transports that are unable to detect deletes. If a PIM Transport component cannot natively detect deletes, then the Hub Transport should invoke a new PIM Transport Interface method after it determines the list of deleted record IDs so that the PIM Transport can determine if a deleted record should appear as a delete in the extract response, or as a exception to a calendar record that is a fanned instance. In the former case, the record ID would remain in the delete list. In the latter case, the record ID would be removed from the delete list and the connector must update the appropriate upsert record so that the master VEVENT contains the start date of the deleted occurrence in the EXDATE list. While the Hub transport might perform this for the Calendar domain, it could do so for every domain, thereby giving PIM Transports the opportunity to modify the Extract Response that ultimately gets sent to the Hub. Such methods as `PIMExtractResponse, FinalizeExtractResponse (PIMExtractResponse response)` would be an appropriate signature.

Use **reflective synchronization** to test calendar synchronization. The PIM Transport Interface would need a new method that returns the connector type. The Hub Transport would call this method at some point before reading configuration profiles from BDSS and it would load a profile whose name corresponds to the return value from the new PIM Transport method instead of the profile name corresponding to the connector name.

C.8.1 Calendar Support Methods

This section describes methods of the Run-Time Library (RTL).

C.8.1.1 Hub Contract

The Hub contract uses the following methods and capabilities:

- GetAssocDataIdOfAssociatedData

- Attendee Support

- AttendeeMetadata

C.8.1.1.1   GetAssocDataIdOfAssociatedData This method, one included in the Run-Time Library, allows a connector to obtain the associated data ID from an associated data value. The method is similar to the `getUserAssociatedData` and `getRecordAssociatedData` in that a connector can provide a collection of associated data values and get a corresponding collection of `assocdata` IDs along with a status code indicating success, failure, or not-in-database for each requested `assocDataID`.

For example, if the connector stores `assocData` in BDSS having an `assocDataId` of *MyCalendarAssocDataZ* having a group name of *MyGroup* containing a collection of associated data key values pairs of {Key1,Value1}, {Key2,Value2},{Key3,Value3}, and the connector wants to later get the value *MyCalendarAssocDataZ*, it can call this API specifying *MyGroup* and any or all of {Key1,Value1}, {Key2,Value2} or {Key3,Value3}. The API queries the assoc data table for a row having the specified group name and key/value pairs to obtain the ID. The API must ensure to account for scenarios where the assoc data value is segmented across multiple rows.

C.8.1.1.2   Attendee Support The Hub provides support for **attendee resolution** when it calls the connector methods for creating or updating a calendar record by using its user map metadata to provide the PIM User IDs in terms of the targeted system as part of the record metadata that accompanies a CREATE/UPDATE request. For the Hub to provide this metadata during a CREATE/UPDATE operation, a source connector must provide the attendees, in terms of the source system, of a calendar record in an extract response. The BDSS Connector RecordMetadata has a data member that holds the attendee metadata.

For example, when the Hub extracts *cjones'* calendar from a BPEl server, the BPEL Task Connector would specify *cjones* in the record metadata associated with the **upsert record**. The Hub then determines that the record must be created in the *chuck.jones@sample.com* user's Exchange calendar. The Hub performs a lookup for *cjones* in its user map and find that *cjones* is mapped to *chuck.jones@sample.com* in Exchange. Because the users are mapped, the Hub would provide *chuck.jones@sample.com* instead of *cjones* in the record metadata associated with the upsert record that the Hub provides when it calls the Exchange 2007 Connector to create the record.

To support this, the BDSS Connector RecordMetadata must be updated to have a data member of type AttendeeMetadata to hold attendee metadata.

C.8.1.1.3   AttendeeMetadata `AttendeeMetada` is a WSDL construct in the BDSS WSDL for the connector interface. The class manages attendee information associated with a Hub calendar record.

Given that:

- A recurring calendar meeting can have exceptions containing different attendees than the master record or other exceptions

- A target connector that creates or updates a calendar cannot use the attendee information contained within the ICAL (because the attendee information in the ICAL is defined in terms of the source connector)

This class provides a target connector with the ability to determine the attendees associated with each occurrence of the meeting. This implies a map-of-collections construct whereby the map keys off the start date of an occurrence and the collection for a given key contains the attendees associated with the occurrence. If the map has a single entry, this would imply a calendar record with no exceptions. If the map has multiple entries, this would imply a calendar record that has exceptions whereby the exception contains a different set of attendees than the master record.

C.8.1.2 Connector Contract

The following sections define the contractual obligations a connector implementation must adhere to synchronize calendar records using BDSS. Connectors must also adhere to the contractual obligations defined in the connector interface document.

C.8.1.2.1 Federated versus Non-federated If the calendar system with which the connector exchanges data is a **federated PIM**, then the connector must abstract the federation model away from the Hub. For example, if a single calendar meeting is similar to the Siebel model and stores calendar records in a relational database such that each record has a unique row ID and that visibility to the calendar record is achieved by adding a foreign key from the calendar record to a user table. When the connector extracts a given calendar record, it can fabricate a PIM Record ID that is the concatenation of the row ID of the Calendar record and the row ID of the attendee for each attendee when each is synchronized. In a subsequent synchronization session, if the connector is directed by BDSS to perform UPDATE or DELETE operations, the Hub would provide the fabricated ID and the connector can parse out the two row IDs and then perform the appropriate operation.

Connectors of a federated system may be required to translate a given BDSS request to a different operation. For instance, assume that the calendar model of the system is such that attendees can only modify meetings that they organize. If BDSS directs the connector to perform an Update operation, but the connector determines that the row ID of the synchronizing user is an attendee and not an organizer, the connector may be able to simply no-op the BDSS call and perhaps log a message. That is, a no-op might be appropriate if the connector determines that the organizer is sync-enabled because the change, if any, would synchronize when the organizer synchronizes. (If the source is a **non-federated PIM**, then the organizer may not of changed his or her copy of the meeting). A no-op might not be appropriate if the connector determines that the organizer is enabled for synchronization (either because the user is not defined in the organization or the user simply is not added to the BDSS user map for synchronization). Similarly, if BDSS directs the connector to delete a record and the connector determines that an attendee deleted the record, then the connector would remove the attendee from the calendar record instead of deleting the entire record (that is, delete the foreign key identifying the attendee user from the calendar record instead of deleting the entire calendar record). If the connector gets a delete from an organizer, then the calendar record itself could be deleted. Because the order in which users synchronize is not guaranteed, the connector could no-op future deletes if it receives a delete for an attendee sometime after the organizers' delete was synchronized.

It is a connector implementation detail regarding how the abstracting the federation is done, but the implementation must ensure the following behavior:

- The connector should send a delete for each attendee that has been removed from a meeting when each attendee user synchronizes. For example, if Record A has User A and User B as the organizer and attendee, respectively, and User C was removed, then a delete should be in the extract response when User A synchronizes and a delete should be in the extract response when User B synchronizes.

- If the calendar record is deleted, as opposed to a deleted attendee, then a delete should be included in the extract for the organizer and each attendee.

- If the record is converted from a meeting (with attendees) to an appointment (no attendees), then the connector should provide an upsert for the remaining user associated with the calendar record (may or may not be the organizer, depending on the calendar model in the PIM) and a delete for the remaining user(s).

- The connector should provide an upsert record for each user associated with a calendar record on initial extractions or if the record is updated in ongoing extractions. For example, if Record A has User A and User B as the organizer and attendee, then Record A must be provided in the extract response to the Hub when User A synchronizes for the first time and when User B synchronizes for the first time. Additionally, if Record A is ever updated by a user, it must be provided when User A synchronizes and when User B synchronizes. If the update to Record A was a simple addition of a new attendee, then User C, then Record A must be provided in the extract response for User A, User B and User C.

- The connector should not propagate an **echo** for updates done through BDSS by a particular user, but should propagate an update for any other users associated with the record. For example, suppose Record A in the federated system has User A, User B and User C as the attendees and the record is synchronized during Session 1. Then suppose that during Session 2, the connector receives a `deleteRecord` call to delete User C. The connector would remove User C from the calendar record instead of removing the calendar record. During Session 3, a upsert should appear in the extract responses of User A and User B, indicating the new attendee list (User C is no longer attending). When User C synchronizes, the extract response sent to the Hub should not have a delete for the record because this delete would be considered a delete-echo. In this way, if the attendee mapped to User C deletes the record from the other system, BDSS ensures that the attendees mapped to User A and User B in the other system have their records updated to indicate User C is no longer attending the meeting.

C.8.1.2.2 Attendee Resolution Even though the Hub provides attendee resolution support, connector implementations should account for the following scenarios:

- Use Case 1: The user is not configured in BDSS for synchronization. For example, suppose a calendar record originating from a BPEL server has an attendee named *tsmythe* and that *tsmythe* has a valid BPEL server and Exchange account, but *tsmythe* does not want his BPEL data to synchronize through BDSS. The record is then extracted by the BPEL Task Connector. When *cjones* synchronizes, it would specify *cjones* and *tsmythe* as attendees in the record metadata. The Hub would be able to provide the Exchange 2007 Connector with *chuck.jones@sample.com*, but would not be able to provide an Exchange user ID for *tsmythe* in the record metadata when calling the Exchange 2007 Connector to create the record.

- Use Case 2: The user is not recognized as someone within the organization. For example, suppose an Exchange user at Comany X, *john.smith@otherdomain.com*, organizes a meeting and invites *chuck.jones@sample.com* who accepts the meeting and, therefore, has a calendar item in his Exchange folder. Further, suppose *john.smith@otherdomain.com* is not defined in the Exchange and BPEL systems.

When BDSS synchronizes the Hub user that maps *cjones* to *chuck.jones@sample.com*, the Exchange 2007 Connector would not be able to resolve *john.smith@otherdomain.com* as a user within the Exchange organization. It would specify *john.smith@otherdomain.com* in the record metadata. When the Hub then pushes the record to the BPEL server, the BPEL Task Connector would also not be able to resolve *john.smith@otherdomain.com* to a BPEL user.

- Use Case 3: The attendee information contained in the record metadata may not be sufficient for a receiving connector when creating or updating the record in the target system. In this case, the connector should use the provided information to attempt to resolve the user. For example, suppose when the Exchange connector creates the calendar record, it has to specify the attendees First Name, Last Name, and the users e-mail address. The connector could specify the e-mail address contained in the record metadata in a query to Active Directory whereby the query returns the First Name and Last Name of the user with a matching e-mail address.

If a connector cannot resolve a given attendee to a user in the connectors' system, then the connector must ensure to provide the unresolved attendee back to the Hub during subsequent extractions. This is to ensure the attendee is not removed from the meeting when the record update is pushed to another connector.

The options that allow the connector to restore unresolved attendees associated with a calendar record when extracting a given calendar record include:

- Extend the BDSS data model to have a table that would hold attendee metadata and define a new Run-Time Library API that would allow a connector to set or get the metadata.

- Use the record-level or user-level associated data features of BDSS. If record-level associated data is used, there is the potential of having many duplicate unresolved attendees in the BDSS store. Consequently, user-level associated data might be a better option. If user-level associated data is used, there is still the potential of having duplicate, unresolved attendees in the BDSS store, but the duplicates would be less when compared to using record-level associated data. To avoid duplicates, connectors may create a special BDSS PIM User whose sole purpose is to hold unresolved attendees.

- Write the unresolvable attendees to the PIM record itself at CREATE/UPDATE time so that when the record is subsequently extracted, the connector can obtain the unresolved attendees. If the system that a connector exchanges data with allows ambiguous values for attendees, the connector can simply write the unresolvable attendees to the attendee field(s). If, however, the system requires the attendee to be unambiguous, the connector should write the unresolvable attendees to a custom field that is not visible by default to end users.

> **Note:** Maintaining an in-memory cache of unresolved attendees may not be a viable option because a given user may be synchronized by different connector instances of the same connector type. For example, suppose there are two instances of the Exchange 2007 Connector known as Instance 1 and Instance 2 that are deployed on different servers. During synchronization Session 1, *chuck.jones@sample.com* is synchronized through Instance 1. Because *john.smith@otherdomain.com* was not resolvable, the connector decides not to include this attendee when it creates the Exchange record. When an update is made to the meeting and BDSS subsequently performs another synchronization session, *chuck.jones@sample.com* is synchronized through Instance 2. During this session, Instance 2 would have no simple way of restoring *john.smith@otherdomain.com* when the record is extracted (assuming connector instances do not communicate with each other).

Connectors that are unable to persist the unresolvable attendees on the system record itself must either use a BDSS solution (record- or user- associated data) or devise their own method.

If an attendee to a meeting identifies a group of attendees, or Distribution List, the connector must not provide the Distribution List as the attendee in the ICAL. Instead, the connector should provide each member of the group in the ICAL. This distribution list expansion is necessary because a group of attendees in one system may or may not be defined in another system and there is no guarantee that group membership would be the same in cases where a group name or Distribution List name matches across systems. The connector must also ensure to expand the members of all embedded group or Distribution Lists and ensure that no duplicates are contained in the ICAL. For example, if:

- "Distribution List A" has members User1 and User2

- "Distribution List B" has members Distribution List A, User1, User2, and User3

- A calendar record has attendees listed as:
  - "Distribution List A"
  - "Distribution List B"
  - "User3"

then the ICAL should have User 1, User 2 and User 3 only, even though the calendar record has duplicates because of the group membership.

C.8.1.2.3 Recurrence Patterns Connectors must attempt to support all ICAL recurrence patterns.

During extraction, a connector should convert a recurrence pattern from its native definition(s) to some ICAL equivalent. Similarly, when a connector creates or updates a calendar record, it should convert the ICAL recurrence pattern to an equivalent native pattern.

In some cases, however, a connector may not be able to natively support a given ICAL recurrence pattern. In these cases, the connector should still attempt to satisfy the create/update request made by the Hub. The primary means of resolving this is fanning. This form of fanning, not to be confused with non-federated systems, is the process creating multiple calendar records in a manner that mimics the original pattern.

Connectors should always fan a record to a set of non-recurring calendar records such that each instance has a start date that occurs on a date that the recurrence pattern defines. The connector is responsible for abstracting such operations from the Hub.

> **Note:** For Oracle Fusion Middleware 11*g* release 1, BDSS does not support EXRULE.

C.8.1.3 Fanning

The following sections provide information on connector contracts related to fanning.

C.8.1.3.1 Extraction When the connector extracts a calendar record from its system, it must construct a ICAL representation of the record that conforms to the ICAL RFC.

A connector that has previously fanned a recurrence pattern must coalesce all of the fanned instances into a single ICAL representation containing the original recurrence pattern when the record is subsequently synchronized. When reconstructing the recurrence pattern from fanned records connector implementations should:

1. Recognize that a change to a fanned instance in the PIM translates to an exception in the ICAL and should be represented as a separate VEVENT per the requirement listed in the Calendar Hub Domain Schema section.

2. A deletion of any fanned instance in the PIM translates to a EXDATE of the master VEVENT

3. Fanned instances should be coalesced in to a single ICAL message and the `ExtractResponse` should be such that fanned instances are not a part of the response. Additionally, the connector must provide an appropriate PIM Record Version.

4. If fanned instances are updated by the PIM User through a client such that a recurrence pattern is applied to it, the connector must synchronize the fanned, but now recurring, record as a separate record and specify a EXDATE on the original record when performing the extract. It should also account for the use case whereby such an extracted record is in conflict with a corresponding record from another system and looses the conflict. In such a case, the connector would delete the recurring appointment and replace it with the winning record data.

5. Metadata about each fanned instance should be maintained so that when the extraction is performed the connector can:
   - Determine the set of fanned instances
   - Determine how each fanned instance should appear in the ICAL (that is, master VEVENT, exception VEVENT). This implies the connector must track the state of each fanned instance across multiple synchronization sessions. For instance, suppose there are three fanned records that were synchronized during Session 1 (that is, Record A, Record B and Record C). All of these records must be coalesced in to a single ICAL having some recurrence pattern and the same meeting ID. Then if an update to Record A and a delete of Record C are synchronized during Session 2, Record A would appear in the ICAL as a exception VEVENT per Step 1), Record C should appear in the master VEVENT as a EXDATE per Step 2). During Session 2, suppose Record B is in the extract response and both Record A and Record C are not, the connector would add Record B as an exception VEVENT and must also provide Record A as an exception VEVENT and a EXDATE in the master VEVENT for Record C even though Record A and Record C are not in the extract response.

As mentioned, the connector must store metadata about the record and each fanned instance to aid in the reconstruction of a recurrence pattern. The connector may store metadata both in the BDSS data store as associated data, or in hidden fields on the PIM record. What the connector stores is a connector implementation detail, one that is not necessarily the same across all connectors.

If the connector stores record associated data in the BDSS data store and the Run-Time Library returns an error and the individual status codes indicates that the failure occurred because the data was not in the database for all the requested associated data, and that the synchronization state provided is empty/null, then the user has been re-initialized (if the Run-Time Library returns an error and there are individual status code indicating success and some indicating failure due to not being in the database, then this is an irrecoverable error). The connector should account for this use case so it can coalesce the fanned occurrences when performing the initial extract - perhaps by matching records by the Meeting ID.

C.8.1.3.2 CreateRecord When the Hub calls `CreateRecord` the connector must examine the received ICAL to determine whether the record must be fanned. It should first examine each RRULE, RDATE, and EXDATE across all VEVENTs to determine the final recurrence pattern. If the final recurrence pattern cannot be natively represented, then the connector should fan the record and store whatever metadata it requires to restore the pattern in subsequent sessions.

If a failure occurs when creating any fanned instance, the connector should stop creating the fanned instances and delete any successfully created fanned instance related to the record being fanned and return an error to the Hub. The connector would probably want to log messages (note that such create-then-update in the same session would not be reflected in a subsequent session because both operations occur in the same session).

C.8.1.3.3 UpdateRecord When the Hub calls `UpdateRecord`, the connector must determine if the record being updated has been fanned previously. If the record has been fanned previously, it must apply the received ICAL to the fanned instances, potentially having to re-fan the received pattern (if a new pattern was applied to the record and it cannot be natively represented). If re-fanning is done, the connector may have to first delete the old fanned instances and then create the new instances. The connector must also suppress these deletes in the subsequent extraction.

If the connector gets an update to a record it previously fanned and the ICAL has a exception VEVENT whose start date was not within the fanning limit of the original fanning, the connector must create the exception in the PIM.

The connector must account for the following fanning-related use cases listed in Table C–38. How the connector deals with each use case is a connector implementation detail.

Table C–38   Fanning Use-Cases

Pattern in Update Supported	Record Previously Fanned	Comments
True	True	Represents a unsupported pattern to supported pattern change across synchronization sessions.
True	False	Normal Update. No fanning.
False	True	Represents a unsupported pattern across synchronization sessions.

If a failure occurs when updating existing fanned instances (that is, true updates are being performed on the fanned instances versus delete and refanning), the connector should stop the updating and any successfully updated fanned instance related to the record being updated should be restored and an error should be returned to the Hub. The connector would probably log messages. Such update-then-update-to-restore operations occurring in the same session would be reflected in a subsequent session.

C.8.1.3.4 DeleteRecord When the Hub calls `DeleteRecord`, the connector must determine if the record being deleted has been previously fanned. If so, the connector must delete all fanned instances.

If a failure occurs when deleting existing fanned instances before deleting them all, the connector is not required to restore the deleted fanned instances it had successfully deleted as the delete is retried in subsequent sessions because the **synchronization state** is not saved (assuming the connector stops the deletes and returns an error back to the Hub).

C.8.2 Exchange 2007 PIM Transport Calendar Support

The PIM Transport component supports the following:

- Organizer and Attendees
- Record Associated Data and Exchange Metadata
- Supported Recurrence Patterns
- Time Zones
- Date Time Values
- PIM Record IDs
- ExtractDomain
- CreatePIMRecord
- UpdatePIMRecord
- DeletePIMRecord

C.8.2.1 Organizer and Attendees

Exchange allows for ambiguous names to be listed as attendees. Therefore, the Exchange 2007 Connector writes the list of attendees it obtains from the metadata directly to the Exchange records To, CC and BCC fields.

The BDSS Required Attendees are written to the Exchange To field, the Optional Attendees are written to the Exchange CC field and the Resources are to the Exchange BCC field.

The PIM Transport throws an error if it receives a update request for a meeting for a given user if that user does not appear as a attendee or an organizer.

C.8.2.1.1 Attendee Resolution The PIM Transport uses the Exchange Web Services, the ResolveNames, and ExpandDL, to resolve attendees and distribution list membership.

C.8.2.1.2 Attendee Cache A per-user-per-synchronization-session cache of attendees is created for a synchronizing user. The cache is created when the calendar domain extraction begins and populated with attendee metadata as each calendar record in the extract response is processed. When processing the attendees for a given calendar record, the connector looks in the cache for a match. If a match exists, then the

information in the cache is used when building the ICAL for the record. If no match exists, then the attendee metadata is added to the cache and then to the ICAL.

C.8.2.2 Record Associated Data and Exchange Metadata

Table C–39 describes the metadata that the PIM Transport stores either in the BDSS data store as record-level associated data or in the Exchange store in hidden fields to enable calendar synchronization.

Table C–39   PIM Metadata Stored by the PIM Transport

Name	Value	Comments
BDSSFabricatedId	A unique ID	Used to represent an Exchange calendar record. If a record is fanned by the connector, the ID is used to represent a collection of fanned records
		BDSS - Stored as the PIM Record ID
		Exchange - Each fanned instance in Exchange has this ID written to a hidden field

C.8.2.3 Supported Recurrence Patterns

Table C–40 describes the supported recurrence patterns and gives some examples of each kind. Each sample has a non-ICAL description of the pattern followed by an ICAL RRULE value that represents the pattern. This table does not list all possible patterns.

Table C–40   Supported Recurrence Patterns

Exchange Pattern	Examples of the ICAL RRule Property Value
Daily	Daily, every two days, forever.
	`FREQ=DAILY,INTERVAL=2,WKST=SU`
	Daily, every day, for 10 occurrences.
	`FREQ=DAILY;COUNT=10;INTERVAL=1;WKST=SU`
	Daily, every 5 days, until the end of July
	`FREQ=DAILY;UNTIL=20090731T173000Z;INTERVAL=5;WKST=SU`
	Daily, every weekday, forever
	`FREQ=DAILY;INTERVAL=1;BYDAY=MO,TU,WE,TH,FR;WKST=SU`
	Daily, every weekday, for 10 occurrences.
	`FREQ=DAILY;COUNT=10;INTERVAL=1;BYDAY=MO,TU,WE,TH,FR;WKST=SU`
	Daily, every weekday, until the end of July.
	`FREQ=DAILY;UNTIL=20090731T1730000Z;INTERVAL=1;BYDAY=MO,TU,WE,TH,FR;WKST=SU`
Weekly	Every other week on Monday, Wednesday, and Friday, forever.
	`FREQ=WEEKLY;INTERVAL=2;BYDAY=MO, WE, FR;WKST=SU`
	Every other week on Monday, Wednesday, and Friday, for 10 occurrences.
	`FREQ=WEEKLY;COUNT=10;INTERVAL=1;BYDAY=MO, WE, FR;WKST=SU`
	Every three weeks on Monday, Wednesday and Friday until the end of July.
	`FREQ=WEEKLY;UNTIL=20090731T1730000Z;INTERVAL=3;BYDAY=MO, WE, FR;WKST=SU`

Table C–40   (Cont.) Supported Recurrence Patterns

Exchange Pattern	Examples of the ICAL RRule Property Value
Absolute	The 15th of every other month forever.
Monthly	`FREQ=MONTHLY;INTERVAL=2;BYMONTHDAY=15;WKST=SU`
Relative	Last weekday of every month
Monthly	`FREQ=MONTHLY;INTERVAL=3;BYDAY=MO, TU, WE, TH, FR;BYSETPOS=-1;WKST=SU`
	Last weekend day of every month
	`FREQ=MONTHLY;INTERVAL=1;BYDAY=SU, SA;BYSETPOS=-1;WKST=SU`
	First day of every month for 10 occurrences.
	`FREQ=MONTHLY;COUNT=10;INTERVAL=1;BYDAY=SU, MO, TU, WE, TH, FR, SA; BYSETPOS=; WKST=SU`
	Every fourth Thursday of every other month until the end of October in 2010.
	`FREQ=MONTHLY;UNTIL=20101028T183000Z;INTERVAL=2;BYDAY=4 TH;WKST=SU`
	Monthly every 3rd Thursday.
	`FREQ=MONTHLY;COUNT=10;INTERVAL=1;BYDAY=3 TH;WKST=SU`
	The 4th Monday of every three months.
	`FREQ=MONTHLY;INTERVAL=3;BYDAY=4 MO;WKST=SU`
Absolute	Every year on the 4th of July
Yearly	`YEARLY;INTERVAL=1;BYMONTHDAY=4;BYMONTH=7;WKST=SU`
Relative	Every second of July for 10 occurrences
Yearly	`RRULE:FREQ=YEARLY;COUNT=10;INTERVAL=1;BYDAY=SU, MO, TU, WE, TH, FR, SA;`
	`BYMONTH=7;BYSETPOS=2;WKST=SU`
	Last weekend day of July until 2018.
	`FREQ=YEARLY;UNTIL=20180729T190000Z;INTERVAL=1;BYDAY=SU, SA;BYMONTH=7; BYSETPOS=-1;WKST=SU`

C.8.2.4 Time Zones

An Exchange `CalendarItemType` has two time zone-related properties: a `TimeZone` and `MeetingTimeZone`. The `TimeZone` property is read-only and provides a test-only description of a time zone for a calendar record. The `MeetingTimeZone` property is the property that should be set on new calendar or updated calendar items.

The PIM Transport must convert the `ICAL VTIMEZONE` value to a `MeetingTimeZone` equivalent when creating or updating a calendar record and vice-versa when extracting.

C.8.2.5 Date Time Values

Exchange Web Services store all date and time values in coordinated universal time (UTC). The connector must examine the form of ICAL date time values and convert it to UTC if needed. This should be simple enough to do since the ICAL is also required to have VTIMEZONE information.

C.8.2.6 PIM Record IDs

There are cases where the Exchange 2007 server may cause a given calendar record to have a new PIM Record ID when the record is updated by a user (see Microsoft Support at http://support.microsoft.com/ for more information). Therefore, the PIM Transport must store both the `ItemIdType` the `MeetingId` as record level associated data for every calendar record it creates or extracts.

Anytime the PIM Transport must extract, update or delete an Exchange record, it must take in to account that a given operation may fail because of this record recreation process that Exchange does on certain calendar records.

In general, the PIM Transport accomplishes this by first attempting a given operation using the available `ItemIdType`. If the operation fails and the result code indicates that the failure occurred because the record no longer exists in the Exchange store, then the connector must determine if this error occurred because the end user really deleted the record or it occurred because of the exchange record recreation process. The connector determines this by querying Exchange for the `ItemIdType` of a calendar record containing the same `MeetingId` as that contained in the record associated data provided in the metadata that accompanies the update record call. If the query returns a match, then the record was re-created by Exchange, otherwise, the record was deleted by the user. If the record was re-created by Exchange, the operation is retried using the new `ItemIdType` returned from the query.

The following sections describe additional details regarding what the Exchange PIM Transport must do to account for changing PIM Record IDs when extracting, creating, updating and deleting records from Exchange.

C.8.2.7 ExtractDomain

Using the, `GetAssocDataIdOfAssociatedData` method, which is within the Run-Time Library, the Exchange PIM Transport can be updated to not implement the `DeleteUnaware` interface as it can perform the necessary translation from a `ItemIdType` to the PIM Record ID stored in the BDSS data store. Switching to performing delete detection in the PIM Transport saves an additional query to the Exchange server.

The connector extracts the domain following the same logic as other domains. However, it must take additional steps to coalesce any previously fanned instances into a single `PIMUpsertRecord` containing a single ICAL message.

C.8.2.7.1 Fanning Logic
If the extract response contains any fanned instances, the connector must identify all related fanned instances both in the extract response those not in the extract response as those not extracted may have been updated or deleted in prior sessions and determine how each fanned instance should be represented in the single ICAL message having the original recurrence pattern.

As part of the reconstructing process, the connector must have a master `VEVENT` that contains the original record data and original recurrence pattern. Since the connector must get all fanned instances of a record and coalesce them in to a single ICAL, the connector could attempt to use a fanned instance whose `FanningState` is "Fanned" as this state indicates the initial state of a fanned record. This option, however, fails in the use case where each fanned instance has undergone a change (this is equivalent to a recurring pattern having an exception on each occurrence). Therefore, the PIM Transport stores the master VEVENT as record-level associated data.

For example, suppose during Session 1, the Engine calls `CreateRecord` for a record having a pattern of "The first day of every three months for four occurrences" and that Exchange does not support this pattern. The PIM Transport would fan this pattern to

four non-recurring instances that fall on the first day of every three months. Before Session 2, the Outlook user changes the start date of the first fanned instance and deletes the second fanned instance. When Session 2 occurs, the connector would re-construct the ICAL such that the RRULE pattern is "The first day of every three months for four occurrences" and the extract response would have one upsert containing the calendar record whose date was changed and one delete containing the record id that was deleted. The master VEVENT would be obtained either from record associated data or from the Exchange record and the upsert would appear in the ICAL as a exception VEVENT. The delete would appear in the ICAL as a EXDATE of the master VEVENT. Before Session 3, assume the user updates the third fanned instance by changing the start time on the record. Then when Session 3 occurs, the extract response contains only the third fanned instance, but it must obtain the other instances so that the pattern reflects the changes made during Session 2. The connector must query Exchange for the other records and reconstruct the pattern such that the master VEVENT has a EXDATE for the second occurrence (because it was deleted in Session 2); and two exception VEVENTs - one for the first fanned instance (whose date was changed) and one for the third fanned instance (whose time was changed).

For cases when a user adds a recurrence pattern to a fanned instance, the connector treats the fanned instance as a deleted occurrence of the original pattern and keep the record in the extract response as a separate record instead of including it in the coalesced ICAL. A receiving connector would hide the occurrence of the original pattern due to the EXDATE and create a new recurring record for the fanned exception record.

> **Note:** The ICAL RFC indicates any VEVENT can have any number of RRULEs, RDATEs, EXRULEs and EXDATEs. While it is simple to represent an exception as having a recurrence rule, it is difficult to represent a exception having a recurring pattern which also has exceptions (and this exception containing a recurring pattern which can have its own exceptions pattern could recurse infinitely).

C.8.2.7.2 Changed PIM Record ID Logic Since Exchange can change the PIM Record ID, the connector must determine if a given PIM Record ID contained in the Extract Response has been changed. This can be achieved by detecting a mis-matched `ItemIdType` value of the fanned instance extracted and the `ItemIdTypes` stored as BDSS record associated data for a given original start date (also stored in a hidden field on the Exchange record and as associated data). If a change in id is detected, the connector should update the record associated data to contain the new `ItemIdType`.

C.8.2.8 CreatePIMRecord

The following sections describe the processes that result from a `CallPIMRecord` method call:

- Fanning Logic
- Changed PIM Record Id Logic

C.8.2.8.1 Fanning Logic The PIM Transport evaluates all RRULE, RDATE, and EXDATE properties contained in each VEVENT of a calendar record to determine if the record must be fanned.

If fanning is required, the connector is required to provide BDSS with a single unique ID for the collection of fanned records. Since the PIM Transport essentially does this for other domains by providing a given records' PR_SOURCE_KEY, the connector

provides the PR_SOURCE_KEY of the first fanned instance as the PIM Record ID to represent the collection of fanned records.

Each fanned instance created in Exchange has the following hidden fields as described below:

- `OriginalStartDate`—contains the start date of the fanned instance. This value never changes for the life of the record.

- `BDSSFabricatedPIMRecordId`—contains the record id that is provided to BDSS as the PIM Record ID. It is the PR_SOURCE_KEY of the first fanned instance as mentioned above.

> **Note:** The first fanned instance does not have this hidden field value. The connector accounts for this difference when coalescing related fanned records into a single instance.

- Potentially the master VEVENT data, pending open issue.

- `MeetingId`—the meeting id of the Exchange record.

In addition to the hidden Exchange fields, the PIM Transport has the following record-level associated data that is associated with the fabricated record ID:

- `FannedState`—One for each fanned instance.

- `OriginalStartDate`—One for each fanned instance.

- `ItemIdType`—One for each fanned instance.

- Potentially the master VEVENT data.

If the PIM Transport must fan the record, it calculates the start date for each occurrence and, based on the fanning limit configuration value, create a collection of records that represent the received pattern.

The PIM Transport fans the record if:

- The collection of RRULEs cannot be represented by a supported recurrence pattern.

- If there are any RDATEs in the ICAL.

C.8.2.8.2 Changed PIM Record Id Logic For record creation, a change to the PIM Record ID does not affect the connector, because the record does not exist in Exchange until the method creates it. It must, however, provide record-level associated data so that when the record(s) are modified and extracted, the extraction code can determine whether Exchange has changed the PIM Record ID.

For the Calendar domain, as with all other domains, the Exchange PIM Transport provides a given records' PR_SOURCE_KEY value as the PIM Record ID and store the Exchange Web Service ID, or ItemIdType, as record associated data. The connector must ensure to specify the ItemIdType anytime it interacts with the Exchange record and the PR_SOURCE_KEY anytime it interacts with BDSS.

C.8.2.9 UpdatePIMRecord

This section describes the following process that occur on a UpdatePIMRecord method call:

- Fanning Logic

- [Meeting Id Processing](#)

- [Changed PIM Record Id Logic](#)

C.8.2.9.1   Fanning Logic The PIM Transport determines if the record being updated has been previously fanned by examining the associated data that is provided in the record associated data when the UpdateRecord is called.

The PIM Transport determines if the received ICAL can be natively represented in Exchange by examining the collection of RRULE, RDATE, and EXDATE attributes.

Table C–41 describes the actions that the PIM Transport performs when updating a calendar record:

Table C–41   PIM Transport Update Actions

Pattern in Update Supported?	Record Previously Fanned?	Actions
True	True	Represents a unsupported pattern to supported pattern change across synchronization sessions ■ Delete fanned instances ■ Create new record with supported pattern ■ Keep old `assocdata` related to fanned instances for subsequent extraction so the delete echoes can be removed from the extract response ■ Ensure the created record has the fabricated id as a hidden field

Table C–41   (Cont.) PIM Transport Update Actions

Pattern in Update Supported?	Record Previously Fanned?	Actions
True	False	Perform update
False	True	Represents a unsupported pattern across synchronization sessions
		If the original recurrence pattern changed:
		■ Delete fanned instances
		■ Create new fanned instances
		■ Keep old `assocdata` related to fanned instances for subsequent extraction, ensuring to set the `FannedState` to `Refanned`
		■ Set the echo flag for each created record
		■ Ensure each fanned instance has the same fabricated id provided as the PIM Record Id in the update
		If the original recurrence pattern did not change:
		■ Correlate any exceptions in the received ICAL to existing fanned instances and update the fanned instance with the exception data
		■ Apply master VEVENT data, excluding recurrence pattern data, to the remaining fanned instances
		■ Set the echo flag for each updated fanned instance
False	False	Represents a supported pattern to unsupported pattern change across synchronization sessions
		■ Delete recurring record
		■ Create new fanned instances
		■ Delete old `assocdata`
		■ Create new `assocdata` holding the fanning metadata

If the PIM Transport receives an update and finds that it previously fanned the pattern and a given fanned instance has a `FannedState` of `RecurringException`, but the master VEVENT does not have a EXDATE matching the original start date stored in associated data then the connector must assume that a conflict occurred and Exchange lost. In this case, the connector should delete the fanned record(s) having a FannedState of `RecurringException` and update the associated data. These delete(s) are propagated to other connector(s) in the subsequent synchronization session to delete the recurring record that was synchronized separately due to the fanned exception having a recurrence pattern.

C.8.2.9.2   Meeting Id Processing   The connector stores the `MeetingId` contained in the ICAL to a hidden Exchange field on each updated record and not update the `MeetingId` of the existing calendar record. When extraction is performed, the `MeetingId` from the hidden field is placed in the ICAL instead of the one on the record. The `MeetingId` stored in the hidden field may or may not be the same as the

actual `MeetingId` stored on the record for the meeting. The case where they are the same implies either BDSS or an Exchange client has created the record. Where they are different implies that the update is the result of a conflict and Exchange lost. In either case, it should not be an issue as the connector returns the `MeetingId` that is contained in the ICAL during the update in subsequent sessions to the Hub. By returning the `MeetingID`, the connector does not inadvertently break the meeting management performed by Exchange itself or by Exchange clients.

C.8.2.9.3   Changed PIM Record Id Logic The PIM Transport performs updates as described in Section C.8.2.6, "PIM Record IDs."

C.8.2.10   DeletePIMRecord

The Exchange PIM Transport always has to obtain the record associated data from the BDSS RTL when performing a delete operation for any domain because it always specifies a fabricated Id, also known as the PR_SOURCE_KEY, for the record but must specify the ItemIdType when performing the delete in Exchange.

C.8.2.10.1   Fanning Logic The PIM Transport deletes all fanned instances of a record if the record had been previously fanned by obtaining the record associated data for the record and deleting each instance.

C.8.2.10.2   Changed PIM Record Id Logic The PIM Transport performs deletes as described in Section C.8.2.6, "PIM Record IDs."

D

Custom Field XSLTs and XSDs

This appendix describes the XLSTs and XSDs used to transform custom fields for an Exchange 2007 Connector.

This document includes the following:

- Overview of Custom Field Transformations
- Defining the ICAL Field Definitions
- Transforming the Exchange Connector's Custom Fields into ICAL
- Transforming the ICAL fields into Exchange Connector Fields
- Mapping Windows Time Zones to Java-Standard Time Zones

D.1 Overview of Custom Field Transformations

As noted in Chapter 9, "Synchronizing the Calendar Domain,"

You can add custom fields to a set of XSDs that are not exposed to the Hub using the secondary translation method. To enable the Exchange 2007 Connector to synchronize with the Hub, you must create the following XSD files:

- An XSD file that defines the custom field definitions for the Exchange 2007 Connector
- An XSD file that defines the ICAL counterparts of the field definitions
- An XSLT file that transforms the Exchange 2007 Field definitions into ICAL definitions
- An XSLT file that transforms the ICAL field definitions into Exchange 2007 definitions

These files are considered secondary files because they are never exposed to the Hub. To enable the Exchange 2007 Connector to synchronize these fields, these XSD files are mapped to secondary XSL files.

D.1.1 Defining Custom Fields

Example D–1 illustrates an XSD that defines custom field definitions.

Example D–1   Defining Custom Fields for the Exchange Connector

```
<?xml version="1.0" encoding="windows-1252" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:etl="http://xmlns.oracle.com/bdss/connector/exchange2007/types"
```

```
        xmlns:exch="http://xmlns.oracle.com/bdss/connector/exchange2007/CalendarDomain/exc
hange2007Connector"
                xmlns:hub="http://xmlns.oracle.com/bdss/hub/TypeLibrary"

        targetNamespace="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalen
darDomain"
                elementFormDefault="qualified" exch:PimDomainId="IPM.Appointment"
                exch:FolderClass="IPF.Appointment"
                exch:DomainClassType="CalendarItemType"
                exch:ExchangeDomainBatchSize="512"
                exch:DisposalType="MoveToDeletedItems"
                exch:BodyType="Text"
                exch:FanningLimit="3"
                exch:ConflictResolutionType="AlwaysOverwrite">
  <xsd:import namespace="http://xmlns.oracle.com/bdss/hub/TypeLibrary"
                schemaLocation="../../../hub/HubTypeLibrary.xsd"/>
  <xsd:import
namespace="http://xmlns.oracle.com/bdss/connector/exchange2007/types"
                schemaLocation="../../exchange2007typelibrary.xsd"/>
  <xsd:element name="ExchangeCalendar">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="AdjacentMeetingCount" type="xsd:integer"
                    minOccurs="0" maxOccurs="1" nillable="true"
                    exch:MappedInXsl="false" exch:IsCalculated="true"/>
        <xsd:element name="AllowNewTimeProposal" type="xsd:boolean"
                    minOccurs="0" maxOccurs="1" nillable="true"
                    exch:MappedInXsl="false"/>
        <xsd:element name="AppointmentReplyTime" type="xsd:dateTime"
                    minOccurs="0" maxOccurs="1" nillable="true"
                    exch:MappedInXsl="false"/>
        <xsd:element name="AppointmentSequenceNumber" type="xsd:integer"
                    minOccurs="0" maxOccurs="1" nillable="true"
                    exch:MappedInXsl="false"/>
        <xsd:element name="AppointmentState" type="xsd:integer" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
        <xsd:element name="Body" type="xsd:string" minOccurs="0" maxOccurs="1"
                    nillable="true" exch:MappedInXsl="true"
                    exch:IsCalculated="true"/>
        <xsd:element name="CalendarItemType" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
        <xsd:element name="Categories" type="hub:HubTokens" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="ConferenceType" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
        <xsd:element name="ConflictingMeetingCount" type="xsd:integer"
                    minOccurs="0" maxOccurs="1" nillable="true"
                    exch:MappedInXsl="false" exch:IsCalculated="true"/>
        <xsd:element name="DateTimeStamp" type="xsd:dateTime" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
        <xsd:element name="Duration" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
        <xsd:element name="End" type="xsd:dateTime" minOccurs="0" maxOccurs="1"
                    nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="IsAllDayEvent" type="xsd:boolean" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
        <xsd:element name="IsCancelled" type="xsd:boolean" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"
                    exch:IsCalculated="true"/>
        <xsd:element name="IsMeeting" type="xsd:boolean" minOccurs="0"
```

```
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
<xsd:element name="IsOnlineMeeting" type="xsd:boolean" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
<xsd:element name="IsRecurring" type="xsd:boolean" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"
                    exch:IsCalculated="true"/>
<xsd:element name="IsResponseRequested" type="xsd:boolean" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
<xsd:element name="ItemClass" type="xsd:string" minOccurs="1"
                    maxOccurs="1" nillable="false" exch:MappedInXsl="true"/>
<xsd:element name="LegacyFreeBusyStatus" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
<xsd:element name="Location" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
<xsd:element name="MeetingRequestWasSent" type="xsd:boolean"
                    minOccurs="0" maxOccurs="1" nillable="true"
                    exch:MappedInXsl="false"/>
<xsd:element name="MeetingWorkspaceUrl" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
<xsd:element name="MyResponseType" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
<xsd:element name="NetShowUrl" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
<xsd:element name="OptionalAttendees" type="hub:HubTokens" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"
                    exch:IsCalculated="true"/>
<xsd:element name="Organizer" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
<xsd:element name="OriginalStart" type="xsd:dateTime" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"
                    exch:IsCalculated="true"/>
<xsd:element name="Importance" type="etl:ExchangeImportance"
                    default="Normal" exch:MappedInXsl="true"/>
<xsd:element name="ReminderDueBy" type="xsd:dateTime" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
<xsd:element name="ReminderIsSet" type="xsd:boolean" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
<xsd:element name="ReminderMinutesBeforeStart"
                    type="etl:ExchangeUnsignedLong" minOccurs="0" maxOccurs="1"
                    nillable="true" exch:MappedInXsl="true"/>
<xsd:element name="RequiredAttendees" type="hub:HubTokens" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"
                    exch:IsCalculated="true"/>
<xsd:element name="Resources" type="hub:HubTokens" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"
                    exch:IsCalculated="true"/>
<xsd:element name="Sensitivity" type="etl:ExchangeSensitivity"
                    exch:MappedInXsl="true"/>
<xsd:element name="Start" type="xsd:dateTime" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
<xsd:element name="Subject" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
<xsd:element name="TimeZone" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="false"/>
<xsd:element name="UID" type="xsd:string" minOccurs="0" maxOccurs="1"
                    nillable="true" exch:MappedInXsl="true"/>
<xsd:element name="When" type="xsd:dateTime" minOccurs="0" maxOccurs="1"
                    nillable="true" exch:MappedInXsl="false"/>
<xsd:element name="FabricatedId" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true"
```

```
                            exch:PropertyName="BDSSFabricatedId"
                            exch:DistinguishedPropertySetId="PublicStrings"
                            exch:PropertyType="String" exch:MappedInXsl="true"/>
                            <!-- Meeting fields -->
            <xsd:element name="ResponseStatus" type="etl:ExchangeUnsignedLong"
minOccurs="0"
                            maxOccurs="1" nillable="true"
                            exch:PropertyId="0x8218"
                            exch:DistinguishedPropertySetId="Appointment"
                            exch:PropertyType="Integer" exch:MappedInXsl="true"/>
            <xsd:element name="MeetingStatus" type="etl:ExchangeUnsignedLong"
minOccurs="0"
                            maxOccurs="1" nillable="true"
                            exch:PropertyId="0x8217"
                            exch:DistinguishedPropertySetId="Appointment"
                            exch:PropertyType="Integer" exch:MappedInXsl="true"/>
            <xsd:element name="ResponseRequested" type="xsd:boolean" minOccurs="0"
                            maxOccurs="1" nillable="true"
                            exch:PropertyTag="0x0063"
                            exch:PropertyType="Boolean" exch:MappedInXsl="true"/>
            <xsd:element name="OwnerAppointmentId" type="etl:ExchangeUnsignedLong"
minOccurs="0"
                            maxOccurs="1" nillable="true"
                            exch:PropertyTag="0x0062"
                            exch:PropertyType="Integer" exch:MappedInXsl="true"/>
            <xsd:element name="ReplyTo" type="xsd:string" minOccurs="0"
                            maxOccurs="1" nillable="true"
                            exch:DistinguishedPropertySetId="InternetHeaders"
                            exch:PropertyName="reply-to"
                            exch:PropertyType="String" exch:MappedInXsl="true"/>
        </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

D.2 Defining the ICAL Field Definitions

Example D–2 illustrates an XSD that defines the ICAL definitions for the custom fields.

Example D–2   Defining ICAL Definitions

```
<?xml version="1.0" encoding="windows-1252" ?>
<!--  $Header:
fts/connectors/exchange2007/oracle/bdss/connectors/exchange2007/pimtransport/trans
form/exchange/calendar/intermediatetranslation/veventasxml.xsd st_fts_lharding_
fanningsupport/2 2009/07/04 14:35:59 lharding Exp $ -->
<!--
 Copyright (c) 2008, 2009, Oracle and/or its affiliates. All rights reserved.

   NAME
     exchange2007contact.xsd

   DESCRIPTION
     Contains XML schema definition that for VEVENT properties of a ICAL
     message

   NOTES
     <other useful comments, qualifications, etc.>
```

```
   MODIFIED   (MM/DD/YY)
      lharding 01/10/09 - initial version
-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalendarVEVENT"
         xmlns:etl="http://xmlns.oracle.com/bdss/connector/exchange2007/types"

xmlns:exch="http://xmlns.oracle.com/bdss/connector/exchange2007/CalendarDomain/exc
hange2007Connector"
         xmlns:hub="http://xmlns.oracle.com/bdss/hub/TypeLibrary"

targetNamespace="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalen
darVEVENT"
         elementFormDefault="qualified">
  <xsd:import namespace="http://xmlns.oracle.com/bdss/hub/TypeLibrary"
            schemaLocation="../../../hub/HubTypeLibrary.xsd"/>
  <xsd:import
namespace="http://xmlns.oracle.com/bdss/connector/exchange2007/types"
            schemaLocation="../../exchange2007typelibrary.xsd"/>
  <xsd:complexType name="TriggerParams">
    <xsd:sequence>
      <xsd:element name="RELATED" type="xsd:string" minOccurs="0" maxOccurs="1"
                  nillable="true" exch:MappedInXsl="true"/>
      <xsd:element name="PERIOD" type="xsd:string" minOccurs="0" maxOccurs="1"
                  nillable="true" exch:MappedInXsl="true"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="VALARM">
    <xsd:sequence>
      <xsd:element name="ACTION" type="xsd:string" minOccurs="0" maxOccurs="1"
                  nillable="true" exch:MappedInXsl="true"/>
      <xsd:element name="DESCRIPTION" type="xsd:string" minOccurs="0"
                  maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
      <xsd:element name="TRIGGER" type="TriggerParams" minOccurs="0"
                  maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="ExchangeCalendarVEvent">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="VALARM" type="VALARM" minOccurs="0" maxOccurs="1"
                    nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="CALSCALE" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="CATEGORIES" type="hub:HubTokens" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="CLASS" type="xsd:string" minOccurs="0" maxOccurs="1"
                    nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="COMMENT" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="CREATED" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="DESCRIPTION" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="DTEND" type="xsd:string" minOccurs="0" maxOccurs="1"
                    nillable="true" exch:MappedInXsl="true"/>
        <xsd:element name="DTSTAMP" type="xsd:string" minOccurs="0"
                    maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
```

```
                      <xsd:element name="DTSTART" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="DURATION" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="EXDATE" type="hub:HubTokens" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="LAST-MODIFIED" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="false" exch:MappedInXsl="true"/>
                      <xsd:element name="LOCATION" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="ATTENDEE" type="hub:HubTokens" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="ORGANIZER" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="PRIORITY" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="RECURRENCE-ID" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="RESOURCES" type="hub:HubTokens" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="RELATED-TO" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="SEQUENCE" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="STATUS" type="xsd:string" minOccurs="0" maxOccurs="1"
                                   nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="SUMMARY" type="xsd:string" minOccurs="0"
                                   maxOccurs="1" nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="TRANSP" type="xsd:string" minOccurs="0" maxOccurs="1"
                                   nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="GEO" type="xsd:string" minOccurs="0" maxOccurs="1"
                                   nillable="false" exch:MappedInXsl="true"/>
                      <xsd:element name="UID" type="xsd:string" minOccurs="1" maxOccurs="1"
                                   nillable="false" exch:MappedInXsl="true"/>
                      <xsd:element name="URL" type="xsd:string" minOccurs="0" maxOccurs="1"
                                   nillable="true" exch:MappedInXsl="true"/>
                      <xsd:element name="X-MICROSOFT-CDO-ALLDAYEVENT" type="xsd:boolean"
                                   minOccurs="0" maxOccurs="1" nillable="true"
                                   exch:MappedInXsl="true"/>
                      <xsd:element name="X-ORACLE-BDSS-FABRICATED-ID" type="xsd:string"
                                   minOccurs="1" maxOccurs="1" nillable="false"
                                   exch:MappedInXsl="true"/>
                      <xsd:element name="X-MICROSOFT-CDO-OWNERAPPTID"
              type="etl:ExchangeUnsignedLong"
                                   minOccurs="0" maxOccurs="1" nillable="true"
                                   exch:MappedInXsl="true"/>
                      <xsd:element name="X-ORACLE-BDSS-SYNCING-USER" type="xsd:string"
                                   minOccurs="1" maxOccurs="1" nillable="false"
                                   exch:MappedInXsl="true"/>
                 </xsd:sequence>
             </xsd:complexType>
          </xsd:element>
       </xsd:schema>
```

D.3 Transforming the Exchange Connector's Custom Fields into ICAL

Example D–3 illustrates the XLST that transforms the custom fields defined in
Example D–1 into ICAL.

Example D–3   Transforming the Custom Fields into ICAL

```
<?xml version="1.0" encoding="UTF-8" ?>
<?oracle-xsl-mapper <!-- SPECIFICATION OF MAP SOURCES AND TARGETS, DO NOT MODIFY.
-->
  <mapSources>
    <source type="XSD">
      <schema location="calendaritemtype.xsd"/>
      <rootElement name="ExchangeCalendar"
namespace="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalendarDom
ain"/>
    </source>
  </mapSources>
  <mapTargets>
    <target type="XSD">
      <schema location="veventasxml.xsd"/>
      <rootElement name="ExchangeCalendarVEvent"
namespace="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalendarVEV
ENT"/>
    </target>
  </mapTargets>
  <!-- GENERATED BY ORACLE XSL MAPPER 11.1.1.0.0(build 090218.0200.2549) AT [WED
MAY 13 20:14:06 CDT 2009]. -->
?>
<xsl:stylesheet version="1.0"

xmlns:xpath20="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.fun
ctions.Xpath20"

xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"

xmlns:ns1="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalendarVEV
ENT"

xmlns:mhdr="http://www.oracle.com/XSL/Transform/java/oracle.tip.mediator.service.c
ommon.functions.MediatorExtnFunction"

xmlns:oraext="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.func
tions.ExtFunc"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:dvm="http://www.oracle.com/XSL/Transform/java/oracle.tip.dvm.LookupValue"
                xmlns:hwf="http://xmlns.oracle.com/bpel/workflow/xpath"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:med="http://schemas.oracle.com/mediator/xpath"

xmlns:exch="http://xmlns.oracle.com/bdss/connector/exchange2007/CalendarDomain/exc
hange2007Connector"

xmlns:ids="http://xmlns.oracle.com/bpel/services/IdentityService/xpath"

xmlns:xdk="http://schemas.oracle.com/bpel/extension/xpath/function/xdk"

xmlns:xref="http://www.oracle.com/XSL/Transform/java/oracle.tip.xref.xpath.XRefXPa
thFunctions"

xmlns:icalconverter="http://www.oracle.com/XSL/Transform/java/oracle.bdss.connecto
rs.exchange2007.pimtransport.transform.exchange.calendar.intermediatetranslation.I
CalConverter"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:ns0="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalendarDom
ain"
                xmlns:ora="http://schemas.oracle.com/xpath/extension"
                xmlns:hub="http://xmlns.oracle.com/bdss/hub/TypeLibrary"

xmlns:socket="http://www.oracle.com/XSL/Transform/java/oracle.tip.adapter.socket.P
rotocolTranslator"

xmlns:etl="http://xmlns.oracle.com/bdss/connector/exchange2007/types"
                xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
                exclude-result-prefixes="xsi xsl exch xsd ns0 hub etl ns1 xpath20
bpws mhdr oraext dvm hwf med ids xdk xref ora socket ldap">
  <xsl:template match="/">
    <ns1:ExchangeCalendarVEvent>
      <ns1:VALARM>
        <xsl:choose>
          <xsl:when test='/ns0:ExchangeCalendar/ns0:ReminderIsSet = "true"'>
            <ns1:ACTION>
              <xsl:text disable-output-escaping="no">DISPLAY</xsl:text>
            </ns1:ACTION>
          </xsl:when>
        </xsl:choose>
        <xsl:choose>
          <xsl:when test='/ns0:ExchangeCalendar/ns0:ReminderIsSet = "true"'>
            <ns1:DESCRIPTION>
              <xsl:text disable-output-escaping="no">REMINDER</xsl:text>
            </ns1:DESCRIPTION>
          </xsl:when>
        </xsl:choose>
        <xsl:choose>
          <xsl:when test='/ns0:ExchangeCalendar/ns0:ReminderIsSet = "true"'>
            <ns1:TRIGGER>
              <ns1:PERIOD>
                <xsl:value-of
select="icalconverter:ConvertReminderMinutesToDuration(/ns0:ExchangeCalendar/ns0:R
eminderMinutesBeforeStart)"/>
              </ns1:PERIOD>
              <ns1:RELATED>
                <xsl:text disable-output-escaping="no">START</xsl:text>
              </ns1:RELATED>
            </ns1:TRIGGER>
          </xsl:when>
        </xsl:choose>
      </ns1:VALARM>
      <ns1:CALSCALE>
        <xsl:text disable-output-escaping="no">GREGORIAN</xsl:text>
      </ns1:CALSCALE>
      <ns1:CATEGORIES>
        <xsl:for-each select="/ns0:ExchangeCalendar/ns0:Categories/ns0:Token">
          <hub:Token>
            <xsl:value-of select="."/>
          </hub:Token>
        </xsl:for-each>
      </ns1:CATEGORIES>
      <xsl:choose>
        <xsl:when test='/ns0:ExchangeCalendar/ns0:Sensitivity = "Private"'>
          <ns1:CLASS>
            <xsl:text disable-output-escaping="no">PRIVATE</xsl:text>
          </ns1:CLASS>
        </xsl:when>
```

```
              <xsl:when test='/ns0:ExchangeCalendar/ns0:Sensitivity = "Confidential"'>
                <ns1:CLASS>
                  <xsl:text disable-output-escaping="no">CONFIDENTIAL</xsl:text>
                </ns1:CLASS>
              </xsl:when>
              <xsl:otherwise>
                <ns1:CLASS>
                  <xsl:text disable-output-escaping="no">PUBLIC</xsl:text>
                </ns1:CLASS>
              </xsl:otherwise>
            </xsl:choose>
            <ns1:DESCRIPTION>
              <xsl:value-of select="/ns0:ExchangeCalendar/ns0:Body"/>
            </ns1:DESCRIPTION>
            <ns1:DTEND>
              <xsl:value-of
select="icalconverter:FormatAsICalDate(/ns0:ExchangeCalendar/ns0:End)"/>
            </ns1:DTEND>
            <ns1:DTSTART>
              <xsl:value-of
select="icalconverter:FormatAsICalDate(/ns0:ExchangeCalendar/ns0:Start)"/>
            </ns1:DTSTART>
            <ns1:LOCATION>
              <xsl:value-of select="/ns0:ExchangeCalendar/ns0:Location"/>
            </ns1:LOCATION>
            <ns1:ORGANIZER>
              <xsl:value-of select="/ns0:ExchangeCalendar/ns0:Organizer"/>
            </ns1:ORGANIZER>
            <xsl:choose>
              <xsl:when test='/ns0:ExchangeCalendar/ns0:Importance = "Low"'>
                <ns1:PRIORITY>
                  <xsl:text disable-output-escaping="no">9</xsl:text>
                </ns1:PRIORITY>
              </xsl:when>
              <xsl:when test='/ns0:ExchangeCalendar/ns0:Importance = "High"'>
                <ns1:PRIORITY>
                  <xsl:text disable-output-escaping="no">1</xsl:text>
                </ns1:PRIORITY>
              </xsl:when>
              <xsl:otherwise>
                <ns1:PRIORITY>
                  <xsl:text disable-output-escaping="no">5</xsl:text>
                </ns1:PRIORITY>
              </xsl:otherwise>
            </xsl:choose>
            <ns1:ATTENDEE>
              <xsl:for-each
select="/ns0:ExchangeCalendar/ns0:RequiredAttendees/ns0:Token">
                <hub:Token>
                  <xsl:value-of select="."/>
                </hub:Token>
              </xsl:for-each>
              <xsl:for-each
select="/ns0:ExchangeCalendar/ns0:OptionalAttendees/ns0:Token">
                <hub:Token>
                  <xsl:value-of select="."/>
                </hub:Token>
              </xsl:for-each>
              <xsl:for-each select="/ns0:ExchangeCalendar/ns0:Resources/ns0:Token">
                <hub:Token>
```

```
        <xsl:value-of select="."/>
      </hub:Token>
    </xsl:for-each>
  </ns1:ATTENDEE>
  <ns1:STATUS>
    <xsl:text disable-output-escaping="no">CONFIRMED</xsl:text>
  </ns1:STATUS>
  <ns1:SUMMARY>
    <xsl:value-of select="/ns0:ExchangeCalendar/ns0:Subject"/>
  </ns1:SUMMARY>
  <ns1:TRANSP>
    <xsl:text disable-output-escaping="no">OPAQUE</xsl:text>
  </ns1:TRANSP>
  <ns1:UID>
    <xsl:value-of select="/ns0:ExchangeCalendar/ns0:UID"/>
  </ns1:UID>
  <ns1:X-MICROSOFT-CDO-OWNERAPPTID>
    <xsl:value-of select="/ns0:ExchangeCalendar/ns0:OwnerAppointmentId"/>
  </ns1:X-MICROSOFT-CDO-OWNERAPPTID>
 <ns1:X-ORACLE-BDSS-FABRICATED-ID>
    <xsl:value-of select="/ns0:ExchangeCalendar/ns0:FabricatedId"/>
  </ns1:X-ORACLE-BDSS-FABRICATED-ID>
 </ns1:ExchangeCalendarVEvent>
 </xsl:template>
</xsl:stylesheet>
```

D.4 Transforming the ICAL fields into Exchange Connector Fields

Example D–4 illustrates the XSLT that translates the ICAL properties of Example D–2
in to the format of the Exchange 2007 Connector.

Example D–4   Translating the ICAL Properties

```
<?xml version="1.0" encoding="UTF-8" ?>
<?oracle-xsl-mapper <!-- SPECIFICATION OF MAP SOURCES AND TARGETS, DO NOT MODIFY.
-->
  <mapSources>
    <source type="XSD">
      <schema location="veventasxml.xsd"/>
      <rootElement name="ExchangeCalendarVEvent"
namespace="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalendarVEV
ENT"/>
    </source>
  </mapSources>
  <mapTargets>
    <target type="XSD">
      <schema location="calendaritemtype.xsd"/>
      <rootElement name="ExchangeCalendar"
namespace="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalendarDom
ain"/>
    </target>
  </mapTargets>
  <!-- GENERATED BY ORACLE XSL MAPPER 11.1.1.0.0(build 090218.0200.2549) AT [WED
MAY 13 20:45:20 CDT 2009]. -->
?>
<xsl:stylesheet version="1.0"

xmlns:xpath20="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.fun
ctions.Xpath20"
```

```
xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"

xmlns:ns0="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalendarVEV
ENT"

xmlns:mhdr="http://www.oracle.com/XSL/Transform/java/oracle.tip.mediator.service.c
ommon.functions.MediatorExtnFunction"

xmlns:oraext="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.func
tions.ExtFunc"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:dvm="http://www.oracle.com/XSL/Transform/java/oracle.tip.dvm.LookupValue"
                xmlns:hwf="http://xmlns.oracle.com/bpel/workflow/xpath"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:med="http://schemas.oracle.com/mediator/xpath"

xmlns:exch="http://xmlns.oracle.com/bdss/connector/exchange2007/CalendarDomain/exc
hange2007Connector"

xmlns:ids="http://xmlns.oracle.com/bpel/services/IdentityService/xpath"

xmlns:xdk="http://schemas.oracle.com/bpel/extension/xpath/function/xdk"

xmlns:xref="http://www.oracle.com/XSL/Transform/java/oracle.tip.xref.xpath.XRefXPa
thFunctions"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:ns1="http://xmlns.oracle.com/bdss/connector/exchange2007/ExchangeCalendarDom
ain"
                xmlns:ora="http://schemas.oracle.com/xpath/extension"

xmlns:icalconverter="http://www.oracle.com/XSL/Transform/java/oracle.bdss.connecto
rs.exchange2007.pimtransport.transform.exchange.calendar.intermediatetranslation.I
CalConverter"
                xmlns:hub="http://xmlns.oracle.com/bdss/hub/TypeLibrary"

xmlns:socket="http://www.oracle.com/XSL/Transform/java/oracle.tip.adapter.socket.P
rotocolTranslator"

xmlns:etl="http://xmlns.oracle.com/bdss/connector/exchange2007/types"
                xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
                exclude-result-prefixes="xsi xsl ns0 exch xsd hub etl ns1 xpath20
bpws mhdr oraext dvm hwf med ids xdk xref ora socket ldap">
  <xsl:template match="/">
    <ns1:ExchangeCalendar>
      <ns1:Body>
        <xsl:value-of select="/ns0:ExchangeCalendarVEvent/ns0:DESCRIPTION"/>
      </ns1:Body>
      <ns1:Categories>
        <xsl:for-each
select="/ns0:ExchangeCalendarVEvent/ns0:CATEGORIES/ns0:Token">
          <hub:Token>
            <xsl:value-of select="."/>
          </hub:Token>
        </xsl:for-each>
      </ns1:Categories>
      <ns1:End>
        <xsl:value-of select="/ns0:ExchangeCalendarVEvent/ns0:DTEND"/>
```

```
          </ns1:End>
          <ns1:ItemClass>
            <xsl:text disable-output-escaping="no">IPM.Appointment</xsl:text>
          </ns1:ItemClass>
          <ns1:Location>
            <xsl:value-of select="/ns0:ExchangeCalendarVEvent/ns0:LOCATION"/>
          </ns1:Location>
          <ns1:OptionalAttendees>
            <xsl:for-each
select="/ns0:ExchangeCalendarVEvent/ns0:OPT-PARTICIPANT/ns0:Token">
               <xsl:choose>
                 <xsl:when
test="not(contains(.,/ns0:ExchangeCalendarVEvent/ns0:X-ORACLE-BDSS-SYNCING-USER))"
>
                   <hub:Token>
                     <xsl:value-of select="substring-after(.,'MAILTO:')"/>
                   </hub:Token>
                 </xsl:when>
               </xsl:choose>
            </xsl:for-each>
          </ns1:OptionalAttendees>
          <ns1:Organizer>
            <xsl:value-of
select="translate(substring-after(/ns0:ExchangeCalendarVEvent/ns0:ORGANIZER,'MAILT
O:'),' ','')"/>
          </ns1:Organizer>
          <xsl:choose>
            <xsl:when test='/ns0:ExchangeCalendarVEvent/ns0:PRIORITY = "1"'>
              <ns1:Importance>
                <xsl:text disable-output-escaping="no">High</xsl:text>
              </ns1:Importance>
            </xsl:when>
            <xsl:when test='/ns0:ExchangeCalendarVEvent/ns0:PRIORITY = "5"'>
              <ns1:Importance>
                <xsl:text disable-output-escaping="no">Normal</xsl:text>
              </ns1:Importance>
            </xsl:when>
            <xsl:when test='/ns0:ExchangeCalendarVEvent/ns0:PRIORITY = "9"'>
              <ns1:Importance>
                <xsl:text disable-output-escaping="no">Low</xsl:text>
              </ns1:Importance>
            </xsl:when>
            <xsl:otherwise>
              <ns1:Importance>
                <xsl:text disable-output-escaping="no">Normal</xsl:text>
              </ns1:Importance>
            </xsl:otherwise>
          </xsl:choose>
          <xsl:choose>
            <xsl:when
test='/ns0:ExchangeCalendarVEvent/ns0:VALARM/ns0:TRIGGER/ns0:PERIOD != ""'>
              <ns1:ReminderIsSet>
                <xsl:text disable-output-escaping="no">True</xsl:text>
              </ns1:ReminderIsSet>
            </xsl:when>
            <xsl:otherwise>
              <ns1:ReminderIsSet>
                <xsl:text disable-output-escaping="no">False</xsl:text>
              </ns1:ReminderIsSet>
            </xsl:otherwise>
```

```
                </xsl:choose>
                <xsl:choose>
                  <xsl:when
test='/ns0:ExchangeCalendarVEvent/ns0:VALARM/ns0:TRIGGER/ns0:PERIOD != ""'>
                    <ns1:ReminderMinutesBeforeStart>
                      <xsl:value-of
select="icalconverter:DurationToMinutesBeforeStart(/ns0:ExchangeCalendarVEvent/ns0
:VALARM/ns0:TRIGGER/ns0:PERIOD)"/>
                    </ns1:ReminderMinutesBeforeStart>
                  </xsl:when>
                  <xsl:otherwise>
                    <ns1:ReminderMinutesBeforeStart/>
                  </xsl:otherwise>
                </xsl:choose>
                <xsl:choose>
                  <xsl:when
test='/ns0:ExchangeCalendarVEvent/ns0:VALARM/ns0:TRIGGER/ns0:PERIOD != ""'>
                    <ns1:ReminderDueBy>
                      <xsl:value-of
select="icalconverter:DurationToReminderDueByDate(/ns0:ExchangeCalendarVEvent/ns0:
DTSTART,/ns0:ExchangeCalendarVEvent/ns0:VALARM/ns0:TRIGGER/ns0:PERIOD)"/>
                    </ns1:ReminderDueBy>
                  </xsl:when>
                  <xsl:otherwise>
                    <ns1:ReminderDueBy/>
                  </xsl:otherwise>
                </xsl:choose>
                <ns1:RequiredAttendees>
                  <xsl:for-each
select="/ns0:ExchangeCalendarVEvent/ns0:REQ-PARTICIPANT/ns0:Token">
                    <xsl:choose>
                      <xsl:when
test="not(contains(.,/ns0:ExchangeCalendarVEvent/ns0:X-ORACLE-BDSS-SYNCING-USER))"
>
                        <hub:Token>
                          <xsl:value-of select="substring-after(.,'MAILTO:')"/>
                        </hub:Token>
                      </xsl:when>
                    </xsl:choose>
                  </xsl:for-each>
                  <xsl:if
test="not(contains(translate(substring-after(/ns0:ExchangeCalendarVEvent/ns0:ORGAN
IZER,'MAILTO:'),' ',
''),/ns0:ExchangeCalendarVEvent/ns0:X-ORACLE-BDSS-SYNCING-USER))">
                    <hub:Token>
                      <xsl:value-of
select="translate(substring-after(/ns0:ExchangeCalendarVEvent/ns0:ORGANIZER,'MAILT
O:'),' ','')"/>
                    </hub:Token>
                  </xsl:if>
                </ns1:RequiredAttendees>
                <ns1:Resources>
                  <xsl:for-each
select="/ns0:ExchangeCalendarVEvent/ns0:NON-PARTICIPANT/ns0:Token">
                    <xsl:choose>
                      <xsl:when
test="not(contains(.,/ns0:ExchangeCalendarVEvent/ns0:X-ORACLE-BDSS-SYNCING-USER))"
>
                        <hub:Token>
                          <xsl:value-of select="substring-after(.,'MAILTO:')"/>
```

```
              </hub:Token>
            </xsl:when>
          </xsl:choose>
        </xsl:for-each>
      </ns1:Resources>
      <xsl:choose>
        <xsl:when test='/ns0:ExchangeCalendarVEvent/ns0:CLASS = "PRIVATE"'>
          <ns1:Sensitivity>
            <xsl:text disable-output-escaping="no">Private</xsl:text>
          </ns1:Sensitivity>
        </xsl:when>
        <xsl:when test='/ns0:ExchangeCalendarVEvent/ns0:CLASS = "CONFIDENTIAL"'>
          <ns1:Sensitivity>
            <xsl:text disable-output-escaping="no">Confidential</xsl:text>
          </ns1:Sensitivity>
        </xsl:when>
        <xsl:otherwise>
          <ns1:Sensitivity>
            <xsl:text disable-output-escaping="no">Normal</xsl:text>
          </ns1:Sensitivity>
        </xsl:otherwise>
      </xsl:choose>
      <ns1:Start>
        <xsl:value-of select="/ns0:ExchangeCalendarVEvent/ns0:DTSTART"/>
      </ns1:Start>
      <ns1:Subject>
        <xsl:value-of select="/ns0:ExchangeCalendarVEvent/ns0:SUMMARY"/>
      </ns1:Subject>
      <ns1:UID>
        <xsl:value-of select="/ns0:ExchangeCalendarVEvent/ns0:UID"/>
      </ns1:UID>
      <ns1:FabricatedId>
        <xsl:value-of
select="/ns0:ExchangeCalendarVEvent/ns0:X-ORACLE-BDSS-FABRICATED-ID"/>
      </ns1:FabricatedId>
      <ns1:OwnerAppointmentId>
        <xsl:value-of
select="/ns0:ExchangeCalendarVEvent/ns0:X-MICROSOFT-CDO-OWNERAPPTID"/>
      </ns1:OwnerAppointmentId>
      <xsl:choose>
        <xsl:when test='/ns0:ExchangeCalendarVEvent/ns0:REQ-PARTICIPANT/ns0:Token
!= "" or /ns0:ExchangeCalendarVEvent/ns0:OPT-PARTICIPANT/ns0:Token != "" or
/ns0:ExchangeCalendarVEvent/ns0:NON-PARTICIPANT/ns0:Token != ""'>
          <ns1:ResponseRequested>
            <xsl:text disable-output-escaping="no">true</xsl:text>
          </ns1:ResponseRequested>
        </xsl:when>
        <xsl:otherwise>
          <ns1:ResponseRequested>
            <xsl:text disable-output-escaping="no">false</xsl:text>
          </ns1:ResponseRequested>
        </xsl:otherwise>
      </xsl:choose>
      <xsl:choose>
        <xsl:when test="contains(/ns0:ExchangeCalendarVEvent/ns0:ORGANIZER,
/ns0:ExchangeCalendarVEvent/ns0:X-ORACLE-BDSS-SYNCING-USER) = 'true'">
          <!-- SyncingUserIsOrganizer -->
          <xsl:choose>
            <xsl:when
test='/ns0:ExchangeCalendarVEvent/ns0:REQ-PARTICIPANT/ns0:Token != "" or
```

```
/ns0:ExchangeCalendarVEvent/ns0:OPT-PARTICIPANT/ns0:Token != "" or
/ns0:ExchangeCalendarVEvent/ns0:NON-PARTICIPANT/ns0:Token != ""'>
                <!-- Attendees Present -->
                <ns1:ResponseStatus>
                  <xsl:text disable-output-escaping="no">1</xsl:text>
                </ns1:ResponseStatus>
                <ns1:MeetingStatus>
                  <xsl:text disable-output-escaping="no">1</xsl:text>
                </ns1:MeetingStatus>
                <ns1:ReplyTo/>
            </xsl:when>
            <xsl:otherwise>
              <!-- No Attendees Present-->
              <ns1:ResponseStatus>
                <xsl:text disable-output-escaping="no">0</xsl:text>
              </ns1:ResponseStatus>
              <ns1:MeetingStatus>
                <xsl:text disable-output-escaping="no">0</xsl:text>
              </ns1:MeetingStatus>
              <ns1:ReplyTo/>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:when>
        <xsl:otherwise>
          <!-- SyncingUserNotOrganizer-->
          <xsl:choose>
            <xsl:when
test='/ns0:ExchangeCalendarVEvent/ns0:REQ-PARTICIPANT/ns0:Token != "" or
/ns0:ExchangeCalendarVEvent/ns0:OPT-PARTICIPANT/ns0:Token != "" or
/ns0:ExchangeCalendarVEvent/ns0:NON-PARTICIPANT/ns0:Token != ""'>
                <!-- Attendees Present -->
                <ns1:ResponseStatus>
                  <xsl:text disable-output-escaping="no">3</xsl:text>
                </ns1:ResponseStatus>
                <ns1:MeetingStatus>
                  <xsl:text disable-output-escaping="no">3</xsl:text>
                </ns1:MeetingStatus>
                <ns1:ReplyTo>
                  <xsl:value-of
select="translate(substring-after(/ns0:ExchangeCalendarVEvent/ns0:ORGANIZER,'MAILT
O:'),' ','')"/>
                </ns1:ReplyTo>
            </xsl:when>
            <xsl:otherwise>
              <!-- No Attendees Present -->
              <ns1:ReplyTo/>
              <ns1:ResponseStatus>
                <xsl:text disable-output-escaping="no">0</xsl:text>
              </ns1:ResponseStatus>
              <ns1:MeetingStatus>
                <xsl:text disable-output-escaping="no">0</xsl:text>
              </ns1:MeetingStatus>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:otherwise>
      </xsl:choose>
    </ns1:ExchangeCalendar>
  </xsl:template>
</xsl:stylesheet>
```

D.5 Mapping Windows Time Zones to Java-Standard Time Zones

Example D–5 illustrates an the XSD that maps Microsoft Windows time zones to Java-standard time zones when building Hub Calendar records. Example D–6 illustrates the XLST file that transforms the time zone mappings.

Example D–5   XSD for Mapping Microsoft Windows Time Zones to Java-Standard Time Zones

```xml
<?xml version="1.0" encoding="windows-1252" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="TimeZoneMapping">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="TimeZone" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="Java" minOccurs="1" nillable="false"
                        maxOccurs="1">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="TimeZoneIds" maxOccurs="1">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="TimeZoneId" maxOccurs="unbounded">
                            <xsd:complexType>
                              <xsd:simpleContent>
                                <xsd:extension base="xsd:string">
                                  <xsd:attribute name="extractionDefault"
                                              type="xsd:string"/>
                                </xsd:extension>
                              </xsd:simpleContent>
                            </xsd:complexType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
              <xsd:element name="Exchange" nillable="false" maxOccurs="1">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="TimeZoneIds" minOccurs="1" maxOccurs="1">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="TimeZoneId" minOccurs="1"
maxOccurs="unbounded">
                            <xsd:complexType>
                              <xsd:simpleContent>
                                <xsd:extension base="xsd:string">
                                  <xsd:attribute name="createUpdateDefault"
                                              type="xsd:string"/>

                                </xsd:extension>
                              </xsd:simpleContent>
                            </xsd:complexType>
                          </xsd:element>
                        </xsd:sequence>
                      </xsd:complexType>
```

```
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="Standard" nillable="false" maxOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="OnsetLocalDate" type="xsd:string"
                         maxOccurs="1"/>
            <xsd:element name="OffsetFrom" type="xsd:integer"
                         maxOccurs="1"/>
            <xsd:element name="OffsetTo" type="xsd:integer"
                         maxOccurs="1"/>
            <xsd:element name="Onset" minOccurs="0">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="Frequency" type="xsd:string"/>
                  <xsd:element name="Interval" type="xsd:integer"/>
                  <xsd:element name="Month" type="xsd:string"/>
                  <xsd:element name="Day" type="xsd:string"/>
                  <xsd:element name="DayOffset" type="xsd:string"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="DayLight">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="OnsetLocalDate" type="xsd:string"
                         maxOccurs="1"/>
            <xsd:element name="OffsetFrom" type="xsd:integer"
                         maxOccurs="1"/>
            <xsd:element name="OffsetTo" type="xsd:integer"
                         maxOccurs="1"/>
            <xsd:element name="Onset" minOccurs="0">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="Frequency" type="xsd:string"/>
                  <xsd:element name="Interval" type="xsd:integer"/>
                  <xsd:element name="Month" type="xsd:string"/>
                  <xsd:element name="Day" type="xsd:string"/>
                  <xsd:element name="DayOffset" type="xsd:string"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

Example D–6   XLST for Transforming Windows Time Zones

```
<?xml version="1.0" encoding="windows-1252" standalone="yes" ?>
<!-- Header:
fts/connectors/exchange2007/oracle/bdss/connectors/exchange2007/pimtransport/trans
form/exchange/calendar/intermedatetranslation/timezoneregistry.xml -->
<!--
 Copyright (c) 2008, 2009, Oracle and/or its affiliates. All rights reserved.

   NAME
     timezoneregistry.xml - maps Java and Exchange time zones

   DESCRIPTION
   This XML file maps Java time zone identifiers to the time zone identifiers
   used by Exchange 2007.

   The following list enumerates the Java time zones that are not present in
   this file because there are no corresponding time zone for Exchange:
   2) All GMT-09 that do not observe daylight savings time:
      Pacific/Gambier, Pacific/Marquesas, Pacific/Palau
   3) All GMT-10 that observe daylight savings time:
      US/Aleutian, America/Atka, America/Adak
   4) All GMT-02 that do not observe daylight savings time:
      Atlantic/South_Georgia
      Brazil/DeNoronha
      America/Noronha
   5) All GMT+14
      Pacific/Kiritimati
   6) All GMT-08 that do not observe daylight savings time:
      Pacific/Pitcairn
   7) All legacy Etc/GMTx timezones

   The time zone entries must conform to the schema defined in
timezoneregistryschema.xsd

  Each time zone mapping has a <TimeZone> parent element and four child elements:
   <Java>      contains all the Java time zones.  This child element has one or more
               <TimeZoneId> elements that contains a Java time zone id.  If there is
               more than one, then one must have a "extractionDefault" attribute
with
               a value of "true"
   <Exchange> contains the Exchange time zone(s).  This child element has at least
               one <TimeZoneId> element containing an Exchange time zone id.  If
there
               is more than one, a "createUpdateDefault" attribute must be specified
               on one of the values.
   <Standard> contains a description of when standard time begins
   <DayLight> contains a description of when daylight time begins

   The <Standard> and <DayLight> elements contain sub-elements as described below:
   OnsetLocalDate - The local date/time when the time sub-component
(Standard/Daylight) begins
   OffsetTo - The offset that becomes effective on the OnsetLocalDate
   OffsetFrom - The offset that was effective prior to the OnsetLocalDate
   Onset - describes a recurrence pattern that the sub-component occurs
      Frequency - Yearly
      Interval - 1 = Every, 2 = Every Other, etc.  If Frequency == Yearly and
                  interval == 1, interpreted as Every Year; if interval == 2,
                  interpreted as Every Other Year, etc.
      Month    - January..December
      Day      - Sunday..Saturday
```

```
            DayOffset - First, Second, Third, Fourth, Last

    NOTES
    EWS reports a warning if any TZ is used other than PST.  In some cases, EWS
    creates the record and it appears in the correct time slot and in other cases
    it does not.  Regardless of how the record appears in outlook, the connector
fails
    the create/update request because of the warning.  Microsoft is looking in to
    this.

    Currently, only have > 1 Exchange time zone for the PST time zone.  The
Exchange
    time zones stamped on Exchange records come from the system registry of the
client
    machine running MS Outlook.  These time zones are not consistant across Windows
    OS versions and service packs.  When MS clears up the TZ issue, this file most
    likely needs to be updated.

    MODIFIED   (MM/DD/YY)
        lharding 06/08/09 - Initial Version
-->
<TimeZoneMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="timezoneregistryschema.xsd">
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Kwajalein</TimeZoneId>
                <TimeZoneId extractionDefault="true">Pacific/Kwajalein</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-12:00) International Dateline West</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>-1200</OffsetFrom>
            <OffsetTo>-1200</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>-1200</OffsetFrom>
            <OffsetTo>-1200</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Pacific/Samoa</TimeZoneId>
                <TimeZoneId>Pacific/Pago_Pago</TimeZoneId>
                <TimeZoneId extractionDefault="true">US/Samoa</TimeZoneId>
                <TimeZoneId>Pacific/Midway</TimeZoneId>
                <TimeZoneId>Pacific/Niue</TimeZoneId>
                <TimeZoneId>Pacific/Apia</TimeZoneId>
                <TimeZoneId>MIT</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
```

```
                        <TimeZoneId>(GMT-11:00) Midway Island, Samoa</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>-1100</OffsetFrom>
                    <OffsetTo>-1100</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>-1100</OffsetFrom>
                    <OffsetTo>-1100</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId>Pacific/Honolulu</TimeZoneId>
                        <TimeZoneId>HST</TimeZoneId>
                        <TimeZoneId extractionDefault="true">US/Hawaii</TimeZoneId>
                        <TimeZoneId>Pacific/Johnston</TimeZoneId>
                        <TimeZoneId>Pacific/Tahiti</TimeZoneId>
                        <TimeZoneId>Pacific/Fakaofo</TimeZoneId>
                        <TimeZoneId>Pacific/Rarotonga</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT-10:00) Hawaii</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>-1000</OffsetFrom>
                    <OffsetTo>-1000</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>-1000</OffsetFrom>
                    <OffsetTo>-1000</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId>America/Anchorage</TimeZoneId>
                        <TimeZoneId>AST</TimeZoneId>
                        <TimeZoneId>America/Yakutat</TimeZoneId>
                        <TimeZoneId>America/Juneau</TimeZoneId>
                        <TimeZoneId>America/Nome</TimeZoneId>
                        <TimeZoneId extractionDefault="true">US/Alaska</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT-09:00) Alaska</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T020000</OnsetLocalDate>
```

```
            <OffsetFrom>-0800</OffsetFrom>
            <OffsetTo>-0900</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0900</OffsetFrom>
            <OffsetTo>-0800</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>First</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>US/Pacific-New</TimeZoneId>
                <TimeZoneId>PST</TimeZoneId>
                <TimeZoneId>Canada/Pacific</TimeZoneId>
                <TimeZoneId>America/Dawson</TimeZoneId>
                <TimeZoneId>America/Vancouver</TimeZoneId>
                <TimeZoneId extractionDefault="true">US/Pacific</TimeZoneId>
                <TimeZoneId>PST8PDT</TimeZoneId>
                <TimeZoneId>America/Los_Angeles</TimeZoneId>
                <TimeZoneId>America/Ensenada</TimeZoneId>
                <TimeZoneId>Canada/Yukon</TimeZoneId>
                <TimeZoneId>America/Whitehorse</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-08:00) Pacific Time (US &amp; Canada)</TimeZoneId>
                <TimeZoneId createUpdateDefault="true">(GMT-08:00) Pacific Time (US
&amp; Canada); Tijuana</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0700</OffsetFrom>
            <OffsetTo>-0800</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
```

```
                        <OffsetFrom>-0800</OffsetFrom>
                        <OffsetTo>-0700</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>April</Month>
                            <Day>Sunday</Day>
                            <DayOffset>First</DayOffset>
                        </Onset>
                    </DayLight>
                </TimeZone>
                <TimeZone>
                    <Java>
                        <TimeZoneIds>
                            <TimeZoneId extractionDefault="true">America/Tijuana</TimeZoneId>
                            <TimeZoneId>Mexico/BajaNorte</TimeZoneId>
                            <TimeZoneId>Mexico/BajaSur</TimeZoneId>
                        </TimeZoneIds>
                    </Java>
                    <Exchange>
                        <TimeZoneIds>
                            <TimeZoneId>(GMT-08:00) Tijuana, Baja California</TimeZoneId>
                        </TimeZoneIds>
                    </Exchange>
                    <Standard>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>-0700</OffsetFrom>
                        <OffsetTo>-0800</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>October</Month>
                            <Day>Sunday</Day>
                            <DayOffset>Last</DayOffset>
                        </Onset>
                    </Standard>
                    <DayLight>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>-0800</OffsetFrom>
                        <OffsetTo>-0700</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>April</Month>
                            <Day>Sunday</Day>
                            <DayOffset>First</DayOffset>
                        </Onset>
                    </DayLight>
                </TimeZone>
                <TimeZone>
                    <Java>
                        <TimeZoneIds>
                            <TimeZoneId extractionDefault="true">US/Arizona</TimeZoneId>
                            <TimeZoneId>America/Phoenix</TimeZoneId>
                            <TimeZoneId>PNT</TimeZoneId>
                            <TimeZoneId>America/Mermosillo</TimeZoneId>
                            <TimeZoneId>MST</TimeZoneId>
                        </TimeZoneIds>
                    </Java>
                    <Exchange>
```

```
            <TimeZoneIds>
                <TimeZoneId>(GMT-07:00) Arizona</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>-0700</OffsetFrom>
            <OffsetTo>-0700</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>-0700</OffsetFrom>
            <OffsetTo>-0700</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>America/Chihuahua</TimeZoneId>
                <TimeZoneId extractionDefault="true">America/Mazatlan</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-07:00) Chihuahua, La Paz, Mazatlan</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0600</OffsetFrom>
            <OffsetTo>-0700</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0700</OffsetFrom>
            <OffsetTo>-0600</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>First</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>America/Inuvik</TimeZoneId>
                <TimeZoneId>America/Cambridge_Bay</TimeZoneId>
                <TimeZoneId extractionDefault="true">US/Mountain</TimeZoneId>
                <TimeZoneId>America/Denver</TimeZoneId>
                <TimeZoneId>America/Edmonton</TimeZoneId>
```

```
                        <TimeZoneId>America/Dawson_Creek</TimeZoneId>
                        <TimeZoneId>America/Yellowknife</TimeZoneId>
                        <TimeZoneId>MST7MDT</TimeZoneId>
                        <TimeZoneId>Navajo</TimeZoneId>
                        <TimeZoneId>America/Boise</TimeZoneId>
                        <TimeZoneId>America/Shiprock</TimeZoneId>
                        <TimeZoneId>Canada/Mountain</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT-07:00) Mountain Time (US &amp; Canada)</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                    <OffsetFrom>-0600</OffsetFrom>
                    <OffsetTo>-0700</OffsetTo>
                    <Onset>
                        <Frequency>Yearly</Frequency>
                        <Interval>1</Interval>
                        <Month>October</Month>
                        <Day>Sunday</Day>
                        <DayOffset>Last</DayOffset>
                    </Onset>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                    <OffsetFrom>-0700</OffsetFrom>
                    <OffsetTo>-0600</OffsetTo>
                    <Onset>
                        <Frequency>Yearly</Frequency>
                        <Interval>1</Interval>
                        <Month>April</Month>
                        <Day>Sunday</Day>
                        <DayOffset>First</DayOffset>
                    </Onset>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId extractionDefault="true">America/El_Salvador</TimeZoneId>
                        <TimeZoneId>America/Tegucigalpa</TimeZoneId>
                        <TimeZoneId>America/Managua</TimeZoneId>
                        <TimeZoneId>America/Belize</TimeZoneId>
                        <TimeZoneId>America/Regina</TimeZoneId>
                        <TimeZoneId>America/Guatemala</TimeZoneId>
                        <TimeZoneId>America/Costa_Rica</TimeZoneId>
                        <TimeZoneId>Pacific/Galapagos</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT-06:00) Central America</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>-0600</OffsetFrom>
```

```
            <OffsetTo>-0600</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>-0600</OffsetFrom>
            <OffsetTo>-0600</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>US/Indiana-Starke</TimeZoneId>
                <TimeZoneId>America/North_Dakota/New_Salem</TimeZoneId>
                <TimeZoneId>CST6CDT</TimeZoneId>
                <TimeZoneId>America/Indiana/Tell_City</TimeZoneId>
                <TimeZoneId extractionDefault="true">US/Central</TimeZoneId>
                <TimeZoneId>CST</TimeZoneId>
                <TimeZoneId>Canada/Central</TimeZoneId>
                <TimeZoneId>America/Winnipeg</TimeZoneId>
                <TimeZoneId>America/Rainy_River</TimeZoneId>
                <TimeZoneId>America/Chicago</TimeZoneId>
                <TimeZoneId>America/Rankin_Inlet</TimeZoneId>
                <TimeZoneId>America/Knox_IN</TimeZoneId>
                <TimeZoneId>America/Merida</TimeZoneId>
                <TimeZoneId>America/Cancun</TimeZoneId>
                <TimeZoneId>America/Menominee</TimeZoneId>
                <TimeZoneId>America/Indiana/Knox</TimeZoneId>
                <TimeZoneId>America/North_Dakota/Center</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-06:00) Central Time (US &amp; Canada)</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0500</OffsetFrom>
            <OffsetTo>-0600</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0600</OffsetFrom>
            <OffsetTo>-0500</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>First</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
```

```
<TimeZone>
   <Java>
      <TimeZoneIds>
         <TimeZoneId extractionDefault="true">America/Mexico_City</TimeZoneId>
         <TimeZoneId>Mexico/General</TimeZoneId>
         <TimeZoneId>Chile/EasterIsland</TimeZoneId>
         <TimeZoneId>Pacific/Easter</TimeZoneId>
         <TimeZoneId>America/Monterrey</TimeZoneId>
      </TimeZoneIds>
   </Java>
   <Exchange>
      <TimeZoneIds>
         <TimeZoneId>(GMT-06:00) Guadalajara, Mexico City,
Monterrey</TimeZoneId>
      </TimeZoneIds>
   </Exchange>
   <Standard>
      <OnsetLocalDate>16010101T020000</OnsetLocalDate>
      <OffsetFrom>-0500</OffsetFrom>
      <OffsetTo>-0600</OffsetTo>
      <Onset>
         <Frequency>Yearly</Frequency>
         <Interval>1</Interval>
         <Month>October</Month>
         <Day>Sunday</Day>
         <DayOffset>Last</DayOffset>
      </Onset>
   </Standard>
   <DayLight>
      <OnsetLocalDate>16010101T020000</OnsetLocalDate>
      <OffsetFrom>-0600</OffsetFrom>
      <OffsetTo>-0500</OffsetTo>
      <Onset>
         <Frequency>Yearly</Frequency>
         <Interval>1</Interval>
         <Month>April</Month>
         <Day>Sunday</Day>
         <DayOffset>First</DayOffset>
      </Onset>
   </DayLight>
</TimeZone>
<TimeZone>
   <Java>
      <TimeZoneIds>
         <TimeZoneId>Canada/East-Saskatchewan</TimeZoneId>
         <TimeZoneId extractionDefault="true">Canada/Saskatchewan</TimeZoneId>
         <TimeZoneId>America/Swift_Current</TimeZoneId>
      </TimeZoneIds>
   </Java>
   <Exchange>
      <TimeZoneIds>
         <TimeZoneId>(GMT-06:00) Saskatchewan</TimeZoneId>
      </TimeZoneIds>
   </Exchange>
   <Standard>
      <OnsetLocalDate>16010101T000000</OnsetLocalDate>
      <OffsetFrom>-0600</OffsetFrom>
      <OffsetTo>-0600</OffsetTo>
   </Standard>
   <DayLight>
```

```
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>-0600</OffsetFrom>
                    <OffsetTo>-0600</OffsetTo>
                </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>America/Bogota</TimeZoneId>
                    <TimeZoneId extractionDefault="true">America/Lima</TimeZoneId>
                    <TimeZoneId>America/Jamaica</TimeZoneId>
                    <TimeZoneId>America/Panama</TimeZoneId>
                    <TimeZoneId>America/Coral_Harbour</TimeZoneId>
                    <TimeZoneId>America/Port-au-Prince</TimeZoneId>
                    <TimeZoneId>America/Cayman</TimeZoneId>
                    <TimeZoneId>America/Atikokan</TimeZoneId>
                    <TimeZoneId>America/Resolute</TimeZoneId>
                    <TimeZoneId>America/Guayaquil</TimeZoneId>
                    <TimeZoneId>Jamaica</TimeZoneId>
                    <TimeZoneId>EST</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT-05:00) Bogota, Lima, Quito, Rio Branco</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>-0500</OffsetFrom>
                <OffsetTo>-0500</OffsetTo>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>-0500</OffsetFrom>
                <OffsetTo>-0500</OffsetTo>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId extractionDefault="true">US/Eastern</TimeZoneId>
                    <TimeZoneId>America/Nipigon</TimeZoneId>
                    <TimeZoneId>US/Michigan</TimeZoneId>
                    <TimeZoneId>America/New_York</TimeZoneId>
                    <TimeZoneId>America/Kentucky/Monticello</TimeZoneId>
                    <TimeZoneId>America/Detroit</TimeZoneId>
                    <TimeZoneId>America/Indiana/Vevay</TimeZoneId>
                    <TimeZoneId>America/Nassau</TimeZoneId>
                    <TimeZoneId>EST5EDT</TimeZoneId>
                    <TimeZoneId>Canada/Eastern</TimeZoneId>
                    <TimeZoneId>America/Indiana/Marengo</TimeZoneId>
                    <TimeZoneId>America/Louisville</TimeZoneId>
                    <TimeZoneId>America/Fort_Wayne</TimeZoneId>
                    <TimeZoneId>America/Indiana/Winamac</TimeZoneId>
                    <TimeZoneId>America/Pangnirtung</TimeZoneId>
                    <TimeZoneId>America/Thunder_Bay</TimeZoneId>
                    <TimeZoneId>America/Grand_Turk</TimeZoneId>
                    <TimeZoneId>America/Toronto</TimeZoneId>
                    <TimeZoneId>IET</TimeZoneId>
```

```
                <TimeZoneId>America/Indiana/Petersburg</TimeZoneId>
                <TimeZoneId>America/Kentucky/Louisville</TimeZoneId>
                <TimeZoneId>America/Indiana/Vincennes</TimeZoneId>
                <TimeZoneId>America/Iqaluit</TimeZoneId>
                <TimeZoneId>America/Montreal</TimeZoneId>
                <TimeZoneId>Cuba</TimeZoneId>
                <TimeZoneId>America/Havana</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-05:00) Eastern Time (US &amp; Canada)</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0400</OffsetFrom>
            <OffsetTo>-0500</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0500</OffsetFrom>
            <OffsetTo>-0400</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>First</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>America/Indianapolis</TimeZoneId>
                <TimeZoneId>America/Indiana/Indianapolis</TimeZoneId>
                <TimeZoneId extractionDefault="true">US/East-Indiana</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-05:00) Indiana (East)</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>-0500</OffsetFrom>
            <OffsetTo>-0500</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>-0500</OffsetFrom>
```

```
                <OffsetTo>-0500</OffsetTo>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>America/Caracas</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT-04:30) Caracas</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>-0430</OffsetFrom>
                <OffsetTo>-0430</OffsetTo>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>-0430</OffsetFrom>
                <OffsetTo>-0430</OffsetTo>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>America/Montreal</TimeZoneId>
                    <TimeZoneId extractionDefault="true">Canada/Atlantic</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT-04:00) Atlantic Time (Canada)</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                <OffsetFrom>-0300</OffsetFrom>
                <OffsetTo>-0400</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>October</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                <OffsetFrom>-0400</OffsetFrom>
                <OffsetTo>-0300</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>First</DayOffset>
                </Onset>
```

```
                        </DayLight>
                    </TimeZone>
                    <TimeZone>
                        <Java>
                            <TimeZoneIds>
                                <TimeZoneId extractionDefault="true">America/La_Paz</TimeZoneId>
                                <TimeZoneId>America/Martinique</TimeZoneId>
                                <TimeZoneId>America/St_Lucia</TimeZoneId>
                                <TimeZoneId>America/Aruba</TimeZoneId>
                                <TimeZoneId>America/Dominica</TimeZoneId>
                                <TimeZoneId>America/St_Vincent</TimeZoneId>
                                <TimeZoneId>America/Puerto_Rico</TimeZoneId>
                                <TimeZoneId>America/Tortola</TimeZoneId>
                                <TimeZoneId>America/Barbados</TimeZoneId>
                                <TimeZoneId>America/Grenada</TimeZoneId>
                                <TimeZoneId>America/Port_of_Spain</TimeZoneId>
                                <TimeZoneId>America/St_Thomas</TimeZoneId>
                                <TimeZoneId>America/Virgin</TimeZoneId>
                                <TimeZoneId>America/St_Barthelemy</TimeZoneId>
                                <TimeZoneId>America/St_Kitts</TimeZoneId>
                                <TimeZoneId>America/Curacao</TimeZoneId>
                                <TimeZoneId>America/Antigua</TimeZoneId>
                                <TimeZoneId>America/Blanc-Sablon</TimeZoneId>
                                <TimeZoneId>America/Guadeloupe</TimeZoneId>
                                <TimeZoneId>America/Marigot</TimeZoneId>
                                <TimeZoneId>America/Montserrat</TimeZoneId>
                                <TimeZoneId>America/Anguilla</TimeZoneId>
                                <TimeZoneId>America/Santo_Domingo</TimeZoneId>
                                <TimeZoneId>PRT</TimeZoneId>
                                <TimeZoneId>America/Rio_Branco</TimeZoneId>
                                <TimeZoneId>America/Eirunepe</TimeZoneId>
                                <TimeZoneId>America/Boa_Vista</TimeZoneId>
                                <TimeZoneId>America/Porto_Acre</TimeZoneId>
                                <TimeZoneId>America/Porto_Velho</TimeZoneId>
                                <TimeZoneId>America/Guyana</TimeZoneId>
                                <TimeZoneId>Brazil/Acre</TimeZoneId>
                                <TimeZoneId>Brazil/West</TimeZoneId>
                            </TimeZoneIds>
                        </Java>
                        <Exchange>
                            <TimeZoneIds>
                                <TimeZoneId>(GMT-04:00) La Paz</TimeZoneId>
                            </TimeZoneIds>
                        </Exchange>
                        <Standard>
                            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                            <OffsetFrom>-0400</OffsetFrom>
                            <OffsetTo>-0400</OffsetTo>
                        </Standard>
                        <DayLight>
                            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                            <OffsetFrom>-0400</OffsetFrom>
                            <OffsetTo>-0400</OffsetTo>
                        </DayLight>
                    </TimeZone>
                    <TimeZone>
                        <Java>
                            <TimeZoneIds>
                                <TimeZoneId>America/Manaus</TimeZoneId>
                            </TimeZoneIds>
```

```
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-04:00) Manaus</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0300</OffsetFrom>
            <OffsetTo>-0400</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0400</OffsetFrom>
            <OffsetTo>-0300</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>First</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId extractionDefault="true">Chile/Continental</TimeZoneId>
                <TimeZoneId>Antarctica/Palmer</TimeZoneId>
                <TimeZoneId>America/Thule</TimeZoneId>
                <TimeZoneId>America/Asuncion</TimeZoneId>
                <TimeZoneId>Atlantic/Bermuda</TimeZoneId>
                <TimeZoneId>America/Glace_Bay</TimeZoneId>
                <TimeZoneId>America/Halifax</TimeZoneId>
                <TimeZoneId>America/Moncton</TimeZoneId>
                <TimeZoneId>America/Goose_Bay</TimeZoneId>
                <TimeZoneId>America/Cuiaba</TimeZoneId>
                <TimeZoneId>America/Campo_Grande</TimeZoneId>
                <TimeZoneId>Atlantic/Stanley</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-04:00) Santiago</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>-0300</OffsetFrom>
            <OffsetTo>-0400</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
```

```
            <Month>March</Month>
            <Day>Saturday</Day>
            <DayOffset>Second</DayOffset>
        </Onset>
    </Standard>
    <DayLight>
        <OnsetLocalDate>16010101T000000</OnsetLocalDate>
        <OffsetFrom>-0400</OffsetFrom>
        <OffsetTo>-0300</OffsetTo>
        <Onset>
            <Frequency>Yearly</Frequency>
            <Interval>1</Interval>
            <Month>October</Month>
            <Day>Saturday</Day>
            <DayOffset>Second</DayOffset>
        </Onset>
    </DayLight>
</TimeZone>
<TimeZone>
    <Java>
        <TimeZoneIds>
            <TimeZoneId>America/St_Johns</TimeZoneId>
            <TimeZoneId extractionDefault="true">CNT</TimeZoneId>
            <TimeZoneId>Canada/Newfoundland</TimeZoneId>
        </TimeZoneIds>
    </Java>
    <Exchange>
        <TimeZoneIds>
            <TimeZoneId>(GMT-03:30) Newfoundland</TimeZoneId>
        </TimeZoneIds>
    </Exchange>
    <Standard>
        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
        <OffsetFrom>-0230</OffsetFrom>
        <OffsetTo>-0330</OffsetTo>
        <Onset>
            <Frequency>Yearly</Frequency>
            <Interval>1</Interval>
            <Month>October</Month>
            <Day>Sunday</Day>
            <DayOffset>Last</DayOffset>
        </Onset>
    </Standard>
    <DayLight>
        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
        <OffsetFrom>-0330</OffsetFrom>
        <OffsetTo>-0230</OffsetTo>
        <Onset>
            <Frequency>Yearly</Frequency>
            <Interval>1</Interval>
            <Month>April</Month>
            <Day>Sunday</Day>
            <DayOffset>First</DayOffset>
        </Onset>
    </DayLight>
</TimeZone>
<TimeZone>
    <Java>
        <TimeZoneIds>
            <TimeZoneId extractionDefault="true">BET</TimeZoneId>
```

```
            <TimeZoneId>Brazil/East</TimeZoneId>
            <TimeZoneId>America/Sao_Paulo</TimeZoneId>
            <TimeZoneId>America/Miquelon</TimeZoneId>
        </TimeZoneIds>
    </Java>
    <Exchange>
        <TimeZoneIds>
            <TimeZoneId>(GMT-03:00) Brasilia</TimeZoneId>
        </TimeZoneIds>
    </Exchange>
    <Standard>
        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
        <OffsetFrom>-0200</OffsetFrom>
        <OffsetTo>-0300</OffsetTo>
        <Onset>
            <Frequency>Yearly</Frequency>
            <Interval>1</Interval>
            <Month>February</Month>
            <Day>Sunday</Day>
            <DayOffset>Second</DayOffset>
        </Onset>
    </Standard>
    <DayLight>
        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
        <OffsetFrom>-0300</OffsetFrom>
        <OffsetTo>-0200</OffsetTo>
        <Onset>
            <Frequency>Yearly</Frequency>
            <Interval>1</Interval>
            <Month>October</Month>
            <Day>Sunday</Day>
            <DayOffset>Third</DayOffset>
        </Onset>
    </DayLight>
</TimeZone>
<TimeZone>
    <Java>
        <TimeZoneIds>
            <TimeZoneId extractionDefault="true">America/Buenos_Aires</TimeZoneId>
            <TimeZoneId>America/Rosario</TimeZoneId>
            <TimeZoneId>America/Argentina/Cordoba</TimeZoneId>
            <TimeZoneId>America/Cordoba</TimeZoneId>
            <TimeZoneId>America/Argentina/Tucuman</TimeZoneId>
            <TimeZoneId>AGT</TimeZoneId>
            <TimeZoneId>America/Argentina/Buenos_Aires</TimeZoneId>
        </TimeZoneIds>
    </Java>
    <Exchange>
        <TimeZoneIds>
            <TimeZoneId>(GMT-03:00) Buenos Aires</TimeZoneId>
        </TimeZoneIds>
    </Exchange>
    <Standard>
        <OnsetLocalDate>16010101T000000</OnsetLocalDate>
        <OffsetFrom>-0200</OffsetFrom>
        <OffsetTo>-0300</OffsetTo>
    </Standard>
    <DayLight>
        <OnsetLocalDate>16010101T000000</OnsetLocalDate>
        <OffsetFrom>-0300</OffsetFrom>
```

```
                    <OffsetTo>-0200</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId>America/Montevideo</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT-03:00) Montevideo</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>-0200</OffsetFrom>
                    <OffsetTo>-0300</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>-0300</OffsetFrom>
                    <OffsetTo>-0200</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId extractionDefault="true">America/Paramaribo</TimeZoneId>
                        <TimeZoneId>America/Recife</TimeZoneId>
                        <TimeZoneId>America/Fortaleza</TimeZoneId>
                        <TimeZoneId>America/Bahia</TimeZoneId>
                        <TimeZoneId>America/Araguaina</TimeZoneId>
                        <TimeZoneId>America/Santarem</TimeZoneId>
                        <TimeZoneId>America/Belem</TimeZoneId>
                        <TimeZoneId>America/Maceio</TimeZoneId>
                        <TimeZoneId>America/Mendoza</TimeZoneId>
                        <TimeZoneId>America/Catamarca</TimeZoneId>
                        <TimeZoneId>America/Jujuy</TimeZoneId>
                        <TimeZoneId>America/Cayenne</TimeZoneId>
                        <TimeZoneId>America/Argentina/Catamarca</TimeZoneId>
                        <TimeZoneId>America/Argentina/Mendoza</TimeZoneId>
                        <TimeZoneId>America/Argentina/San_Juan</TimeZoneId>
                        <TimeZoneId>America/Argentina/Jujuy</TimeZoneId>
                        <TimeZoneId>America/Argentina/San_Luis</TimeZoneId>
                        <TimeZoneId>America/Argentina/La_Rioja</TimeZoneId>
                        <TimeZoneId>America/Argentina/ComodRivadavia</TimeZoneId>
                        <TimeZoneId>America/Argentina/Ushuaia</TimeZoneId>
                        <TimeZoneId>America/Argentina/Rio_Gallegos. Salta</TimeZoneId>
                        <TimeZoneId>Antarctica/Rothera</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT-03:00) Georgetown</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>-0300</OffsetFrom>
```

```
            <OffsetTo>-0300</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>-0300</OffsetFrom>
            <OffsetTo>-0300</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>America/Godthab</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-03:00) Greenland</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0200</OffsetFrom>
            <OffsetTo>-0300</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0300</OffsetFrom>
            <OffsetTo>-0200</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>First</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
    <Java>
    <TimeZoneIds>
    <TimeZoneId></TimeZoneId>
    </TimeZoneIds>
    </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT-02:00) Mid-Atlantic</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>-0100</OffsetFrom>
            <OffsetTo>-0200</OffsetTo>
            <Onset>
```

```
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>September</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                <OffsetFrom>-0200</OffsetFrom>
                <OffsetTo>-0100</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId extractionDefault="true">Atlantic/Azores</TimeZoneId>
                    <TimeZoneId>America/Scoresbysund</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT-01:00) Azores</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                <OffsetFrom>+0000</OffsetFrom>
                <OffsetTo>-0100</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>October</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                <OffsetFrom>-0100</OffsetFrom>
                <OffsetTo>+0000</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
```

```
            <TimeZoneId>Atlantic/Cape_Verde</TimeZoneId>
        </TimeZoneIds>
    </Java>
    <Exchange>
        <TimeZoneIds>
            <TimeZoneId>(GMT-01:00) Cape Verde Is.</TimeZoneId>
        </TimeZoneIds>
    </Exchange>
    <Standard>
        <OnsetLocalDate>16010101T000000</OnsetLocalDate>
        <OffsetFrom>-0100</OffsetFrom>
        <OffsetTo>-0100</OffsetTo>
    </Standard>
    <DayLight>
        <OnsetLocalDate>16010101T000000</OnsetLocalDate>
        <OffsetFrom>-0100</OffsetFrom>
        <OffsetTo>-0100</OffsetTo>
    </DayLight>
</TimeZone>
<TimeZone>
    <Java>
        <TimeZoneIds>
            <TimeZoneId>Africa/El_Aaiun</TimeZoneId>
            <TimeZoneId>Africa/Casablanca</TimeZoneId>
            <TimeZoneId>Greenwich</TimeZoneId>
            <TimeZoneId>Africa/Conakry</TimeZoneId>
            <TimeZoneId>Africa/Bissau</TimeZoneId>
            <TimeZoneId>Africa/Timbuktu</TimeZoneId>
            <TimeZoneId>Africa/Abidjan</TimeZoneId>
            <TimeZoneId>Africa/Freetown</TimeZoneId>
            <TimeZoneId>Africa/Ouagadougou</TimeZoneId>
            <TimeZoneId>Africa/Bamako</TimeZoneId>
            <TimeZoneId extractionDefault="true">GMT</TimeZoneId>
            <TimeZoneId>Atlantic/St_Helena</TimeZoneId>
            <TimeZoneId>Africa/Dakar</TimeZoneId>
            <TimeZoneId>Africa/Banjul</TimeZoneId>
            <TimeZoneId>America/Danmarkshavn</TimeZoneId>
            <TimeZoneId>Africa/Sao_Tome</TimeZoneId>
            <TimeZoneId>Iceland</TimeZoneId>
            <TimeZoneId>Africa/Lome</TimeZoneId>
            <TimeZoneId>Africa/Nouakchott</TimeZoneId>
            <TimeZoneId>UCT</TimeZoneId>
            <TimeZoneId>Universal</TimeZoneId>
            <TimeZoneId>UTC</TimeZoneId>
            <TimeZoneId>Zulu</TimeZoneId>
            <TimeZoneId>Africa/Accra</TimeZoneId>
        </TimeZoneIds>
    </Java>
    <Exchange>
        <TimeZoneIds>
            <TimeZoneId>(GMT) Casablanca</TimeZoneId>
        </TimeZoneIds>
    </Exchange>
    <Standard>
        <OnsetLocalDate>16010101T000000</OnsetLocalDate>
        <OffsetFrom>+0000</OffsetFrom>
        <OffsetTo>+0000</OffsetTo>
    </Standard>
    <DayLight>
        <OnsetLocalDate>16010101T000000</OnsetLocalDate>
```

```
                                <OffsetFrom>+0000</OffsetFrom>
                                <OffsetTo>+0000</OffsetTo>
                            </DayLight>
                        </TimeZone>
                        <TimeZone>
                            <Java>
                                <TimeZoneIds>
                                    <TimeZoneId extractionDefault="true">Africa/Monrovia</TimeZoneId>
                                    <TimeZoneId>Atlantic/Reykjavik</TimeZoneId>
                                </TimeZoneIds>
                            </Java>
                            <Exchange>
                                <TimeZoneIds>
                                    <TimeZoneId>(GMT) Monrovia, Reykjavik</TimeZoneId>
                                </TimeZoneIds>
                            </Exchange>
                            <Standard>
                                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                                <OffsetFrom>+0000</OffsetFrom>
                                <OffsetTo>+0000</OffsetTo>
                            </Standard>
                            <DayLight>
                                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                                <OffsetFrom>+0000</OffsetFrom>
                                <OffsetTo>+0000</OffsetTo>
                            </DayLight>
                        </TimeZone>
                        <TimeZone>
                            <Java>
                                <TimeZoneIds>
                                    <TimeZoneId extractionDefault="true">GB</TimeZoneId>
                                    <TimeZoneId>Europe/Guernsey</TimeZoneId>
                                    <TimeZoneId>GB-Eire</TimeZoneId>
                                    <TimeZoneId>Europe/Jersey</TimeZoneId>
                                    <TimeZoneId>Eire</TimeZoneId>
                                    <TimeZoneId>Europe/Dublin</TimeZoneId>
                                    <TimeZoneId>Europe/Belfast</TimeZoneId>
                                    <TimeZoneId>Europe/London</TimeZoneId>
                                    <TimeZoneId>Europe/Isle_of_Man</TimeZoneId>
                                    <TimeZoneId>Portugal</TimeZoneId>
                                    <TimeZoneId>Europe/Lisbon</TimeZoneId>
                                    <TimeZoneId>Atlantic/Faroe</TimeZoneId>
                                    <TimeZoneId>Atlantic/Canary</TimeZoneId>
                                    <TimeZoneId>Atlantic/Faeroe</TimeZoneId>
                                    <TimeZoneId>Atlantic/Madeira</TimeZoneId>
                                    <TimeZoneId>WET</TimeZoneId>
                                </TimeZoneIds>
                            </Java>
                            <Exchange>
                                <TimeZoneIds>
                                    <TimeZoneId>(GMT) Greenwich Mean Time : Dublin, Edinburgh, Lisbon,
London</TimeZoneId>
                                </TimeZoneIds>
                            </Exchange>
                            <Standard>
                                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                                <OffsetFrom>+0100</OffsetFrom>
                                <OffsetTo>+0000</OffsetTo>
                                <Onset>
                                    <Frequency>Yearly</Frequency>
```

```
            <Interval>1</Interval>
            <Month>October</Month>
            <Day>Sunday</Day>
            <DayOffset>Last</DayOffset>
        </Onset>
    </Standard>
    <DayLight>
        <OnsetLocalDate>16010101T010000</OnsetLocalDate>
        <OffsetFrom>+0000</OffsetFrom>
        <OffsetTo>+0100</OffsetTo>
        <Onset>
            <Frequency>Yearly</Frequency>
            <Interval>1</Interval>
            <Month>April</Month>
            <Day>Sunday</Day>
            <DayOffset>Last</DayOffset>
        </Onset>
    </DayLight>
</TimeZone>
<TimeZone>
    <Java>
        <TimeZoneIds>
            <TimeZoneId>Europe/Prague</TimeZoneId>
            <TimeZoneId>Europe/Budapest</TimeZoneId>
            <TimeZoneId>Europe/Ljubljana</TimeZoneId>
            <TimeZoneId>Europe/Bratislava</TimeZoneId>
            <TimeZoneId extractionDefault="true">Europe/Belgrade</TimeZoneId>
            <TimeZoneId>MET</TimeZoneId>
        </TimeZoneIds>
    </Java>
    <Exchange>
        <TimeZoneIds>
            <TimeZoneId>(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana,
Prague</TimeZoneId>
        </TimeZoneIds>
    </Exchange>
    <Standard>
        <OnsetLocalDate>16010101T030000</OnsetLocalDate>
        <OffsetFrom>+0200</OffsetFrom>
        <OffsetTo>+0100</OffsetTo>
        <Onset>
            <Frequency>Yearly</Frequency>
            <Interval>1</Interval>
            <Month>October</Month>
            <Day>Sunday</Day>
            <DayOffset>Last</DayOffset>
        </Onset>
    </Standard>
    <DayLight>
        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
        <OffsetFrom>+0100</OffsetFrom>
        <OffsetTo>+0200</OffsetTo>
        <Onset>
            <Frequency>Yearly</Frequency>
            <Interval>1</Interval>
            <Month>April</Month>
            <Day>Sunday</Day>
            <DayOffset>Last</DayOffset>
        </Onset>
    </DayLight>
```

```
            </TimeZone>
            <TimeZone>
               <Java>
                  <TimeZoneIds>
                     <TimeZoneId extractionDefault="true">Europe/Brussels</TimeZoneId>
                     <TimeZoneId>Europe/Copenhagen</TimeZoneId>
                     <TimeZoneId>Europe/Madrid</TimeZoneId>
                     <TimeZoneId>Europe/Paris</TimeZoneId>
                  </TimeZoneIds>
               </Java>
               <Exchange>
                  <TimeZoneIds>
                     <TimeZoneId>(GMT+01:00) Brussels, Copenhagen, Madrid,
Paris</TimeZoneId>
                  </TimeZoneIds>
               </Exchange>
               <Standard>
                  <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                  <OffsetFrom>+0200</OffsetFrom>
                  <OffsetTo>+0100</OffsetTo>
                  <Onset>
                     <Frequency>Yearly</Frequency>
                     <Interval>1</Interval>
                     <Month>October</Month>
                     <Day>Sunday</Day>
                     <DayOffset>Last</DayOffset>
                  </Onset>
               </Standard>
               <DayLight>
                  <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                  <OffsetFrom>+0100</OffsetFrom>
                  <OffsetTo>+0200</OffsetTo>
                  <Onset>
                     <Frequency>Yearly</Frequency>
                     <Interval>1</Interval>
                     <Month>April</Month>
                     <Day>Sunday</Day>
                     <DayOffset>Last</DayOffset>
                  </Onset>
               </DayLight>
            </TimeZone>
            <TimeZone>
               <Java>
                  <TimeZoneIds>
                     <TimeZoneId>Europe/Warsaw</TimeZoneId>
                     <TimeZoneId>Europe/Zagreb</TimeZoneId>
                     <TimeZoneId extractionDefault="true">Europe/Sarajevo</TimeZoneId>
                     <TimeZoneId>Europe/Skopje</TimeZoneId>
                  </TimeZoneIds>
               </Java>
               <Exchange>
                  <TimeZoneIds>
                     <TimeZoneId>(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb</TimeZoneId>
                  </TimeZoneIds>
               </Exchange>
               <Standard>
                  <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                  <OffsetFrom>+0200</OffsetFrom>
                  <OffsetTo>+0100</OffsetTo>
                  <Onset>
```

```
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>+0100</OffsetFrom>
            <OffsetTo>+0200</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId extractionDefault="true">Africa/Ndjamena</TimeZoneId>
                <TimeZoneId>Africa/Douala</TimeZoneId>
                <TimeZoneId>Africa/Libreville</TimeZoneId>
                <TimeZoneId>Africa/Brazzaville</TimeZoneId>
                <TimeZoneId>Africa/Lagos</TimeZoneId>
                <TimeZoneId>Africa/Porto-Novo</TimeZoneId>
                <TimeZoneId>Africa/Luanda</TimeZoneId>
                <TimeZoneId>Africa/Kinshasa</TimeZoneId>
                <TimeZoneId>Africa/Niamey</TimeZoneId>
                <TimeZoneId>Africa/Bangui</TimeZoneId>
                <TimeZoneId>Africa/Malabo</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+01:00) West Central Africa</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0100</OffsetFrom>
            <OffsetTo>+0100</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0100</OffsetFrom>
            <OffsetTo>+0100</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Asia/Amman</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
```

```
                    <TimeZoneId>(GMT+02:00) Amman</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T040000</OnsetLocalDate>
                <OffsetFrom>+0300</OffsetFrom>
                <OffsetTo>+0200</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>October</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                <OffsetFrom>+0200</OffsetFrom>
                <OffsetTo>+0300</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId extractionDefault="true">Europe/Athens</TimeZoneId>
                    <TimeZoneId>Europe/Istanbul</TimeZoneId>
                    <TimeZoneId>Europe/Bucharest</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+02:00) Athens, Bucharest, Istanbul</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                <OffsetFrom>+0300</OffsetFrom>
                <OffsetTo>+0200</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>October</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                <OffsetFrom>+0200</OffsetFrom>
                <OffsetTo>+0300</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
```

```
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>Europe/Minsk</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+02:00) Minsk</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                <OffsetFrom>+0300</OffsetFrom>
                <OffsetTo>+0200</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>October</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                <OffsetFrom>+0200</OffsetFrom>
                <OffsetTo>+0300</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>Asia/Beirut</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+02:00) Beirut</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T040000</OnsetLocalDate>
                <OffsetFrom>+0300</OffsetFrom>
                <OffsetTo>+0200</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
```

```
                    <Month>October</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                <OffsetFrom>+0200</OffsetFrom>
                <OffsetTo>+0300</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>Africa/Cairo</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+02:00) Cairo</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                <OffsetFrom>+0300</OffsetFrom>
                <OffsetTo>+0200</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>September</Month>
                    <Day>Wednesday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                <OffsetFrom>+0200</OffsetFrom>
                <OffsetTo>+0300</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>May</Month>
                    <Day>Friday</Day>
                    <DayOffset>First</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>Africa/Gaborone</TimeZoneId>
                    <TimeZoneId>Africa/Lubumbashi</TimeZoneId>
                    <TimeZoneId>Africa/Kigali</TimeZoneId>
```

```
                    <TimeZoneId>Africa/Blantyre</TimeZoneId>
                    <TimeZoneId>Africa/Maputo</TimeZoneId>
                    <TimeZoneId>Africa/Lusaka</TimeZoneId>
                    <TimeZoneId extractionDefault="true">CAT</TimeZoneId>
                    <TimeZoneId>Africa/Harare</TimeZoneId>
                    <TimeZoneId>Africa/Bujumbura</TimeZoneId>
                    <TimeZoneId>Africa/Maseru</TimeZoneId>
                    <TimeZoneId>Africa/Mbabane</TimeZoneId>
                    <TimeZoneId>Africa/Johannesburg</TimeZoneId>
                    <TimeZoneId>Libya</TimeZoneId>
                    <TimeZoneId>Afria/Tripoli</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+02:00) Harare, Pretoria</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>+0200</OffsetFrom>
                <OffsetTo>+0200</OffsetTo>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>+0200</OffsetFrom>
                <OffsetTo>+0200</OffsetTo>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId extractionDefault="true">Europe/Helsinki</TimeZoneId>
                    <TimeZoneId>Europe/Riga</TimeZoneId>
                    <TimeZoneId>Europe/Tallinn</TimeZoneId>
                    <TimeZoneId>Europe/Sofia</TimeZoneId>
                    <TimeZoneId>EET</TimeZoneId>
                    <TimeZoneId>Europe/Uzhgorod</TimeZoneId>
                    <TimeZoneId>Europe/Nicosia</TimeZoneId>
                    <TimeZoneId>Europe/Kaliningrad</TimeZoneId>
                    <TimeZoneId>Asia/Damascus</TimeZoneId>
                    <TimeZoneId>ART</TimeZoneId>
                    <TimeZoneId>Europe/Chisinau</TimeZoneId>
                    <TimeZoneId>Asia/Gaza</TimeZoneId>
                    <TimeZoneId>Europe/Mariehamn</TimeZoneId>
                    <TimeZoneId>Asia/Beirut</TimeZoneId>
                    <TimeZoneId>Asia/Nicosia</TimeZoneId>
                    <TimeZoneId>Turkey</TimeZoneId>
                    <TimeZoneId>Egypt</TimeZoneId>
                    <TimeZoneId>Asia/Amman</TimeZoneId>
                    <TimeZoneId>Europe/Zaporozhye</TimeZoneId>
                    <TimeZoneId>Europe/Simferopol</TimeZoneId>
                    <TimeZoneId>Europe/Tiraspol</TimeZoneId>
                    <TimeZoneId>Europe/Kiev</TimeZoneId>
                    <TimeZoneId>Europe/Vilnius</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn,
```

```
                Vilnius</TimeZoneId>
                        </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T040000</OnsetLocalDate>
                    <OffsetFrom>+0300</OffsetFrom>
                    <OffsetTo>+0200</OffsetTo>
                    <Onset>
                        <Frequency>Yearly</Frequency>
                        <Interval>1</Interval>
                        <Month>October</Month>
                        <Day>Sunday</Day>
                        <DayOffset>Last</DayOffset>
                    </Onset>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                    <OffsetFrom>+0200</OffsetFrom>
                    <OffsetTo>+0300</OffsetTo>
                    <Onset>
                        <Frequency>Yearly</Frequency>
                        <Interval>1</Interval>
                        <Month>April</Month>
                        <Day>Sunday</Day>
                        <DayOffset>Last</DayOffset>
                    </Onset>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId>Asia/Jerusalem</TimeZoneId>
                        <TimeZoneId extractionDefault="true">Israel</TimeZoneId>
                        <TimeZoneId>Asia/Tel_Aviv</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT+02:00) Jerusalem</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0200</OffsetFrom>
                    <OffsetTo>+0200</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0200</OffsetFrom>
                    <OffsetTo>+0200</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId>Africa/Windhoek</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
```

```
        <TimeZoneId>(GMT+02:00) Windhoek</TimeZoneId>
     </TimeZoneIds>
  </Exchange>
  <Standard>
     <OnsetLocalDate>16010101T030000</OnsetLocalDate>
     <OffsetFrom>+0200</OffsetFrom>
     <OffsetTo>+0100</OffsetTo>
     <Onset>
        <Frequency>Yearly</Frequency>
        <Interval>1</Interval>
        <Month>October</Month>
        <Day>Sunday</Day>
        <DayOffset>Last</DayOffset>
     </Onset>
  </Standard>
  <DayLight>
     <OnsetLocalDate>16010101T020000</OnsetLocalDate>
     <OffsetFrom>+0100</OffsetFrom>
     <OffsetTo>+0200</OffsetTo>
     <Onset>
        <Frequency>Yearly</Frequency>
        <Interval>1</Interval>
        <Month>April</Month>
        <Day>Sunday</Day>
        <DayOffset>Last</DayOffset>
     </Onset>
  </DayLight>
</TimeZone>
<TimeZone>
  <Java>
     <TimeZoneIds>
        <TimeZoneId>Asia/Qatar</TimeZoneId>
        <TimeZoneId extractionDefault="true">Asia/Baghdad</TimeZoneId>
        <TimeZoneId>Asia/Bahrain</TimeZoneId>
        <TimeZoneId>Asia/Aden</TimeZoneId>
     </TimeZoneIds>
  </Java>
  <Exchange>
     <TimeZoneIds>
        <TimeZoneId>(GMT+03:00) Baghdad</TimeZoneId>
     </TimeZoneIds>
  </Exchange>
  <Standard>
     <OnsetLocalDate>16010101T040000</OnsetLocalDate>
     <OffsetFrom>+0300</OffsetFrom>
     <OffsetTo>+0300</OffsetTo>
     <Onset>
        <Frequency>Yearly</Frequency>
        <Interval>1</Interval>
        <Month>October</Month>
        <Day>Sunday</Day>
        <DayOffset>First</DayOffset>
     </Onset>
  </Standard>
  <DayLight>
     <OnsetLocalDate>16010101T030000</OnsetLocalDate>
     <OffsetFrom>+0300</OffsetFrom>
     <OffsetTo>+0300</OffsetTo>
     <Onset>
        <Frequency>Yearly</Frequency>
```

```
                    <Interval>1</Interval>
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>First</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId extractionDefault="true">Asia/Kuwait</TimeZoneId>
                    <TimeZoneId>Asia/Riyadh</TimeZoneId>
                    <TimeZoneId>Asia/Riyadh89</TimeZoneId>
                    <TimeZoneId>Asia/Riyadh88</TimeZoneId>
                    <TimeZoneId>Mideast/Riyadh87</TimeZoneId>
                    <TimeZoneId>Mideast/Riyadh88</TimeZoneId>
                    <TimeZoneId>Mideast/Riyadh89</TimeZoneId>
                    <TimeZoneId>Asia/Riyadh87</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+03:00) Kuwait, Riyadh</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>+0300</OffsetFrom>
                <OffsetTo>+0300</OffsetTo>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>+0300</OffsetFrom>
                <OffsetTo>+0300</OffsetTo>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>W-SU</TimeZoneId>
                    <TimeZoneId extractionDefault="true">Europe/Moscow</TimeZoneId>
                    <TimeZoneId>Europe/Volgograd</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+03:00) Moscow, St. Petersburg, Volgograd</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                <OffsetFrom>+0400</OffsetFrom>
                <OffsetTo>+0300</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>October</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
```

```
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>+0300</OffsetFrom>
            <OffsetTo>+0400</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Indian/Antananarivo</TimeZoneId>
                <TimeZoneId>Africa/Asmara</TimeZoneId>
                <TimeZoneId>Africa/Djibouti</TimeZoneId>
                <TimeZoneId extractionDefault="true">EAT</TimeZoneId>
                <TimeZoneId>Indian/Comoro</TimeZoneId>
                <TimeZoneId>Africa/Asmera</TimeZoneId>
                <TimeZoneId>Africa/Kampala</TimeZoneId>
                <TimeZoneId>Africa/Khartoum</TimeZoneId>
                <TimeZoneId>Africa/Addis_Ababa</TimeZoneId>
                <TimeZoneId>Africa/Dar_es_Salaam</TimeZoneId>
                <TimeZoneId>Africa/Mogadishu</TimeZoneId>
                <TimeZoneId>Africa/Nairobi</TimeZoneId>
                <TimeZoneId>Indian/Mayotte</TimeZoneId>
                <TimeZoneId>Antarctica/Syowa</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+03:00) Nairobi</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0300</OffsetFrom>
            <OffsetTo>+0300</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0300</OffsetFrom>
            <OffsetTo>+0300</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Asia/Tbilisi</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+03:00) Tbilisi</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
```

```
                    <Standard>
                        <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                        <OffsetFrom>+0300</OffsetFrom>
                        <OffsetTo>+0300</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>October</Month>
                            <Day>Sunday</Day>
                            <DayOffset>Last</DayOffset>
                        </Onset>
                    </Standard>
                    <DayLight>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>+0300</OffsetFrom>
                        <OffsetTo>+0300</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>April</Month>
                            <Day>Sunday</Day>
                            <DayOffset>Last</DayOffset>
                        </Onset>
                    </DayLight>
                </TimeZone>
                <TimeZone>
                    <Java>
                        <TimeZoneIds>
                            <TimeZoneId>Iran</TimeZoneId>
                            <TimeZoneId extractionDefault="true">Asia/Tehran</TimeZoneId>
                        </TimeZoneIds>
                    </Java>
                    <Exchange>
                        <TimeZoneIds>
                            <TimeZoneId>(GMT+03:30) Tehran</TimeZoneId>
                        </TimeZoneIds>
                    </Exchange>
                    <Standard>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>+0430</OffsetFrom>
                        <OffsetTo>+0330</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>September</Month>
                            <Day>Tuesday</Day>
                            <DayOffset>Fourth</DayOffset>
                        </Onset>
                    </Standard>
                    <DayLight>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>+0330</OffsetFrom>
                        <OffsetTo>+0430</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>April</Month>
                            <Day>Sunday</Day>
                            <DayOffset>First</DayOffset>
                        </Onset>
```

```
            </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId extractionDefault="true">Asia/Muscat</TimeZoneId>
                <TimeZoneId>Asia/Dubai</TimeZoneId>
                <TimeZoneId>Indian/Reunion</TimeZoneId>
                <TimeZoneId>Indian/Mahe</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+04:00) Abu Dhabi, Muscat</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0400</OffsetFrom>
            <OffsetTo>+0400</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0400</OffsetFrom>
            <OffsetTo>+0400</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Asia/Baku</TimeZoneId>
                <TimeZoneId extractionDefault="true">NET</TimeZoneId>
                <TimeZoneId>Asia/Yerevan</TimeZoneId>
                <TimeZoneId>Indian/Mauritius</TimeZoneId>
                <TimeZoneId>Europe/Samara</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+04:00) Baku</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T030000</OnsetLocalDate>
            <OffsetFrom>+0500</OffsetFrom>
            <OffsetTo>+0400</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>+0400</OffsetFrom>
            <OffsetTo>+0500</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
```

```
                    <Interval>1</Interval>
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>Asia/Yerevan</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+04:00) Yerevan</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                <OffsetFrom>+0500</OffsetFrom>
                <OffsetTo>+0400</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>October</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                <OffsetFrom>+0400</OffsetFrom>
                <OffsetTo>+0500</OffsetTo>
                <Onset>
                    <Frequency>Yearly</Frequency>
                    <Interval>1</Interval>
                    <Month>April</Month>
                    <Day>Sunday</Day>
                    <DayOffset>Last</DayOffset>
                </Onset>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>Asia/Kabul</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+04:30) Kabul</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>+0430</OffsetFrom>
                <OffsetTo>+0430</OffsetTo>
            </Standard>
            <DayLight>
```

```
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0430</OffsetFrom>
            <OffsetTo>+0430</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Asia/Yekaterinburg</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+05:00) Ekaterinburg</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T030000</OnsetLocalDate>
            <OffsetFrom>+0600</OffsetFrom>
            <OffsetTo>+0500</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>+0500</OffsetFrom>
            <OffsetTo>+0600</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Asia/Karachi</TimeZoneId>
                <TimeZoneId extractionDefault="true">PLT</TimeZoneId>
                <TimeZoneId>Asia/Samarkand</TimeZoneId>
                <TimeZoneId>Asia/Tashkent</TimeZoneId>
                <TimeZoneId>Asia/Dushanbe</TimeZoneId>
                <TimeZoneId>Asia/Aqtobe</TimeZoneId>
                <TimeZoneId>Asia/Aqtau</TimeZoneId>
                <TimeZoneId>Asia/Oral</TimeZoneId>
                <TimeZoneId>Asia/Ashgabat</TimeZoneId>
                <TimeZoneId>Indian/Maldives</TimeZoneId>
                <TimeZoneId>Indian/Kerguelen</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+05:00) Islamabad, Karachi</TimeZoneId>
```

```
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0500</OffsetFrom>
                    <OffsetTo>+0500</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0500</OffsetFrom>
                    <OffsetTo>+0500</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId>Asia/Tashkent</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT+05:00) Tashkent</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0500</OffsetFrom>
                    <OffsetTo>+0500</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0500</OffsetFrom>
                    <OffsetTo>+0500</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId extractionDefault="true">Asia/Calcutta</TimeZoneId>
                        <TimeZoneId>IST</TimeZoneId>
                        <TimeZoneId>Asia/Kolkata</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT+05:30) Chennai, Kolkata, Mumbai, New
Delhi</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0530</OffsetFrom>
                    <OffsetTo>+0530</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0530</OffsetFrom>
                    <OffsetTo>+0530</OffsetTo>
                </DayLight>
            </TimeZone>
```

```
<TimeZone>
   <Java>
      <TimeZoneIds>
         <TimeZoneId extractionDefault="true">Asia/Colombo</TimeZoneId>
         <TimeZoneId>Antarctica/Mawson</TimeZoneId>
         <TimeZoneId>Antarctica/Vostok</TimeZoneId>
         <TimeZoneId>Indian/Chagos</TimeZoneId>
         <TimeZoneId>Indian/Cocos</TimeZoneId>
      </TimeZoneIds>
   </Java>
   <Exchange>
      <TimeZoneIds>
         <TimeZoneId>(GMT+06:00) Sri Jayawardenepura</TimeZoneId>
      </TimeZoneIds>
   </Exchange>
   <Standard>
      <OnsetLocalDate>16010101T000000</OnsetLocalDate>
      <OffsetFrom>+0600</OffsetFrom>
      <OffsetTo>+0600</OffsetTo>
   </Standard>
   <DayLight>
      <OnsetLocalDate>16010101T000000</OnsetLocalDate>
      <OffsetFrom>+0600</OffsetFrom>
      <OffsetTo>+0600</OffsetTo>
   </DayLight>
</TimeZone>
<TimeZone>
   <Java>
      <TimeZoneIds>
         <TimeZoneId>Asia/Katmandu</TimeZoneId>
      </TimeZoneIds>
   </Java>
   <Exchange>
      <TimeZoneIds>
         <TimeZoneId>(GMT+05:45) Kathmandu</TimeZoneId>
      </TimeZoneIds>
   </Exchange>
   <Standard>
      <OnsetLocalDate>16010101T000000</OnsetLocalDate>
      <OffsetFrom>+0545</OffsetFrom>
      <OffsetTo>+0545</OffsetTo>
   </Standard>
   <DayLight>
      <OnsetLocalDate>16010101T000000</OnsetLocalDate>
      <OffsetFrom>+0545</OffsetFrom>
      <OffsetTo>+0545</OffsetTo>
   </DayLight>
</TimeZone>
<TimeZone>
   <Java>
      <TimeZoneIds>
         <TimeZoneId extractionDefault="true">Asia/Almaty</TimeZoneId>
         <TimeZoneId>Asia/Novosibirsk</TimeZoneId>
         <TimeZoneId>Asia/Omsk</TimeZoneId>
      </TimeZoneIds>
   </Java>
   <Exchange>
      <TimeZoneIds>
         <TimeZoneId>(GMT+06:00) Almaty, Novosibirsk</TimeZoneId>
      </TimeZoneIds>
```

```
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T030000</OnsetLocalDate>
            <OffsetFrom>+0700</OffsetFrom>
            <OffsetTo>+0600</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>+0600</OffsetFrom>
            <OffsetTo>+0700</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>BST</TimeZoneId>
                <TimeZoneId extractionDefault="true">Asia/Dhaka</TimeZoneId>
                <TimeZoneId>Asia/Dacca</TimeZoneId>
                <TimeZoneId>Asia/Bishkek</TimeZoneId>
                <TimeZoneId>Asia/Thimbu</TimeZoneId>
                <TimeZoneId>Asia/Thimphu</TimeZoneId>
                <TimeZoneId>Asia/Qyzylorda</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+06:00) Astana, Dhaka</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0600</OffsetFrom>
            <OffsetTo>+0600</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0600</OffsetFrom>
            <OffsetTo>+0600</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Asia/Rangoon</TimeZoneId>
            </TimeZoneIds>
        </Java>
```

```
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+06:30) Yangon (Rangoon)</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0630</OffsetFrom>
            <OffsetTo>+0630</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0630</OffsetFrom>
            <OffsetTo>+0630</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Asia/Vientiane</TimeZoneId>
                <TimeZoneId extractionDefault="true">VST</TimeZoneId>
                <TimeZoneId>Asia/Phnom_Penh</TimeZoneId>
                <TimeZoneId>Asia/Saigon</TimeZoneId>
                <TimeZoneId>Asia/Ho_Chi_Minh</TimeZoneId>
                <TimeZoneId>Asia/Bangkok</TimeZoneId>
                <TimeZoneId>Asia/Pontianak</TimeZoneId>
                <TimeZoneId>Asia/Jakarta</TimeZoneId>
                <TimeZoneId>Asia/Hovd</TimeZoneId>
                <TimeZoneId>Antarctica/Davis</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+07:00) Bangkok, Hanoi, Jakarta</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0700</OffsetFrom>
            <OffsetTo>+0700</OffsetTo>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0700</OffsetFrom>
            <OffsetTo>+0700</OffsetTo>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId extractionDefault="true">Asia/Krasnoyarsk</TimeZoneId>
                <TimeZoneId>Indian/Christmas</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+07:00) Krasnoyarsk</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
```

```
                            <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                            <OffsetFrom>+0800</OffsetFrom>
                            <OffsetTo>+0700</OffsetTo>
                            <Onset>
                                <Frequency>Yearly</Frequency>
                                <Interval>1</Interval>
                                <Month>October</Month>
                                <Day>Sunday</Day>
                                <DayOffset>Last</DayOffset>
                            </Onset>
                        </Standard>
                        <DayLight>
                            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                            <OffsetFrom>+0700</OffsetFrom>
                            <OffsetTo>+0800</OffsetTo>
                            <Onset>
                                <Frequency>Yearly</Frequency>
                                <Interval>1</Interval>
                                <Month>April</Month>
                                <Day>Sunday</Day>
                                <DayOffset>Last</DayOffset>
                            </Onset>
                        </DayLight>
                    </TimeZone>
                    <TimeZone>
                        <Java>
                            <TimeZoneIds>
                                <TimeZoneId extractionDefault="true">Asia/Hong_Kong</TimeZoneId>
                                <TimeZoneId>Hongkong</TimeZoneId>
                            </TimeZoneIds>
                        </Java>
                        <Exchange>
                            <TimeZoneIds>
                                <TimeZoneId>(GMT+08:00) Beijing, Chongqing, Hong Kong,
Urumqi</TimeZoneId>
                            </TimeZoneIds>
                        </Exchange>
                        <Standard>
                            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                            <OffsetFrom>+0800</OffsetFrom>
                            <OffsetTo>+0800</OffsetTo>
                        </Standard>
                        <DayLight>
                            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                            <OffsetFrom>+0800</OffsetFrom>
                            <OffsetTo>+0800</OffsetTo>
                        </DayLight>
                    </TimeZone>
                    <TimeZone>
                        <Java>
                            <TimeZoneIds>
                                <TimeZoneId extractionDefault="true">Asia/Irkutsk</TimeZoneId>
                                <TimeZoneId>Asia/Ulaanbaatar</TimeZoneId>
                                <TimeZoneId>Asia/Ulan_Bator</TimeZoneId>
                            </TimeZoneIds>
                        </Java>
                        <Exchange>
                            <TimeZoneIds>
                                <TimeZoneId>(GMT+08:00) Irkutsk, Ulaan Bataar</TimeZoneId>
                            </TimeZoneIds>
```

```
            </Exchange>
            <Standard>
               <OnsetLocalDate>16010101T030000</OnsetLocalDate>
               <OffsetFrom>+0900</OffsetFrom>
               <OffsetTo>+0800</OffsetTo>
               <Onset>
                  <Frequency>Yearly</Frequency>
                  <Interval>1</Interval>
                  <Month>October</Month>
                  <Day>Sunday</Day>
                  <DayOffset>Last</DayOffset>
               </Onset>
            </Standard>
            <DayLight>
               <OnsetLocalDate>16010101T020000</OnsetLocalDate>
               <OffsetFrom>+0800</OffsetFrom>
               <OffsetTo>+0900</OffsetTo>
               <Onset>
                  <Frequency>Yearly</Frequency>
                  <Interval>1</Interval>
                  <Month>April</Month>
                  <Day>Sunday</Day>
                  <DayOffset>Last</DayOffset>
               </Onset>
            </DayLight>
         </TimeZone>
         <TimeZone>
            <Java>
               <TimeZoneIds>
                  <TimeZoneId extractionDefault="true">Asia/Kuala_Lumpur</TimeZoneId>
                  <TimeZoneId>Asia/Kuching</TimeZoneId>
                  <TimeZoneId>Asia/Singapore</TimeZoneId>
                  <TimeZoneId>Singapore</TimeZoneId>
                  <TimeZoneId>Asia/Manila</TimeZoneId>
                  <TimeZoneId>Asia/Brunei</TimeZoneId>
               </TimeZoneIds>
            </Java>
            <Exchange>
               <TimeZoneIds>
                  <TimeZoneId>(GMT+08:00) Kuala Lumpur, Singapore</TimeZoneId>
               </TimeZoneIds>
            </Exchange>
            <Standard>
               <OnsetLocalDate>16010101T000000</OnsetLocalDate>
               <OffsetFrom>+0800</OffsetFrom>
               <OffsetTo>+0800</OffsetTo>
            </Standard>
            <DayLight>
               <OnsetLocalDate>16010101T000000</OnsetLocalDate>
               <OffsetFrom>+0800</OffsetFrom>
               <OffsetTo>+0800</OffsetTo>
            </DayLight>
         </TimeZone>
         <TimeZone>
            <Java>
               <TimeZoneIds>
                  <TimeZoneId extractionDefault="true">Australia/Perth</TimeZoneId>
                  <TimeZoneId>Antarctica/Casey</TimeZoneId>
                  <TimeZoneId>Australia/West</TimeZoneId>
                  <TimeZoneId>Australia/Eucla</TimeZoneId>
```

```
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT+08:00) Perth</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0800</OffsetFrom>
                    <OffsetTo>+0800</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0800</OffsetFrom>
                    <OffsetTo>+0800</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId extractionDefault="true">Asia/Kashgar</TimeZoneId>
                        <TimeZoneId>Asia/Harbin</TimeZoneId>
                        <TimeZoneId>Asia/Urumqi</TimeZoneId>
                        <TimeZoneId>Asia/Chungking</TimeZoneId>
                        <TimeZoneId>PRC</TimeZoneId>
                        <TimeZoneId>Asia/Chongqing</TimeZoneId>
                        <TimeZoneId>Asia/Taipei</TimeZoneId>
                        <TimeZoneId>Asia/Shanghai</TimeZoneId>
                        <TimeZoneId>Asia/Macao</TimeZoneId>
                        <TimeZoneId>CTT</TimeZoneId>
                        <TimeZoneId>Asia/Macau</TimeZoneId>
                        <TimeZoneId>Asia/Choibalsan</TimeZoneId>
                        <TimeZoneId>Asia/Makassar</TimeZoneId>
                        <TimeZoneId>Asia/Ujung_Pandang</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT+08:00) Taipei</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0800</OffsetFrom>
                    <OffsetTo>+0800</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+0800</OffsetFrom>
                    <OffsetTo>+0800</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId>Asia/Tokyo</TimeZoneId>
                        <TimeZoneId extractionDefault="true">JST</TimeZoneId>
                        <TimeZoneId>Japan</TimeZoneId>
                        <TimeZoneId>Asia/Dili</TimeZoneId>
```

```
                    <TimeZoneId>Asia/Jayapura</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+09:00) Osaka, Sapporo, Tokyo</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>+0900</OffsetFrom>
                <OffsetTo>+0900</OffsetTo>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>+0900</OffsetFrom>
                <OffsetTo>+0900</OffsetTo>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId>Asia/Seoul</TimeZoneId>
                    <TimeZoneId>Asia/Pyongyang</TimeZoneId>
                    <TimeZoneId extractionDefault="true">ROK</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+09:00) Seoul</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>+0900</OffsetFrom>
                <OffsetTo>+0900</OffsetTo>
            </Standard>
            <DayLight>
                <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                <OffsetFrom>+0900</OffsetFrom>
                <OffsetTo>+0900</OffsetTo>
            </DayLight>
        </TimeZone>
        <TimeZone>
            <Java>
                <TimeZoneIds>
                    <TimeZoneId extractionDefault="true">Asia/Yakutsk</TimeZoneId>
                    <TimeZoneId>Australia/Yancowinna</TimeZoneId>
                    <TimeZoneId>Australia/Broken_Hill</TimeZoneId>
                </TimeZoneIds>
            </Java>
            <Exchange>
                <TimeZoneIds>
                    <TimeZoneId>(GMT+09:00) Yakutsk</TimeZoneId>
                </TimeZoneIds>
            </Exchange>
            <Standard>
                <OnsetLocalDate>16010101T030000</OnsetLocalDate>
                <OffsetFrom>+1000</OffsetFrom>
                <OffsetTo>+0900</OffsetTo>
```

```
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>October</Month>
                            <Day>Sunday</Day>
                            <DayOffset>Last</DayOffset>
                        </Onset>
                    </Standard>
                    <DayLight>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>+0900</OffsetFrom>
                        <OffsetTo>+1000</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>April</Month>
                            <Day>Sunday</Day>
                            <DayOffset>Last</DayOffset>
                        </Onset>
                    </DayLight>
                </TimeZone>
                <TimeZone>
                    <Java>
                        <TimeZoneIds>
                            <TimeZoneId>Australia/South</TimeZoneId>
                            <TimeZoneId extractionDefault="true">Australia/Adelaide</TimeZoneId>
                        </TimeZoneIds>
                    </Java>
                    <Exchange>
                        <TimeZoneIds>
                            <TimeZoneId>(GMT+09:30) Adelaide</TimeZoneId>
                        </TimeZoneIds>
                    </Exchange>
                    <Standard>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>+1030</OffsetFrom>
                        <OffsetTo>+0930</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>April</Month>
                            <Day>Sunday</Day>
                            <DayOffset>Last</DayOffset>
                        </Onset>
                    </Standard>
                    <DayLight>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>+0930</OffsetFrom>
                        <OffsetTo>+1030</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>October</Month>
                            <Day>Sunday</Day>
                            <DayOffset>Last</DayOffset>
                        </Onset>
                    </DayLight>
                </TimeZone>
                <TimeZone>
                    <Java>
```

```
            <TimeZoneIds>
               <TimeZoneId extractionDefault="true">Australia/Darwin</TimeZoneId>
               <TimeZoneId>ACT</TimeZoneId>
               <TimeZoneId>Australia/North</TimeZoneId>
            </TimeZoneIds>
         </Java>
         <Exchange>
            <TimeZoneIds>
               <TimeZoneId>(GMT+09:30) Darwin</TimeZoneId>
            </TimeZoneIds>
         </Exchange>
         <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0930</OffsetFrom>
            <OffsetTo>+0930</OffsetTo>
         </Standard>
         <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+0930</OffsetFrom>
            <OffsetTo>+0930</OffsetTo>
         </DayLight>
      </TimeZone>
      <TimeZone>
         <Java>
            <TimeZoneIds>
               <TimeZoneId>Australia/Lindeman</TimeZoneId>
               <TimeZoneId>Australia/Queensland</TimeZoneId>
               <TimeZoneId>Australia/Lord_Howe</TimeZoneId>
               <TimeZoneId>Australia/LHI</TimeZoneId>
               <TimeZoneId extractionDefault="true">Australia/Brisbane</TimeZoneId>
            </TimeZoneIds>
         </Java>
         <Exchange>
            <TimeZoneIds>
               <TimeZoneId>(GMT+10:00) Brisbane</TimeZoneId>
            </TimeZoneIds>
         </Exchange>
         <Standard>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+1000</OffsetFrom>
            <OffsetTo>+1000</OffsetTo>
         </Standard>
         <DayLight>
            <OnsetLocalDate>16010101T000000</OnsetLocalDate>
            <OffsetFrom>+1000</OffsetFrom>
            <OffsetTo>+1000</OffsetTo>
         </DayLight>
      </TimeZone>
      <TimeZone>
         <Java>
            <TimeZoneIds>
               <TimeZoneId>Australia/Currie</TimeZoneId>
               <TimeZoneId extractionDefault="true">Australia/Sydney</TimeZoneId>
               <TimeZoneId>Australia/NSW</TimeZoneId>
               <TimeZoneId>Australia/ACT</TimeZoneId>
               <TimeZoneId>Australia/Canberra</TimeZoneId>
               <TimeZoneId>Australia/Victoria</TimeZoneId>
               <TimeZoneId>Australia/Melbourne</TimeZoneId>
               <TimeZoneId>AET</TimeZoneId>
            </TimeZoneIds>
```

```
                    </Java>
                    <Exchange>
                        <TimeZoneIds>
                            <TimeZoneId>(GMT+10:00) Canberra, Melbourne, Sydney</TimeZoneId>
                        </TimeZoneIds>
                    </Exchange>
                    <Standard>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>+1100</OffsetFrom>
                        <OffsetTo>+1000</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>April</Month>
                            <Day>Sunday</Day>
                            <DayOffset>Last</DayOffset>
                        </Onset>
                    </Standard>
                    <DayLight>
                        <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                        <OffsetFrom>+1000</OffsetFrom>
                        <OffsetTo>+1100</OffsetTo>
                        <Onset>
                            <Frequency>Yearly</Frequency>
                            <Interval>1</Interval>
                            <Month>October</Month>
                            <Day>Sunday</Day>
                            <DayOffset>Last</DayOffset>
                        </Onset>
                    </DayLight>
                </TimeZone>
                <TimeZone>
                    <Java>
                        <TimeZoneIds>
                            <TimeZoneId extractionDefault="true">Pacific/Guam</TimeZoneId>
                            <TimeZoneId>Pacific/Saipan</TimeZoneId>
                            <TimeZoneId>Pacific/Port_Moresby</TimeZoneId>
                            <TimeZoneId>Antarctica/DumontDUrville</TimeZoneId>
                            <TimeZoneId>Pacific/Yap</TimeZoneId>
                            <TimeZoneId>Pacific/Truk</TimeZoneId>
                        </TimeZoneIds>
                    </Java>
                    <Exchange>
                        <TimeZoneIds>
                            <TimeZoneId>(GMT+10:00) Guam, Port Moresby</TimeZoneId>
                        </TimeZoneIds>
                    </Exchange>
                    <Standard>
                        <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                        <OffsetFrom>+1000</OffsetFrom>
                        <OffsetTo>+1000</OffsetTo>
                    </Standard>
                    <DayLight>
                        <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                        <OffsetFrom>+1000</OffsetFrom>
                        <OffsetTo>+1000</OffsetTo>
                    </DayLight>
                </TimeZone>
                <TimeZone>
                    <Java>
```

```
            <TimeZoneIds>
                <TimeZoneId>Australia/Tasmania</TimeZoneId>
                <TimeZoneId extractionDefault="true">Australia/Hobart</TimeZoneId>
                <TimeZoneId>Asia/Sakhalin</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+10:00) Hobart</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>+1100</OffsetFrom>
            <OffsetTo>+1000</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>April</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
            <OnsetLocalDate>16010101T020000</OnsetLocalDate>
            <OffsetFrom>+1000</OffsetFrom>
            <OffsetTo>+1100</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>First</DayOffset>
            </Onset>
        </DayLight>
    </TimeZone>
    <TimeZone>
        <Java>
            <TimeZoneIds>
                <TimeZoneId>Asia/Vladivostok</TimeZoneId>
            </TimeZoneIds>
        </Java>
        <Exchange>
            <TimeZoneIds>
                <TimeZoneId>(GMT+10:00) Vladivostok</TimeZoneId>
            </TimeZoneIds>
        </Exchange>
        <Standard>
            <OnsetLocalDate>16010101T030000</OnsetLocalDate>
            <OffsetFrom>+1100</OffsetFrom>
            <OffsetTo>+1000</OffsetTo>
            <Onset>
                <Frequency>Yearly</Frequency>
                <Interval>1</Interval>
                <Month>October</Month>
                <Day>Sunday</Day>
                <DayOffset>Last</DayOffset>
            </Onset>
        </Standard>
        <DayLight>
```

```
                    <OnsetLocalDate>16010101T020000</OnsetLocalDate>
                    <OffsetFrom>+1000</OffsetFrom>
                    <OffsetTo>+1100</OffsetTo>
                    <Onset>
                        <Frequency>Yearly</Frequency>
                        <Interval>1</Interval>
                        <Month>April</Month>
                        <Day>Sunday</Day>
                        <DayOffset>Last</DayOffset>
                    </Onset>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId extractionDefault="true">Asia/Magadan</TimeZoneId>
                        <TimeZoneId>Pacific/Efate</TimeZoneId>
                        <TimeZoneId>SST</TimeZoneId>
                        <TimeZoneId>Pacific/Guadalcanal</TimeZoneId>
                        <TimeZoneId>Pacific/Kosrae</TimeZoneId>
                        <TimeZoneId>Pacific/Noumea</TimeZoneId>
                        <TimeZoneId>Pacific/Ponape</TimeZoneId>
                        <TimeZoneId>Pacific/Norfolk</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT+11:00) Magadan, Solomon Is., New
Caledonia</TimeZoneId>
                    </TimeZoneIds>
                </Exchange>
                <Standard>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+1100</OffsetFrom>
                    <OffsetTo>+1100</OffsetTo>
                </Standard>
                <DayLight>
                    <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                    <OffsetFrom>+1100</OffsetFrom>
                    <OffsetTo>+1100</OffsetTo>
                </DayLight>
            </TimeZone>
            <TimeZone>
                <Java>
                    <TimeZoneIds>
                        <TimeZoneId extractionDefault="true">NST</TimeZoneId>
                        <TimeZoneId>NZ</TimeZoneId>
                        <TimeZoneId>Antarctica/South_Pole</TimeZoneId>
                        <TimeZoneId>Pacific/Auckland</TimeZoneId>
                        <TimeZoneId>Antarctica/McMurdo</TimeZoneId>
                        <TimeZoneId>Pacific/Wallis</TimeZoneId>
                        <TimeZoneId>Asia/Anadyr</TimeZoneId>
                        <TimeZoneId>Pacific/Chatham</TimeZoneId>
                        <TimeZoneId>NZ-CHAT</TimeZoneId>
                    </TimeZoneIds>
                </Java>
                <Exchange>
                    <TimeZoneIds>
                        <TimeZoneId>(GMT+12:00) Auckland, Wellington</TimeZoneId>
                    </TimeZoneIds>
```

```
        </Exchange>
        <Standard>
           <OnsetLocalDate>16010101T020000</OnsetLocalDate>
           <OffsetFrom>+1300</OffsetFrom>
           <OffsetTo>+1200</OffsetTo>
           <Onset>
              <Frequency>Yearly</Frequency>
              <Interval>1</Interval>
              <Month>April</Month>
              <Day>Sunday</Day>
              <DayOffset>Third</DayOffset>
           </Onset>
        </Standard>
        <DayLight>
           <OnsetLocalDate>16010101T020000</OnsetLocalDate>
           <OffsetFrom>+1200</OffsetFrom>
           <OffsetTo>+1300</OffsetTo>
           <Onset>
              <Frequency>Yearly</Frequency>
              <Interval>1</Interval>
              <Month>October</Month>
              <Day>Sunday</Day>
              <DayOffset>First</DayOffset>
           </Onset>
        </DayLight>
     </TimeZone>
     <TimeZone>
        <Java>
           <TimeZoneIds>
              <TimeZoneId extractionDefault="true">Pacific/Fiji</TimeZoneId>
              <TimeZoneId>Asia/Kamchatka</TimeZoneId>
              <TimeZoneId>Pacific/Majuro</TimeZoneId>
              <TimeZoneId>Pacific/Nauru</TimeZoneId>
              <TimeZoneId>Pacific/Wake</TimeZoneId>
              <TimeZoneId>Pacific/Tarawa</TimeZoneId>
              <TimeZoneId>Pacific/Funafuti</TimeZoneId>
           </TimeZoneIds>
        </Java>
        <Exchange>
           <TimeZoneIds>
              <TimeZoneId>(GMT+12:00) Fiji, Kamchatka, Marshall Is.</TimeZoneId>
           </TimeZoneIds>
        </Exchange>
        <Standard>
           <OnsetLocalDate>16010101T000000</OnsetLocalDate>
           <OffsetFrom>+1200</OffsetFrom>
           <OffsetTo>+1200</OffsetTo>
        </Standard>
        <DayLight>
           <OnsetLocalDate>16010101T000000</OnsetLocalDate>
           <OffsetFrom>+1200</OffsetFrom>
           <OffsetTo>+1200</OffsetTo>
        </DayLight>
     </TimeZone>
     <TimeZone>
        <Java>
           <TimeZoneIds>
              <TimeZoneId extractionDefault="true">Pacific/Tongatapu</TimeZoneId>
              <TimeZoneId>Pacific/Enderbury</TimeZoneId>
           </TimeZoneIds>
```

```
                    </Java>
                    <Exchange>
                       <TimeZoneIds>
                           <TimeZoneId>(GMT+13:00) Nuku'alofa</TimeZoneId>
                       </TimeZoneIds>
                    </Exchange>
                    <Standard>
                       <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                       <OffsetFrom>+1300</OffsetFrom>
                       <OffsetTo>+1300</OffsetTo>
                    </Standard>
                    <DayLight>
                       <OnsetLocalDate>16010101T000000</OnsetLocalDate>
                       <OffsetFrom>+1300</OffsetFrom>
                       <OffsetTo>+1300</OffsetTo>
                    </DayLight>
                </TimeZone>
            </TimeZoneMapping>
```

E

Troubleshooting BDSS

This appendix lists troubleshooting issues.

This appendix describes synchronization issues.

E.1 Exchange 2007 Connector Fails First-Time Synchronization

Issue

The first-time synchronization using the Exchange 2007 Connector fails with an error indicating `Certificate chain received from <server> was not trusted causing SSL handshake failure`, but subsequent synchronizations succeed.

Solution

BDSS terminates the synchronization if Oracle WebLogic Server loads a certificate store at runtime that contains any certificates with a signature algorithm named SHA256withRSA. In JDK jdk1.6.0_13 and later versions, the `cacerts` store has two certificates that have this signature algorithm. To prevent this error from occurring, you can first remove these certificates, which have aliases of `ttelesecglobalrootclass2ca` and `ttelesecglobalrootclass3ca`, using the `keytool` utility and then restart Oracle WebLogic Server. Alternatively, you can ignore this error.

> **Note:** The commands included in these instructions assume the default store (typically located in `<JDK>\jre\lib\security`) and the default password, *changeit*.

Use the `keytool` utility to remove certificates as follows:

1. Use the following command to list the certificates in the `cacerts` store. Note all of the aliases with entries containing *Signature algorithm name: SHA256withRSA*.

   ```
   keytool -list - v -keystore cacerts -storepass changeit
   ```

2. For each alias obtained from Step 1, use the following command to remove the certificates from the store:

   ```
   keytool -delete -keystore <keystore file name> -alias <certificate alias name>
   -storepass <store pass phrase>
   ```

   For example, enter

   ```
   keytool -delete -keystore cacerts -alias ttelesecglobalrootclass2ca -storepass
   changegit
   ```

> **Note:** You can also ignore this error if it occurs one time after the start of Oracle WebLogic Server and the start of a **synchronization session**. User synchronization fails when BDSS issues the error, but subsequent synchronization sessions do not fail.

Glossary

attendee resolution

The process of determining information about a given attendee associated with a calendar meeting in the system where the meeting resides. This resolution is required because user identifiers are not guaranteed to be the same across the systems.

BPEL

Business Process Execution Language. An XML-based markup language for composing a set of discrete web services into an end-to-end process flow.

connector

A component within the BDSS architecture that sends data to, and retrieves data from, the PIM servers.

connector name

The name for a set of configurations for a connector that serve a specific set of PIM server instances.

connector instance

A representation of a specific installation of a **connector**. The instance can be a single connector, or it can represent a group of connectors that are accessed from the same load-balanced URI (Universal Resource Identifier).

connector user

A representation of an actual user of a PIM server.

Dispatcher

A **Hub** component that sends batches of users to the **Engine**.

domain

A class of records, such as a calendar, business contacts, personal contacts, or tasks.

echo

A record extracted record from a PIM server whose last update was done by a BDSS-initiated change from a prior Synchronization Session (as opposed to a change made by an end user).

Engine

A **Hub** component that synchronizes the user lists received from the **Dispatcher** with records extracted by from the PIM Servers by a **connector**.

federated PIM

For the purposes of this document, a federated PIM is a PIM that stores a single copy of a record and provides one or more users visibility and access to the record. For example, Siebel can be considered a federated PIM because Siebel allows multiple users to share a single Contact, Task or Calendar record. For example, if a calendar meeting (MeetingA) is organized by Siebel UserA with an attendee Siebel UserB, then both UserA and UserB have visibility and access to a single instance of the calendar record. See also **non-federated PIM**.

Hub

Orchestrates the synchronization of data for the system, includes the **Engine** and the **Dispatcher**.

Hub domain

An abstract representation of a **domain**. Each PIM connector has a set of connector domains that are mapped to a central set of HUB domains.

Hub record

Each Hub record represents a mapping between the PIM records for a user.

Hub user

A mapping of PIM server users to other PIM server users. The Hub user does not represent an actual user; it is a key value that represents a collection of PIM server users. There is a one-to-one relationship such that any PIM server can only associate one user to a Hub user; however, an arbitrary number of PIM servers or PIM users can be mapped into the collection. See also **Hub User ID**.

Hub User ID

An identifier used by the Hub that uniquely identifies a **Hub user**. A Hub User ID is mapped to a **PIM User ID**.

master record

A calendar record containing properties that describe the records' recurrence pattern.

natural map

Indicates a conceptual mapping of ICAL and PIM fields that have the same semantics, but may have different names. ICalendar (ICAL) RFC 2445 defines a fairly complete and generic set of fields. Because each PIM also has a set of fields defined for a calendar record, its fields may not have the same name as their ICAL counterparts. For example, while the ICAL `LOCATION` field maps to the non-indexed Exchange field called `Location`, the ICAL `DTSTART` field would map to the Microsoft Exchange calendar `StartDate` field.

non-federated PIM

For the purposes of this document, a non-federated PIM is a PIM that stores individual copies of related records. For example, Microsoft Exchange is a non-federated PIM because each user has their own copy of Contact, Task and Calendar records. If a calendar meeting (Meeting A) is organized by Exchange User A with an attendee Exchange User B, then both User A and User B get a separate, but related, copy of the calendar record. Exchange relates the two copies by ensuring both have the same meeting identifier (in terms of ICalendar, it is the UID). See also **federated PIM**.

occurrence

An occurrence is a single instance of a recurrence series.

PIM Server Endpoint

A connector-defined value used with configuration metadata to route the synchronization of a given user to a particular cluster of connectors.

PIM User ID

An identifier used by a connector for uniquely identifying a PIM user on the PIM server with which the connector communicates. The PIM User ID can be anything that the connector requires, but it is typically a user account name such as *SBROWN* or a SMTP e-mail address such as *sharon.brown@example.com*.

recurrence exception

A calendar occurrence that has undergone a change that applies only to the occurrence and not the entire recurrence series.

reflective synchronization

Used in development scenarios, reflective synchronization refers to a scenario in which a single PIM server is configured to have 2 connector entries in the BDSS system, so that data can be synchronized from one account in the PIM Server to another account in the server.

Scheduler

An external component that triggers the **Dispatcher** at set time intervals.

synchronization session

A session in which the **Hub** synchronizes the data for a **Hub user** for one or more Hub domains between one or more PIM stores.

synchronization state

Connector-provided data used by the **Engine** to track a user's domain records that have been exported successfully from a PIM server.

upsert record

A Hub record extracted from, or pushed to, a PIM server.